Project Report

Prediction of CO_2 Emissions by country using IBM Watson

Team Members:
Siddhant Gupta - 19BCE10399
Sakshi Goel - 19BCE10396
Jyoti Kasera - 19BCE10326
Shivangi Singh - 19BAI10097

1) INTRODUCTION

1.1 Overview

Carbon emissions and environmental protection issues have brought pressure from the International community during Chinese Economic Development.Recently, the Chinese Govt. announced that carbon emissions per unit of GDP would fall by 60-65% compared with 2005 and non-fossil fuel energy would account for 20% of primary energy consumption by 2030.Hence, forecasting energy consumption is significant to emissions reduction and upgrading energy supply in the Beijing-Tianjin-Hebei region. This study thoroughly analyzes carbon emissions' main energy sources, including coal, petrol, natural gas, and coal power in this region.

1.2 Purpose

The growth of CO2 emissions is a major contributing factor to the speed of climate change.In the current energy consumption landscape, a lot of emphasis is also placed on growing consumption of renewable energy sources (ex: wind, solar). In order to prioritize which initiative has the highest impact on reducing CO2 emissions, states must be equipped with high-performing, predictive tools for future emissions.In this project, we aim to provide such a tool that uses a decision-tree machine-learning algorithm to predict state emissions growth, based on theoretic changes in a state's energy consumption profile.

2) LITERATURE SURVEY

2.1 Existing problem

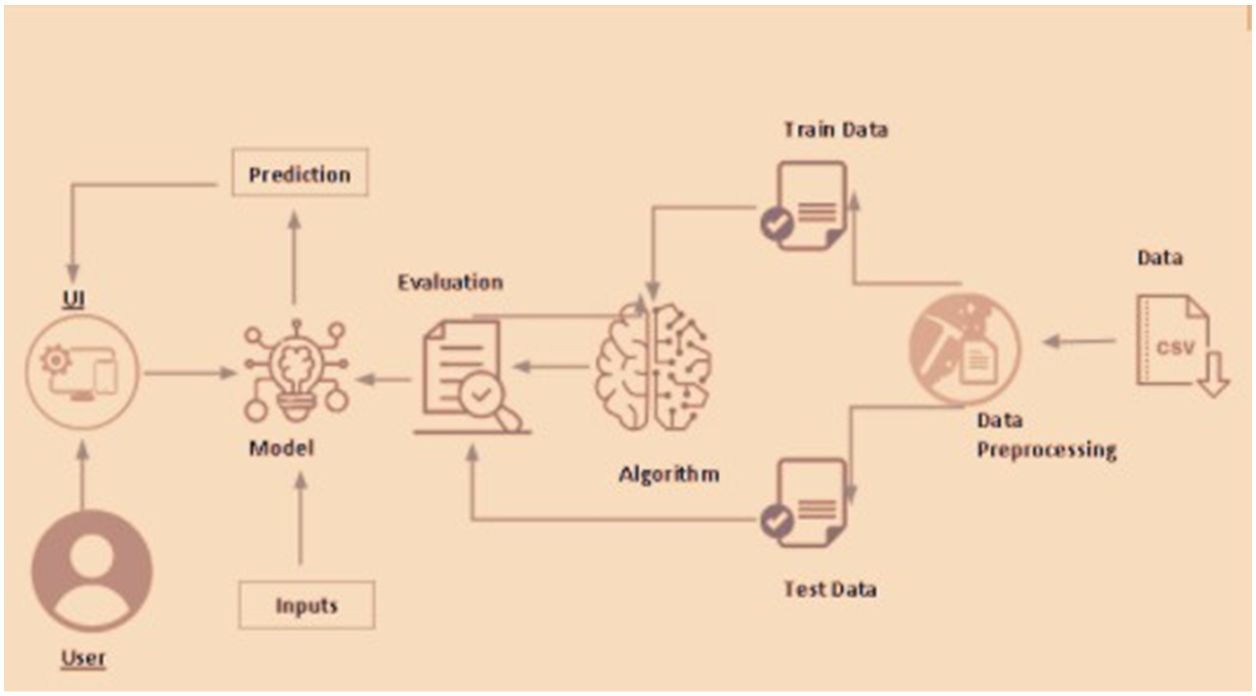
The SVM model was proposed to predict expenditure of carbon (CO2) emission. The energy consumption such as electrical energy and burning coal is input variable that directly increases CO2 emissions were conducted to build the model. The objective is to monitor the CO2 emission based on the electrical energy and burning coal used from the production process. This idea proposes a novel hybrid model combined principal component analysis (PCA) with regularized extreme learning machine (RELM) to make CO_2 emissions prediction based on the data from 1978 to 2014 in China. First eleven variables are selected on the basis of Pearson coefficient test. Partial autocorrelation function (PACF) is utilized to determine the lag phases of historical CO_2 emissions so as to improve the rationality of input selection.

