GLOBAL COUNTRY MACRO DATASET

Machine Learning project to predict Infant Mortality Rate

Bhavan's Vivekananda College of Science, Humanities and Commerce

Group 5

N.Keerthana Reddy

A.Deepak Mudiraj

Lokesh Kumar

K.Manohar

ABSTRACT

This project analyzes factors affecting infant mortality rates globally, using variables such as life expectancy, birth rates, GDP, and healthcare metrics. Through regression analysis, significant trends and relationships between socio-economic indicators and infant health outcomes are identified, providing insights into potential areas for policy improvements.

OBJECTIVE

To identify a suitable machine learning model for predicting infant mortality rates using various socioeconomic and health-related indicators, aiming to improve public health strategies and outcomes.

CONTENT

1. INTRODUCTION - 4
2. LITERATURE REVIEW - 5
3. DATA PREPROCESSING - 7
4. EXPLORATORY DATA ANALYSIS - 12
5. DATA MODELING & EVALUATION - 20
6. SUMMARY - 27
7. APPENDIX - 31

INTRODUCTION

The global country macro dataset is a collection of detailed information about countries around the world, covering areas like the economy, population statistics, health, and environmental factors. This data helps us understand global trends and patterns, which can be used to make recommendations for better policies.

The dataset includes important details such as a country's economic performance (GDP), birth rates, infant mortality rates, how long people live (life expectancy), the percentage of people living in cities, and employment levels. This gives a complete picture of a country’s overall development and quality of life.

This dataset enables researchers to uncover relationships between variables, identify disparities between countries, and create predictive models supporting informed decision-making in healthcare, economic development, and environmental sustainability.

LITERATURE REVIEW-1

Measuring Global Macroeconomic Uncertainty and Cross-Country Uncertainty Spillovers by Graziano Moramarco

- Research into global macroeconomic uncertainty often focuses on how uncertainty spreads and affects different countries. A common method used to study this is the Global Vector Autoregressive (GVAR) model. This model helps researchers understand how economic uncertainty in one country can impact others by measuring both local and international sources of uncertainty at the same time.

- The GVAR model incorporates a range of indicators, including financial market volatility, economic policy uncertainty, and survey-forecast measures, enabling it to capture fluctuations during significant economic disruptions such as the global financial crisis and the COVID-19 pandemic.

