

## Data Collection and Preprocessing Phase

Date	15 july 2024
Team ID	740040
Project Title	Predicting co2 emissions by countries using machine learning
Maximum Marks	6 Marks

### Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	Description
Data Overview	Basic statistics, dimensions, and structure of the data.
Univariate Analysis	Exploration of individual variables (mean, median, mode, etc.).
Bivariate Analysis	Relationships between two variables (correlation, scatter plots).
Multivariate Analysis	Patterns and relationships involving multiple variables.
Outliers and Anomalies	Identification and treatment of outliers.
<b>Data Preprocessing Code Screenshots</b>	

## Loading Data

### READING THE DATASET

```
[ ] #Reading the dataset
data=pd.read_csv("/content/Indicators.csv")
```

### DATASET

```
[ ] data.shape
```

```
(5656458, 6)
```

```
[ ] #Representing first 5 values from the dataset
```

```
data.head()
```

	CountryName	CountryCode	IndicatorName	IndicatorCode	Year	Value
0	Arab World	ARB	Adolescent fertility rate (births per 1,000 wo...	SP.ADO.TFRT	1960	1.335609e+02
1	Arab World	ARB	Age dependency ratio (% of working-age populat...	SP.POP.DPND	1960	8.779760e+01
2	Arab World	ARB	Age dependency ratio, old (% of working-age po...	SP.POP.DPND.OL	1960	6.634579e+00
3	Arab World	ARB	Age dependency ratio, young (% of working-age ...	SP.POP.DPND.YG	1960	8.102333e+01
4	Arab World	ARB	Arms exports (SIPRI trend indicator values)	MS.MIL.XPRT.KD	1960	3.000000e+06

## Handling Missing Data

### HANDLING MISSING DATA

```
[ ] #Returns true if any columns having null values
```

```
data.isnull().any()
```

```
CountryName    False
CountryCode    False
IndicatorName   False
IndicatorCode   False
Year           False
Value          False
dtype: bool
```

```
[ ] #Used for finding the null values
```

```
data.isnull().sum()
```

```
CountryName    0
CountryCode    0
IndicatorName   0
IndicatorCode   0
Year           0
Value          0
dtype: int64
```

## Splitting data

### SPLITTING DATA

```
[ ] #Splitting dataset into train and test
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=1)
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(4525166, 4)
(1131292, 4)
(4525166, 1)
(1131292, 1)
```

Model training	<p><b>TRAINING THE MODEL</b></p> <pre>[ ] #Training the model  from sklearn.ensemble import RandomForestRegressor rand = RandomForestRegressor(n_estimators=10,random_state=52,n_jobs=-1) rand.fit(x_train,y_train)</pre> <p>&lt;ipython-input-53-6c838af2cded&gt;:5: DataConversionWarning: A column-vector y was passed whe rand.fit(x_train,y_train)</p> <pre>*      RandomForestRegressor       RandomForestRegressor(n_estimators=10, n_jobs=-1, random_state=52)</pre>
Model evaluation	<pre>ypred = rand.predict(x_test) print(ypred)</pre> <pre>[2.23526022e+00 7.92900024e+01 4.63113569e+01 ... 9.33333333e+00  3.45749686e+01 6.00578821e+09]</pre> <pre>[ ] #Accuracy score #To check how well our model is performing on the test data rand.score(x_train,y_train)</pre> <pre>0.9829119449040941</pre>
Saving the model	<p><b>SAVING THE MODEL</b></p> <pre>[ ] #Saving our model by importing pickle file  import pickle pickle.dump(rand, open('C02.pickle', 'wb'))</pre>