2.2 Proposed solution

A Machine Learning Model for calculating CO2 emission by country, Due to the increasingly deteriorating environment, it is time for the government to upgrade the energy consumption structure. Using Machine Learning Algorithms,Python Language with Machine Learning,Statistics and Graphs and their relations,build web applications using the Flask framework, applying the knowledge on building ML Model and applying different algorithms according to the dataset and based on visualization.

3) THEORETICAL ANALYSIS

3.1 Block diagram



3.2 Hardware / Software designing

Hardware: Laptop(Windows,Linux,MacOS)/Desktop
Software: Python,Python Web Frameworks,Python For Data Analysis,Python For Data Visualization,Machine Learning,Anaconda Navigator,Jupyter,Numpy,Pandas,Matplotlib,Scikit-Learn.

4) EXPERIMENTAL INVESTIGATIONS

Dataset Collection:

Collect the data set or create the dataset.

ML depends heavily on data, without data, it is impossible for an “AI” to learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions.There are many features which are responsible for CO-2 Emission of Countries, e.g. Country Name, Country Code, Indicator

Name etc. For better prediction of the CO2 Emission of Countries, we should consider as many relevant features as possible.
We can collect dataset from different open sources like kaggle.com, data.gov, UCI machine learning repository etc.The kaggle repository link is:
https://www.kaggle.com/ashukr/exploring-co2-emission?select=Indicators.csv

Data Preprocessing :

Here, we are reading the dataset(.csv) from the system using pandas and storing it in a variable ‘df’. It's time to begin building your text classifier! The data has been loaded into a DataFrame called df. The .head() method is particularly informative.We might have your data in .csv files, .excel files or .tsv files or something else. But the goal is the same in all cases. If you want to analyse that data using pandas, the first step will be to read it into a data structure that’s compatible with pandas.load a .csv data file into pandas. There is a function for it, called read_csv().We will need to locate the directory of the CSV file at first (it’s more efficient to keep the dataset in the same directory as your program).

Check Unique Values In Dataset:

Often, a DataFrame will contain columns that are having some unique values from which we can find out the unique records which are present in the dataset.Find the min year and max year for count and also,find how many years of data we have.

CO-2 Emission Of The Countries:

As our dataset is very huge, we are dealing with a few countries like the USA,SGP,IND,BRB,ARB. Here we are selecting the indicator name which is Co2-Emission (metric tons per capita) and also take the country code to find the co2 emission over the time.Select CO2 emissions for the Arab Country and stage is just those indicators matching the ARB for country code and CO2 emissions over time.Select CO2 emissions for the Barbados Country and stage is just those indicators matching the BRB for country code and CO2 emissions over time.Select CO2 emissions for the India Country and stage is just those indicators matching the IND for country code and CO2 emissions. Select CO2 emissions for the Singapore Country and stage is just those indicators matching the SGP for country code and CO2 emissions over time.

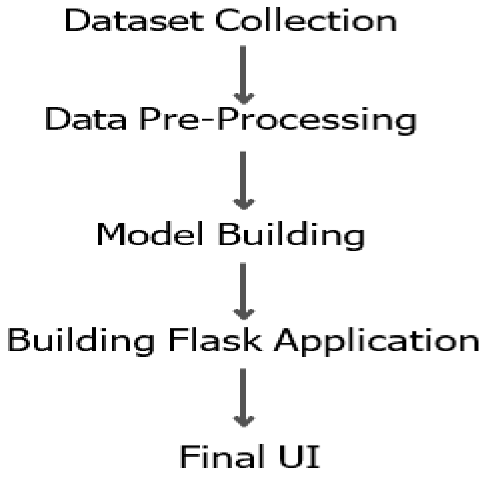
Observing Target,Numerical And Categorical Columns:

Here, data.types will return us all the different types of data present in our data and return_counts will give us the count.Let's start observing the categorical and numerical columns and check whether the above count is correct or not.