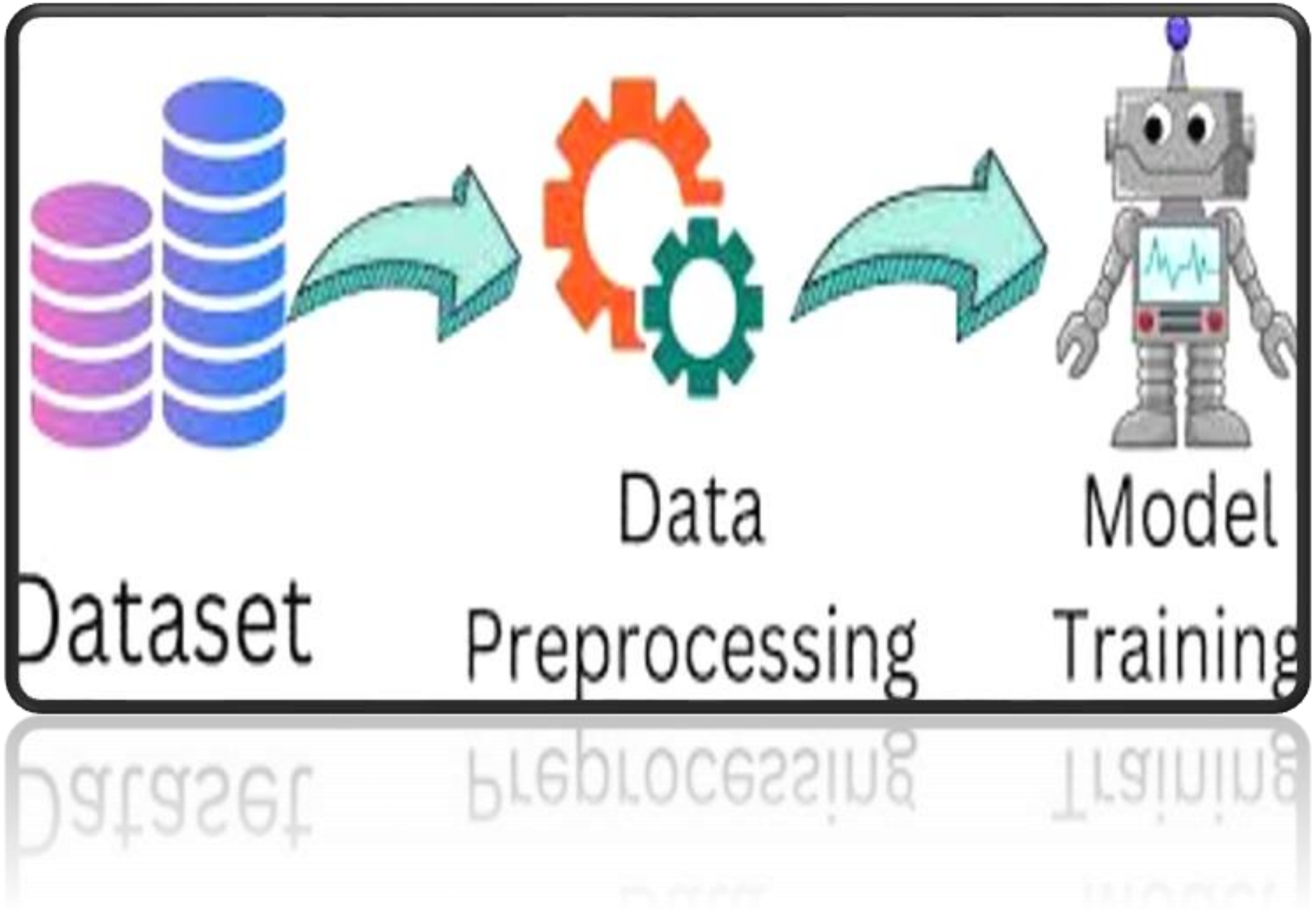
LITERATURE REVIEW-2

Measuring the impact of the digital economy in developing countries: A systematic review and meta- analysis by Abdulkarim A. Oloyede

- Research indicates that the digital economy is a major driver of global growth, but accurately measuring its impact is difficult, especially in developing nations. Current measurement methods vary and often overlook important aspects, potentially underestimating the digital economy’s full contribution. A unified and flexible measurement system is needed to better assess its impact and guide effective policies for sustainable growth.

- Existing indices often fail to fully capture the digital economy’s scope, leading to potential underestimations of its contributions. This highlights the need for a unified, context-sensitive framework that supports accurate measurement and facilitates policy-making for sustainable digital economic growth worldwide.

DATA PREPROCESSING



Data

Dataset: Our dataset contains 35 variables and 195 records

Source: https://drive.google.com/drive/folders/1vGSRCnhqSxEH53BgLqhh32F1qNKqfOim?usp=drive_link

Variables:

Country	Density	Abbreviati	Agricultura	Land Area(	Armed For	Birth Rate	Calling Coc	Capital/Ma	Co2-Emiss	CPI
Afghanista	60	AF	58.10%	6,52,230	3,23,000	32.49	93	Kabul	8,672	149.9
Albania	105	AL	43.10%	28,748	9,000	11.78	355	Tirana	4,536	119.05
Algeria	18	DZ	17.40%	23,81,741	3,17,000	24.28	213	Algiers	1,50,006	151.36
Andorra	164	AD	40.00%	468		7.2	376	Andorra la	469	
Angola	26	AO	47.50%	12,46,700	1,17,000	40.73	244	Luanda	34,693	261.73
Antigua an	223	AG	20.50%	443	0	15.33	1	St. John's,	557	113.81
Argentina	17	AR	54.30%	27,80,400	1,05,000	17.02	54	Buenos Air	2,01,348	232.75
Armenia	104	AM	58.90%	29,743	49,000	13.99	374	Yerevan	5,156	129.18
Australia	3	AU	48.20%	77,41,220	58,000	12.6	61	Canberra	3,75,908	119.8
Austria	109	AT	32.40%	83,871	21,000	9.7	43	Vienna	61,448	118.06
Azerbaijan	123	AZ	57.70%	86,600	82,000	14	994	Baku	37,620	156.32
The Bahan	39	BS	1.40%	13,880	1,000	13.97	1	Nassau, Ba	1,786	116.22
Bahrain	2,239	BH	11.10%	765	19,000	13.99	973	Manama	31,694	117.59
Banglades	1,265	BD	70.60%	1,48,460	2,21,000	18.18	880	Dhaka	84,246	179.68
Barbados	668	BB	23.30%	430	1,000	10.65	1	Bridgetow	1,276	134.09
Belarus	47	BY	42.00%	2,07,600	1,55,000	9.9	375	Minsk	58,280	
Belgium	383	BE	44.60%	30,528	32,000	10.3	32	City of Bru	96,889	117.11
Belize	17	BZ	7.00%	22,966	2,000	20.79	501	Belmopan	568	105.68
Benin	108	BJ	33.30%	1,12,622	12,000	36.22	229	Porto-Nov	6,476	110.71