Data Visualization:

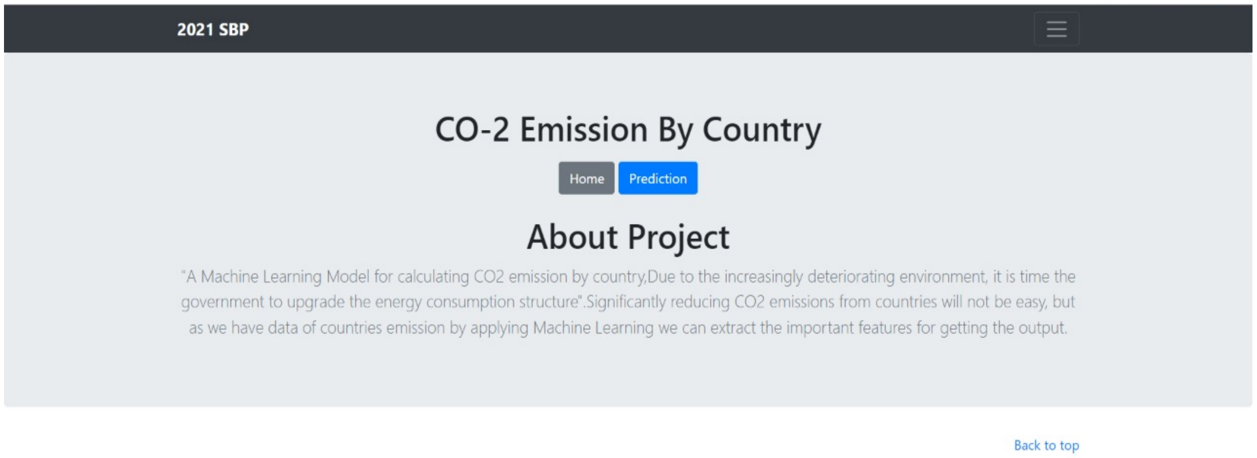
Data visualization is where a given data set is presented in a graphical format. It helps the detection of patterns, trends and correlations that might go undetected in text-based data.Understanding your data and the relationship present within it is just as important as any algorithm used to train your machine learning model. In fact, even the most sophisticated machine learning models will perform poorly on data that wasn't visualized and understood properly.

5) FLOWCHART



1. Install Required Libraries.
2. Data Collection.
    - Collect the dataset or Create the dataset
3. Data Pre- processing.
    - Import the Libraries.
    - Importing the dataset.
    - Check unique values in dataset
    - Check CO-2 Emissions of the countries.
    - Understanding Data Type and Summary of features.
    - Observing Target,Numerical and Categorical Columns
    - Checking for Null Values.
    - Data Visualization.
    - Splitting Data into Train and Test.
4. Model Building
    - Training and testing the model
    - Evaluation of Model
    - Saving the Model
5. Application Building
    - Create an HTML file
    - Build a Python Code
6. Final UI
    - Dashboard Of the flask app.