Continuous variable	**Categorical variable**
Birth Rate	Abbreviation
Calling Code	Agricultural Land(%)
Fertility rate	Land Area(Km2)
Infant mortality	Armed Forces size
Life expectancy	Capital/Major City
Maternal mortality ratio	Co2-Emissions
Physicians per thousand	CPI
Latitude	CPI Change (%)
Longitude	Currency-Code
	Forested Area (%)
	Gasoline Price
	GDP
	Gross primary education enrollment (%)
	Gross tertiary education enrollment (%)
	Largest city

Data Cleaning

- Removing Columns: We removed unnecessary columns from the data for our analysis.

- Checked for missing values and unique values

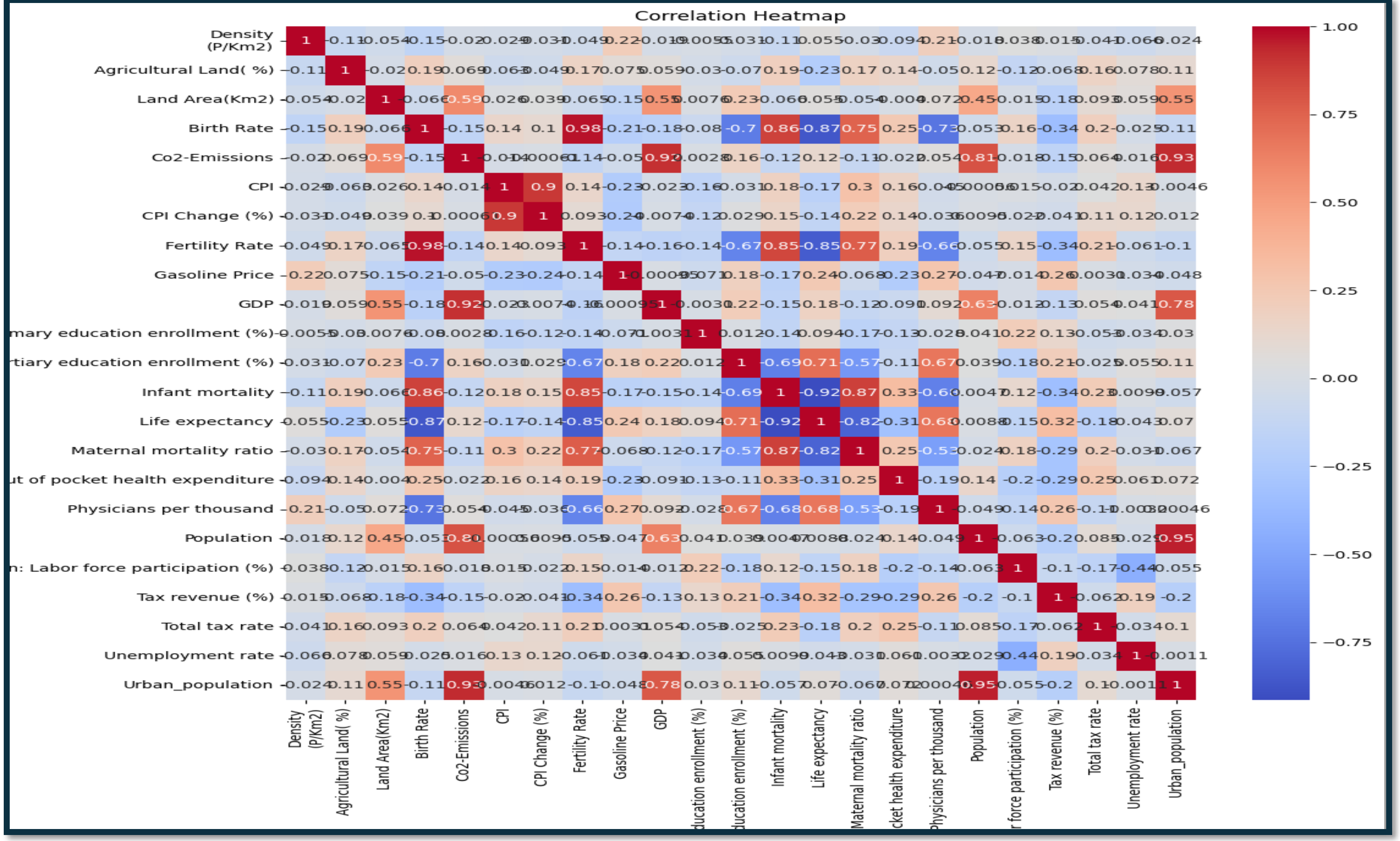
- We removed symbols from the data, such as '$', '%', and others.