6) RESULT

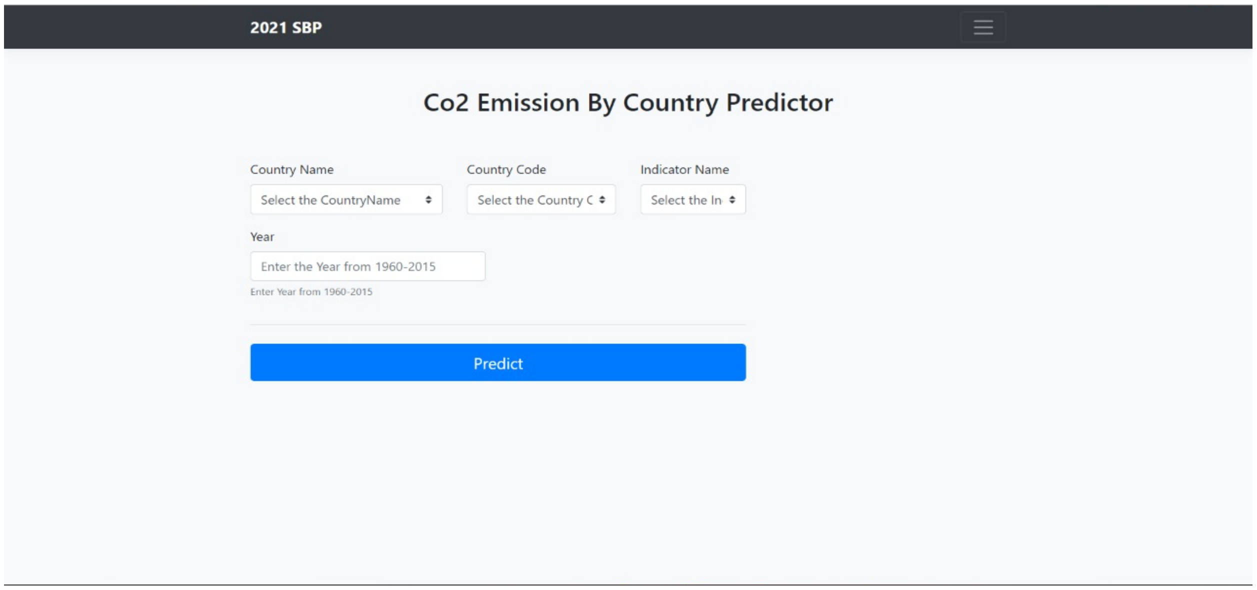


2021 SBP

CO-2 Emission By Country

About Project

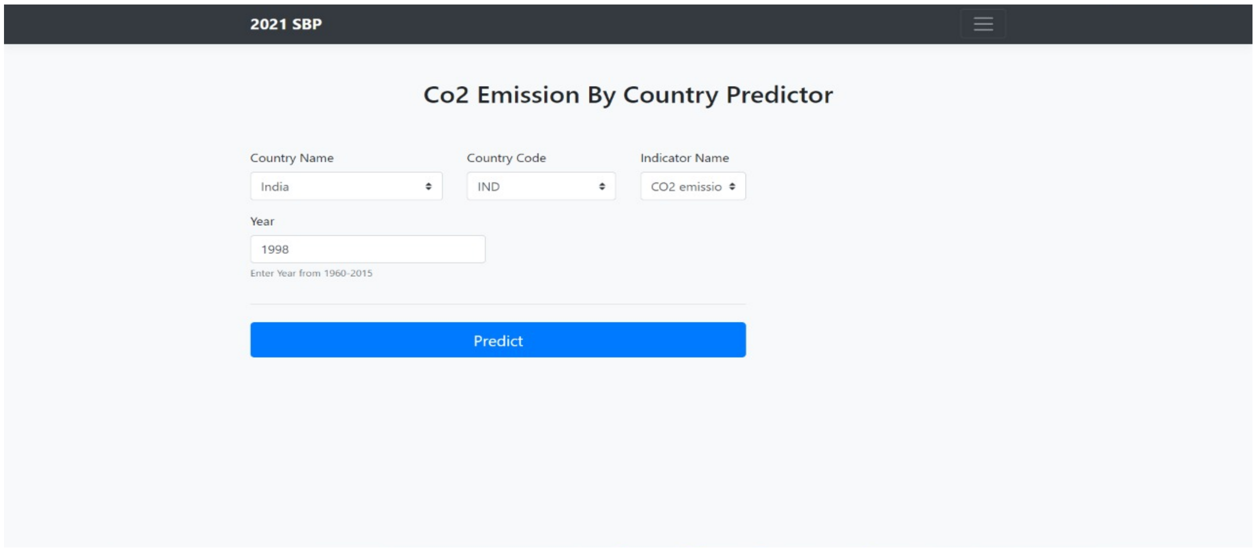
"A Machine Learning Model for calculating CO2 emission by country,Due to the increasingly deteriorating environment, it is time the government to upgrade the energy consumption structure".Significantly reducing CO2 emissions from countries will not be easy, but as we have data of countries emission by applying Machine Learning we can extract the important features for getting the output.



2021 SBP

Co2 Emission By Country Predictor

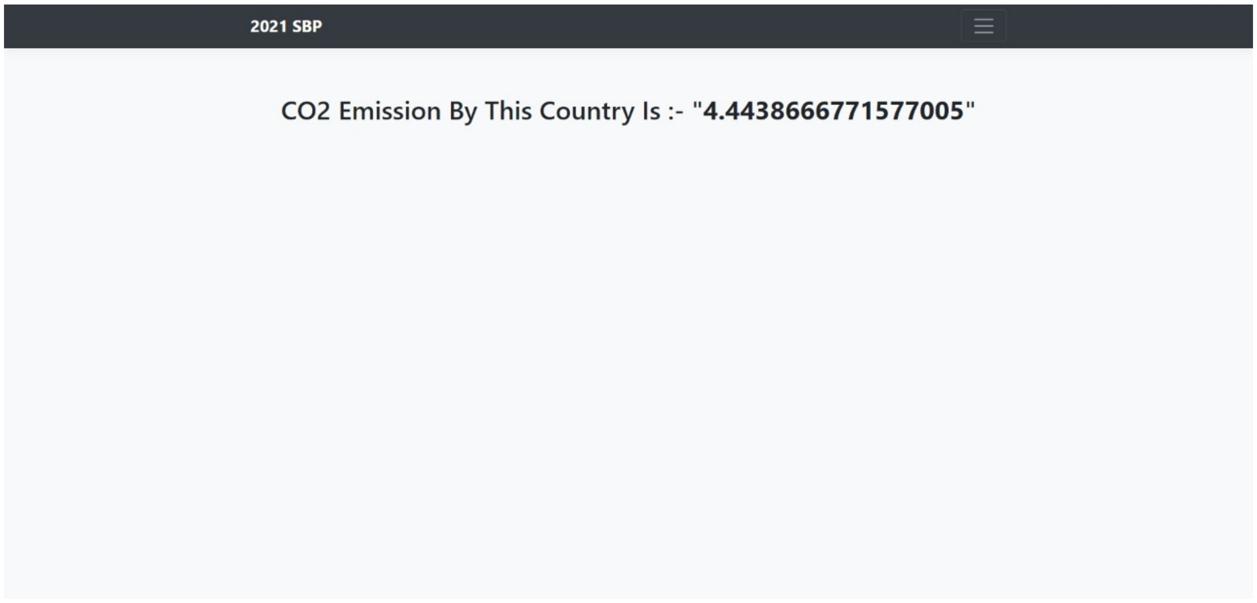
Country Name: Select the CountryName
Country Code: Select the Country C
Indicator Name: Select the In
Year: Enter the Year from 1960-2015
Enter Year from 1960-2015



2021 SBP

Co2 Emission By Country Predictor

Country Name: India
Country Code: IND
Indicator Name: CO2 emissio
Year: 1998
Enter Year from 1960-2015


2021 SBP

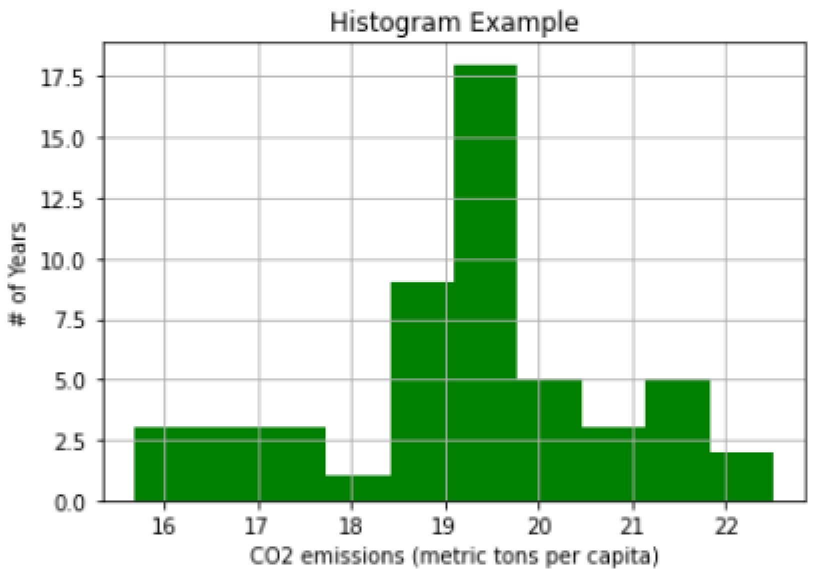
CO2 Emission By This Country Is :- "**4.4438666771577005**"

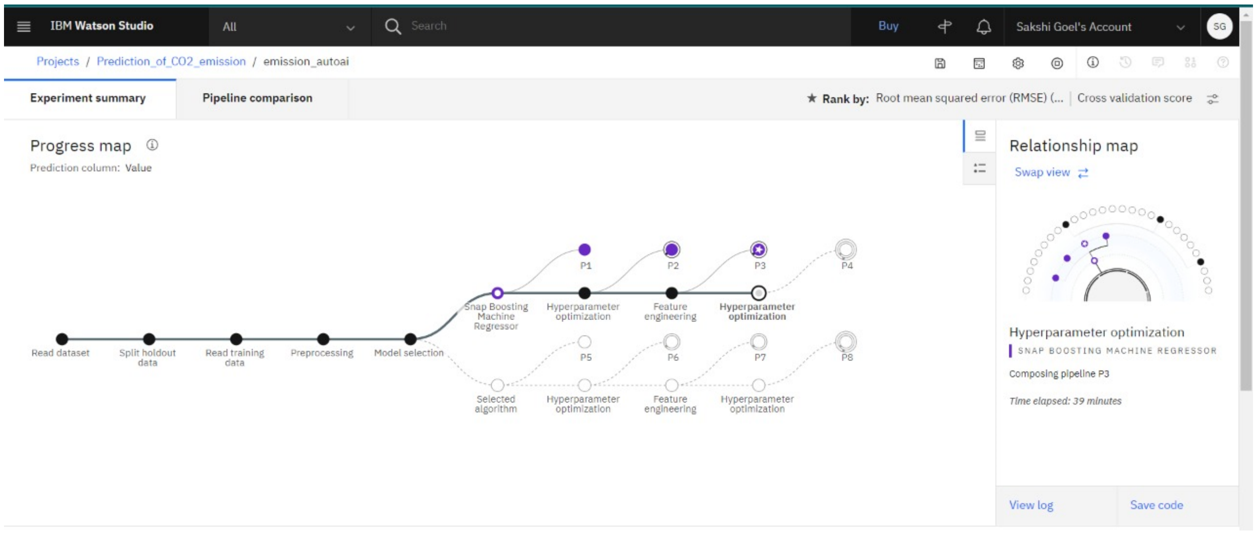
```
# the histogram of the data
plt.hist(hist_data, 10, density=False, facecolor='green')

plt.xlabel(stage['IndicatorName'].iloc[0])
plt.ylabel('# of Years')
plt.title('Histogram Example')

plt.grid(True)

plt.show()
```