- The data does not contain any unique values.

- We replaced the missing numeric values with the mean and the missing categorical values with the mode.

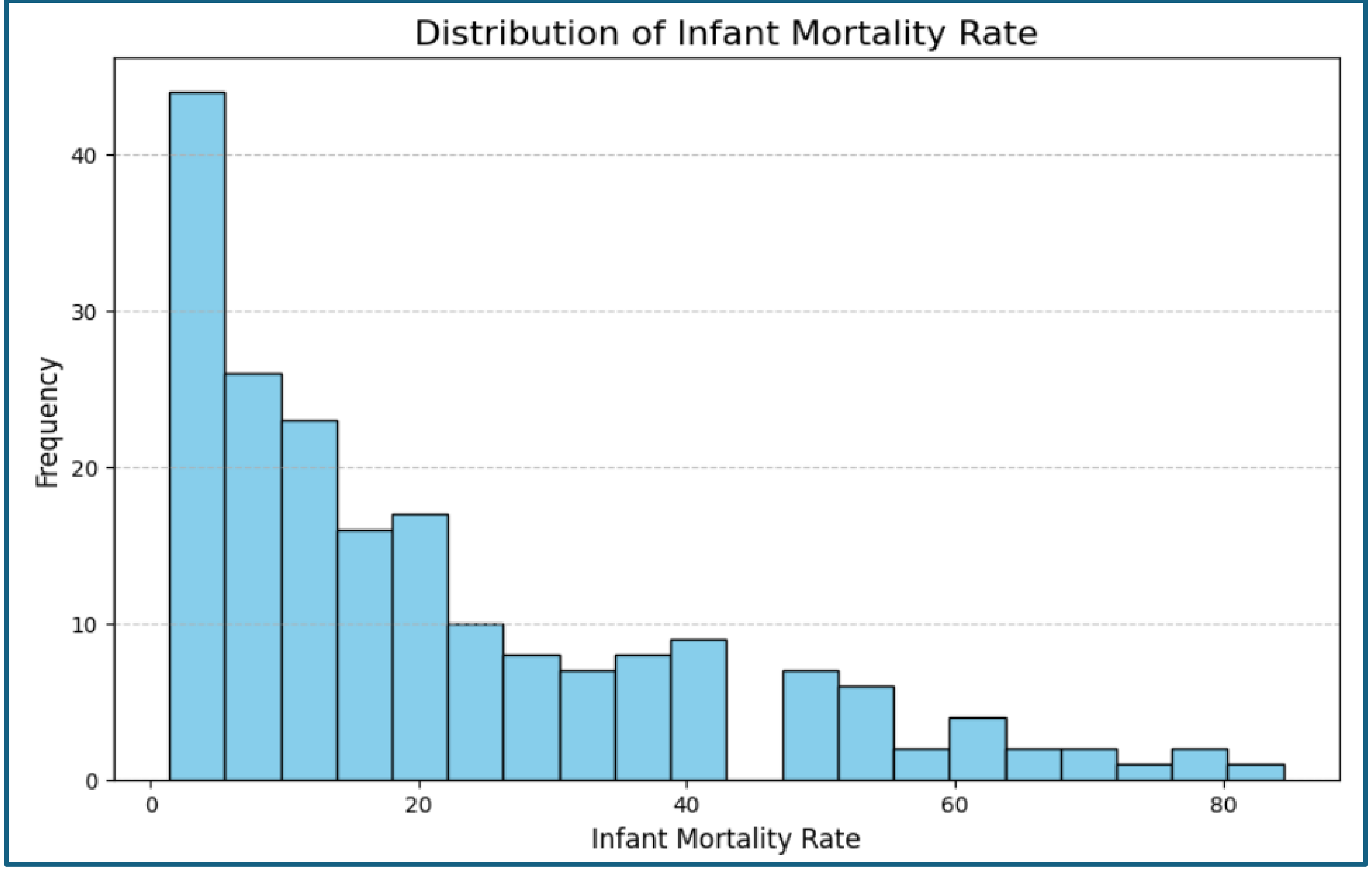
EXPLORATORY DATA ANALYSIS

Correlation matrix



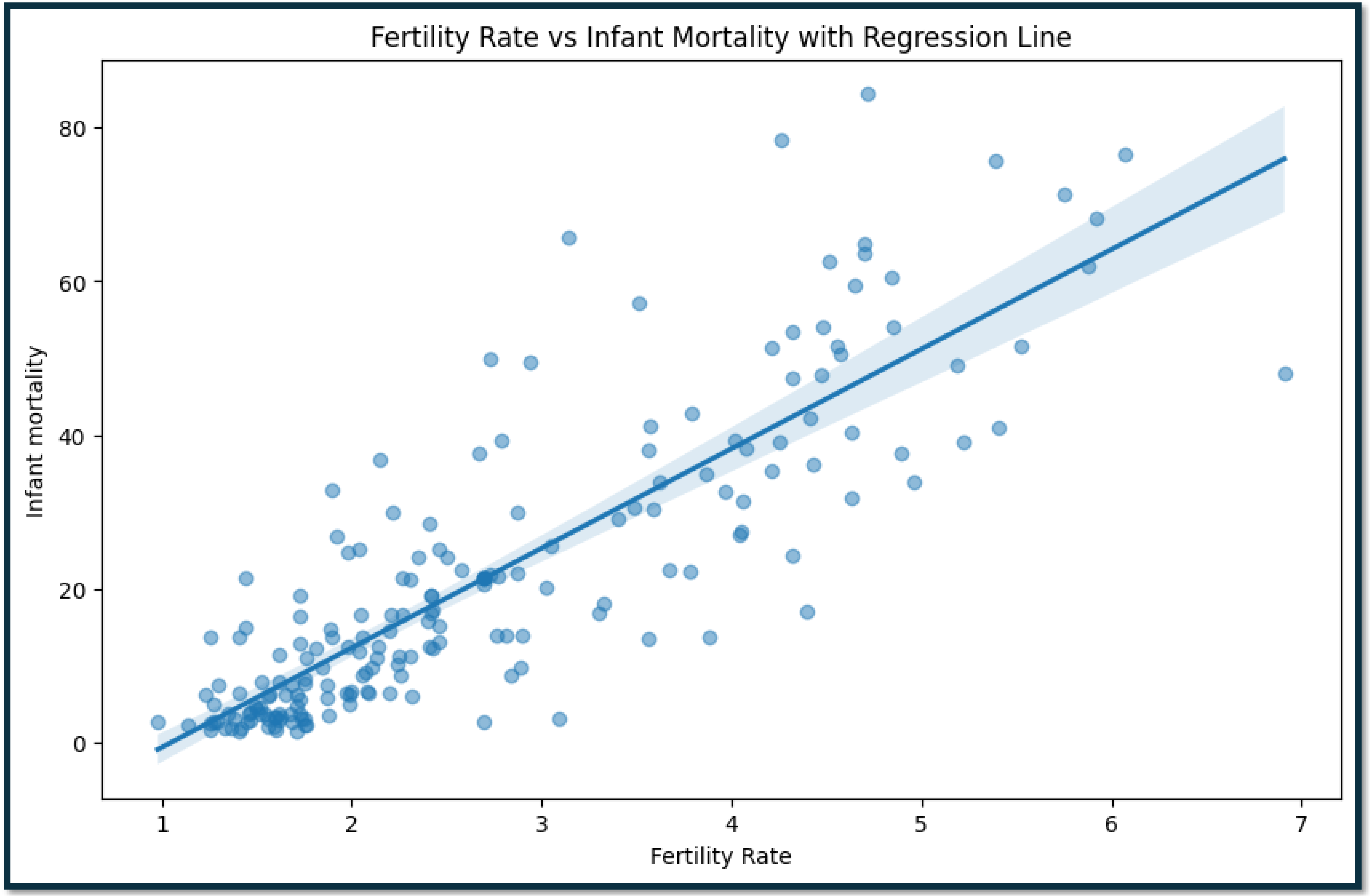
- The image shows a correlation heatmap depicting the relationships between various socioeconomic and health-related variables.
- Each cell in the heatmap represents the correlation coefficient between the pair of variables, with color coding indicating the strength and direction of the correlation.