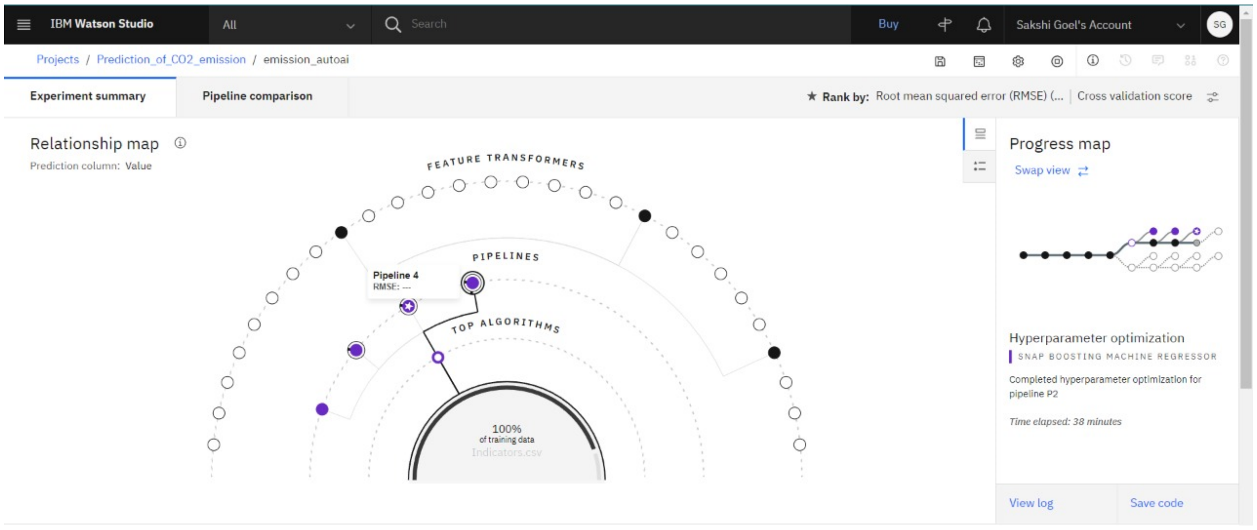


Hyperparameter optimization

SNAP BOOSTING MACHINE REGRESSOR

Composing pipeline P3

Time elapsed: 39 minutes

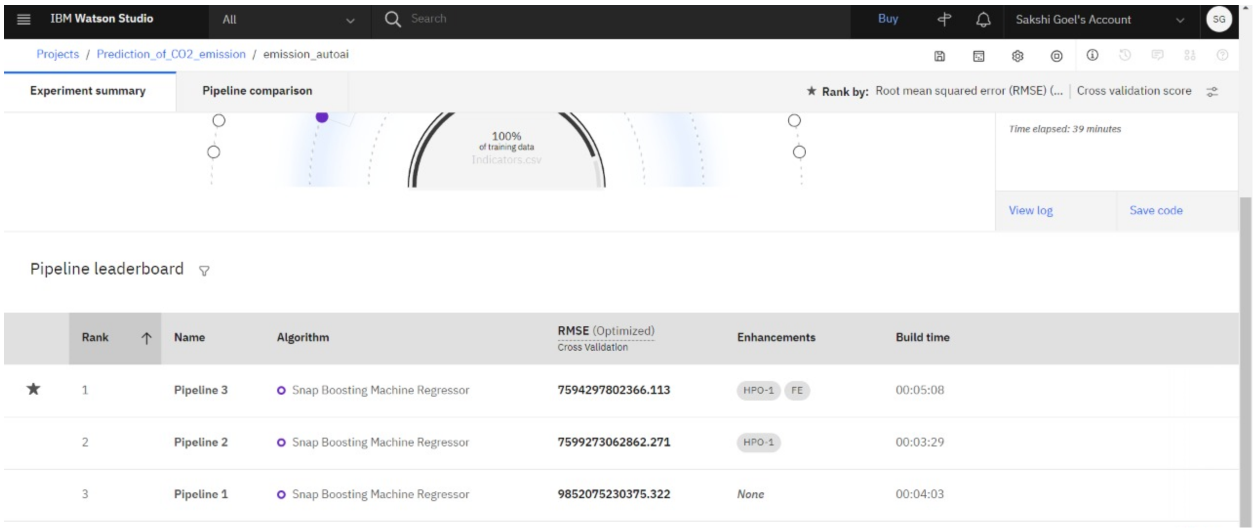


Hyperparameter optimization

SNAP BOOSTING MACHINE REGRESSOR

Completed hyperparameter optimization for pipeline P2

Time elapsed: 38 minutes



Pipeline leaderboard

	Rank ↑	Name	Algorithm	RMSE (Optimized) Cross Validation	Enhancements	Build time
★	1	Pipeline 3	Snap Boosting Machine Regressor	7594297802366.113	HPO-1 FE	00:05:08
	2	Pipeline 2	Snap Boosting Machine Regressor	7599273062862.271	HPO-1	00:03:29
	3	Pipeline 1	Snap Boosting Machine Regressor	9852075230375.322	*None*	00:04:03

7) ADVANTAGES & DISADVANTAGES

ADVANTAGES

With the advantages such as fast convergence speed, high training accuracy and no manual tuning, the ELM model has been successfully applied to forecasting problems in many fields, such as wind speed,electricity load, oil price and so on.However,ELM is based on the empirical risk minimization principle which easily causes an overfitting phenomenon.

Therefore, in order to guarantee the global optimization and generalization ability,the RELM model, in which the calculation process of Moore-Penrose generalized inverse and the introduction of the regularization factor are added to ELM, is used for CO2 emission prediction in this paper.

DISADVANTAGES

Since,we have a large amount of dataset and it required a greater amount of GPU. So it takes time to deploy the model on the cloud and random and estimators were used to get an approximation which can achieve a higher percentage of accuracy.

RELM, a new kind of neural network , is firstly introduced into CO2 emission prediction, which overcomes the disadvantages of slow learning speed, the need of numerous training samples, overfitting and so on in the previous research.

The correlations among influential factors are paid close attention in this paper, thus PCA is utilized to manipulate them for dimension reduction to improve the computational efficiency and forecasting precision.

8) APPLICATIONS

- Machine Learning models such as SVM, tuned SVM, linear model etc are used in Estimating CO2 emissions from Electricity generation
- The banking and Financial services industry uses Machine Learning to detect and reduce fraud, measure market risk and identify opportunities.
- Machine Learning plays a key part in security as they typically use predictive analysis to improve services and performance, but also to detect anomalies, fraud, understand consumer behavior and enhance data security.
- ML Algorithms are also known for its recommendation algorithms like in Facebook or YouTube where similar content will be suggested to engage the user

9) CONCLUSION

The Support Vector Machine model can be applied to predict CO2 emissions from energy consumption which can give us more accuracy. This model is used to monitor electrical energy and burning coal which affect the amount of CO2 emitted. Trial and error approach was applied in order to obtain a better prediction model with a lower error. The results obtained show that the lower error (RMSE) value was 0.004 with optimal parameters for the SVM model of 0.1 for the C parameter and 0 for Epsilon. Prediction with high accuracy can give information concerning CO2 emissions Furthermore, the main objective in this work is to achieve a lower RMSE when designing the model prediction. It can be concluded that with the high accuracy of the prediction model, then the lower RMSE value must be obtained .

10) FUTURE SCOPE

With increasing investment in adaptation, one important challenge for the future will be to establish the right sets of indicators for identifying priorities, as well as monitoring and evaluation frameworks for adaptation. While progress on mitigation can be interpreted from trends in national GHG emissions,comparable measurable outcomes do not yet exist for adaptation. The difficulties in monitoring and evaluating adaptation range from the ambiguous definition of adaptation to the identification of targets and the choice of indicators used to monitor performance. Consequently, while international discussions on adaptation have focused on implementation of adaptation and the associated costs, systematic evaluation of how much progress is being made in this direction is generally lacking and needs further development.