Histogram



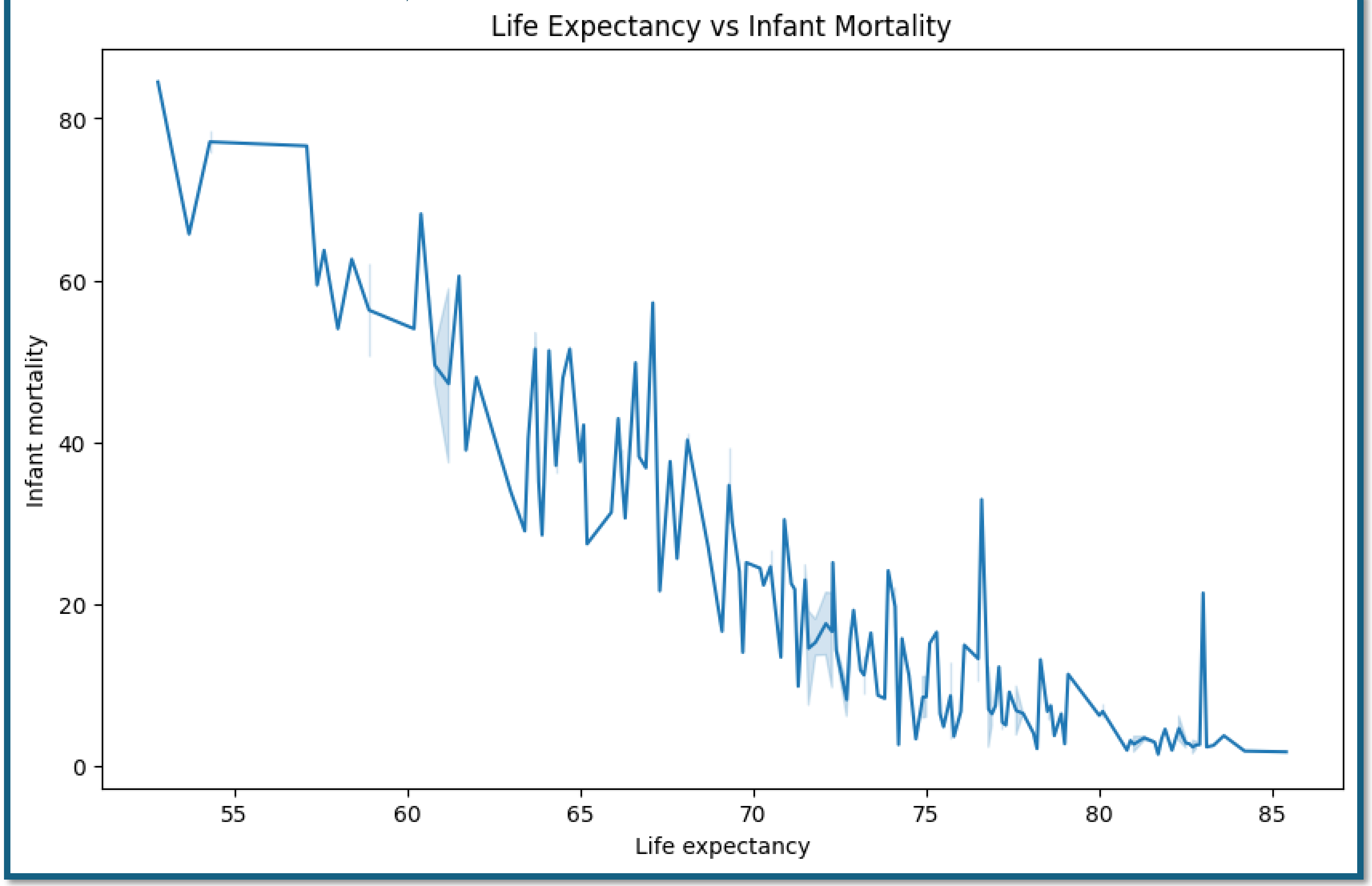
The histogram shows the distribution of infant mortality rates. The data is skewed to the right, with most countries having a lower infant mortality rate, and only a few countries showing higher rates above 40.

Scatter plot



The scatter plot shows a positive correlation between fertility rate and infant mortality, with a regression line indicating that higher fertility rates are associated with higher infant mortality.