11) BIBLIOGRAPHY

- https://www.ibm.com/ibm/responsibility/2015/assets/downloads/IBM_2015_CR_report.pdf
- https://www.eeer.org/upload/eer-1489554553.pdf
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html#:~:text=A%20random%20forest%20regressor.,accuracy%20and%20control%20 over%2d Fitting
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html#:~:text=A%20random%20forest%20regressor.,accuracy%20and%20control%20 over%2d Fitting

12) APPENDIX

A. Source Code

```
# importing the necessary dependencies
from flask import Flask,request,render_template
import numpy as np
import pandas as pd
import pickle
import os
app=Flask(__name__)# initializing a flask app
#filepath="I:\SmartBridge Projects\Co2 emission\co2.pickle"
#model=pickle.load(open(co2.pickle,'rb'))
with open('co2.pickle', 'rb') as handle:
    model = pickle.load(handle)
@app.route('/')# route to display the home page
def home():
    return render_template('index.html') #rendering the home page
@app.route('/Prediction',methods=['POST','GET'])
def prediction(): # route which will take you to the prediction page
    return render_template('index1.html')
@app.route('/Home',methods=['POST','GET'])
def my_home():
    return render_template('index.html')
@app.route('/predict',methods=["POST","GET"])# route to show the predictions
in a web UI
def predict():
    #  reading the inputs given by the user
    input_feature=[float(x) for x in request.form.values() ]
    features_values=[np.array(input_feature)]
    feature_name=['CountryName', 'CountryCode', 'IndicatorName','Year']
    x=pd.DataFrame(features_values,columns=feature_name)
        # predictions using the loaded model file
    prediction=model.predict(x)
    print("Prediction is:",prediction)
     # showing the prediction results in a UI
    return render_template("result.html",prediction=prediction[0])
if __name__=="__main__":

    # app.run(host='0.0.0.0', port=8000,debug=True)    # running the app
    port=int(os.environ.get('PORT',5000))
    app.run(port=port,debug=True,use_reloader=False)
```

In [98]:
```
###BULIDING MODEL
#RandomForestRegressor
```

In [99]:
```
from sklearn.ensemble import RandomForestRegressor
```

In [100…
```
rand=RandomForestRegressor(n_estimators=10,random_state=52)
rand.fit(x_train,y_train)
```

```
C:\Users\hp\AppData\Local\Temp/ipykernel_14312/1340635713.py:2: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please
change the shape of y to (n_samples,), for example using ravel().
  rand.fit(x_train,y_train)
```
Out[100…
```
RandomForestRegressor(n_estimators=10, random_state=52)
```

In [101…
```
x_test
```

Out[101…
```
array([[   3,    3,  602, 1965],
       [   0,    0,  775, 1994],
       [   2,    2,  843, 2011],
       ...,
       [   4,    4,   40, 1987],
       [   0,    0,  854, 1971],
       [   1,    1,  236, 1990]], dtype=int64)
```

In [102…
```
from collections import Counter as c
c(data["CountryCode"])
```

Out[102…
```
Counter({0: 17115, 1: 17611, 2: 35721, 3: 22600, 4: 24425})
```

In [109…
```
###Multi Linear Regression
from sklearn.linear_model import LinearRegression
```

In [110…
```
mr = LinearRegression()
mr.fit(x_train,y_train)
```

Out[110…
```
LinearRegression()
```

In [111…
```
y_linpred=mr.predict(x_test)
```

In [112…
```
r2_score(y_test,y_linpred)
```

Out[112…
```
0.00918262523882607
```

In [113…
```
data.head()
```

Out[113…

	CountryName	CountryCode	IndicatorName	IndicatorCode	Year	Value
0	0	0	40	1181	1960	1.335609e+02
1	0	0	44	1204	1960	8.779760e+01
2	0	0	45	1205	1960	6.634579e+00
3	0	0	46	1206	1960	8.102333e+01
4	0	0	86	636	1960	3.000000e+06

In [114…
```
import pickle
pickle.dump(rand,open("co2.pickle","wb"))
```

In [115…
```
###DecisionTreeRegressor
```

In [116…
```
from sklearn.tree import DecisionTreeRegressor
```

In [117…
```
dt=DecisionTreeRegressor(random_state=0)
```

In [118…
```
dt.fit(x_train,y_train)
```

Out[118…
```
DecisionTreeRegressor(random_state=0)
```

In [119…
```
y_dtpred=dt.predict(x_test)
```

In [120…
```
r2_score(y_test,y_dtpred)
```

Out[120…
```
0.9894087744610102
```

In [121…
```
import pickle
pickle.dump(dt,open("co2.pickle","wb"))
```