Line plot



The line plot shows an inverse relationship between life expectancy and infant mortality, with infant mortality decreasing as life expectancy increases.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Infant mortality   R-squared:                       0.871
Model:                             OLS   Adj. R-squared:                  0.851
Method:                  Least Squares   F-statistic:                     43.19
Date:                 Wed, 06 Nov 2024   Prob (F-statistic):           2.45e-49
Time:                         15:02:48   Log-Likelihood:                -508.77
No. Observations:                  156   AIC:                             1062.
Df Residuals:                      134   BIC:                             1129.
Df Model:                           21
Covariance Type:             nonrobust
===============================================================================================================
                                                 coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------------------------
Density
(P/Km2)                                   2.536e-06      0.001      0.003      0.997      -0.002       0.002
Agricultural Land( %)                            0.0215      0.028      0.770      0.442      -0.034       0.077
Land Area(Km2)                                9.012e-08   3.54e-07      0.254      0.800   -6.11e-07    7.91e-07
Birth Rate                                       0.7250      0.422      1.719      0.088      -0.109       1.559
Co2-Emissions                                -3.101e-06   3.19e-06     -0.973      0.332    -9.4e-06     3.2e-06
CPI                                             -0.0035      0.012     -0.285      0.776      -0.028       0.021
CPI Change (%)                                   0.0527      0.127      0.415      0.679      -0.198       0.304
Fertility Rate                                  -0.5298      3.108     -0.170      0.865      -6.677       5.617
Gasoline Price                                  -0.9058      1.865     -0.486      0.628      -4.594       2.783
GDP                                           6.693e-13    6.91e-13      0.969      0.334   -6.97e-13    2.04e-12
Gross primary education enrollment (%)          -0.0042      0.052     -0.080      0.936      -0.108       0.100
Gross tertiary education enrollment (%)         -0.0906      0.031     -2.889      0.005      -0.153      -0.029
Life expectancy                                 -0.0675      0.095     -0.713      0.477      -0.255       0.120
Maternal mortality ratio                         0.0348      0.005      6.620      0.000       0.024       0.045
Out of pocket health expenditure                 0.1370      0.037      3.672      0.000       0.063       0.211
Physicians per thousand                         -0.4557      0.601     -0.759      0.449      -1.644       0.732
Population                                    7.582e-09    1.67e-08      0.454      0.651   -2.55e-08    4.06e-08
Population: Labor force participation (%)        0.0622      0.069      0.902      0.369      -0.074       0.199
Tax revenue (%)                                  0.0577      0.102      0.567      0.571      -0.144       0.259
Total tax rate                                   0.0288      0.030      0.957      0.340      -0.031       0.088
Unemployment rate                                0.1813      0.144      1.260      0.210      -0.103       0.466
Urban_population                              5.272e-09    5.04e-08      0.105      0.917   -9.43e-08    1.05e-07
==============================================================================
Omnibus:                       35.217   Durbin-Watson:                   1.938
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               64.507
Skew:                           1.060   Prob(JB):                     9.83e-15
Kurtosis:                       5.331   Cond. No.                     1.41e+13
==============================================================================
```

SIGNIFICANCE TEST

Fitting ordinary least squares model with all the features: Leading Current Reactive Power(LeRP) has Greatest P-value which is insignificant

SIGNIFICANCE TEST

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Infant mortality   R-squared:                       0.825
Model:                             OLS   Adj. R-squared:                  0.805
Method:                  Least Squares   F-statistic:                     41.04
Date:                 Wed, 06 Nov 2024   Prob (F-statistic):           1.34e-44
Time:                         15:02:48   Log-Likelihood:                -532.59
No. Observations:                  156   AIC:                             1099.
Df Residuals:                      139   BIC:                             1151.
Df Model:                           16
Covariance Type:             nonrobust
===============================================================================================================
                                               coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------------------------
Density
(P/Km2)                                0.0004      0.001      0.434      0.665      -0.001      0.002
Agricultural Land( %)                        0.0395      0.031      1.268      0.207      -0.022       0.101
Land Area(Km2)                            1.433e-07   3.88e-07      0.370      0.712   -6.23e-07     9.1e-07
CPI                                          0.0026      0.003      0.824      0.411      -0.004       0.009
Fertility Rate                               7.6369      0.807      9.468      0.000       6.042       9.232
Gasoline Price                               0.9851      2.086      0.472      0.637      -3.138       5.109
GDP                                      -6.273e-14   3.92e-13     -0.160      0.873   -8.38e-13    7.12e-13
Gross primary education enrollment (%)       0.0210      0.057      0.366      0.715      -0.093       0.135
Gross tertiary education enrollment (%)     -0.1129      0.035     -3.252      0.001      -0.182      -0.044
Life expectancy                             -0.3353      0.097     -3.459      0.001      -0.527      -0.144
Out of pocket health expenditure             0.2117      0.039      5.466      0.000       0.135       0.288
Physicians per thousand                     -0.5107      0.652     -0.784      0.435      -1.799       0.778
Population                                1.832e-09   5.54e-09      0.331      0.741   -9.12e-09    1.28e-08
Population: Labor force participation (%)    0.1996      0.073      2.729      0.007       0.055       0.344
Tax revenue (%)                              0.0409      0.115      0.357      0.722      -0.186       0.267
Total tax rate                               0.0474      0.033      1.426      0.156      -0.018       0.113
Unemployment rate                            0.3562      0.160      2.233      0.027       0.041       0.672
==============================================================================
Omnibus:                       22.394   Durbin-Watson:                   1.837
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               30.289
Skew:                           0.826   Prob(JB):                     2.65e-07
Kurtosis:                       4.390   Cond. No.                     8.20e+12
==============================================================================
```

Removing Insignificant variables: After removing variable having greatest P-value (greater than 0.05)

Multicollinearity Check

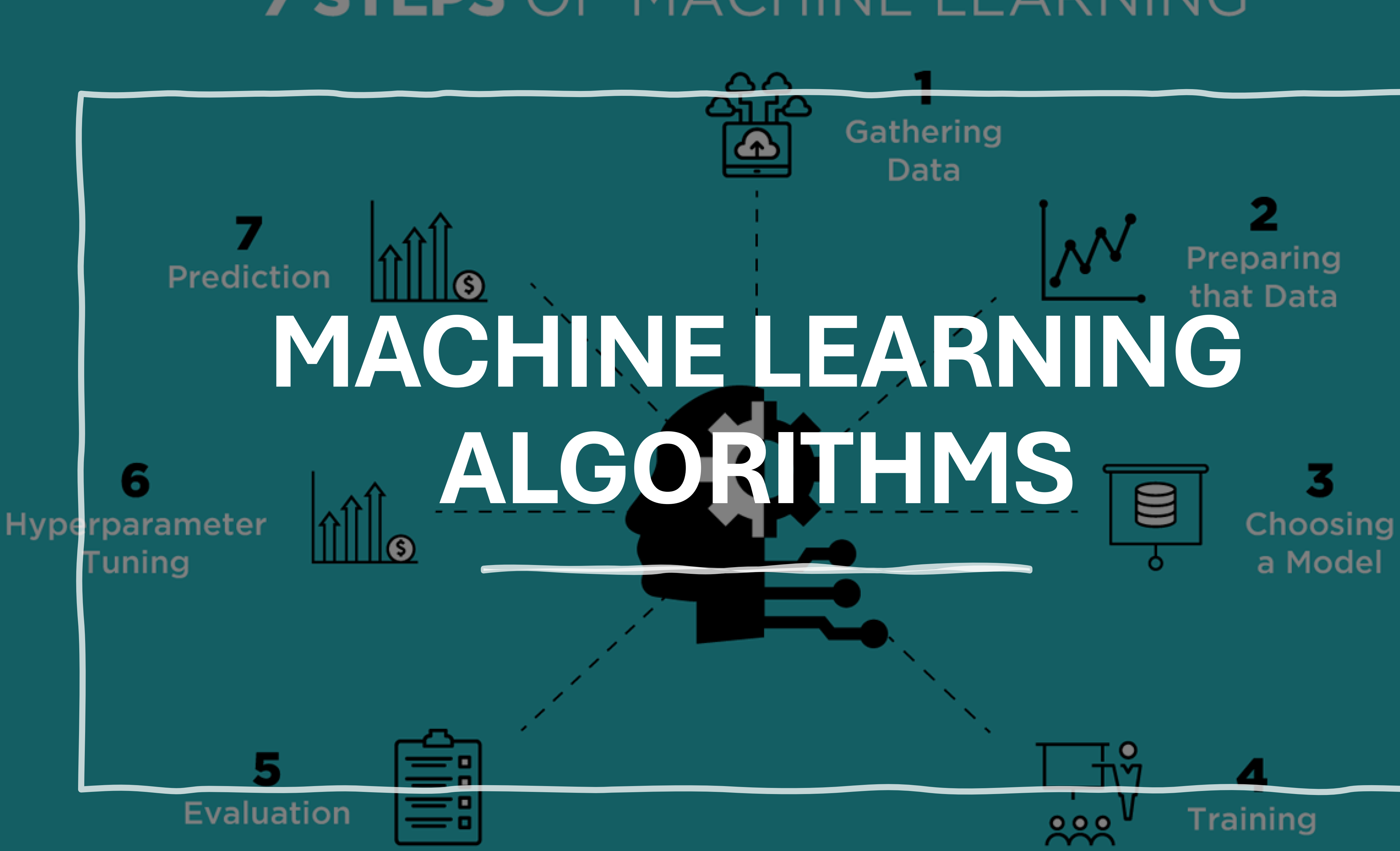
Variables with the greatest variance inflation factor (VIF >3) were removed

	variables	VIF
0	Density\n(P/Km2)	1.1
1	Agricultural Land(%)	1.3
2	Land Area(Km2)	1.9
3	Birth Rate	52.7
4	Co2-Emissions	28.6
5	CPI	23.8
6	CPI Change (%)	23.9
7	Fertility Rate	45.8
8	Gasoline Price	1.5
9	GDP	9.6
10	Gross primary education enrollment (%)	1.2
11	Gross tertiary education enrollment (%)	2.8
12	Life expectancy	1.5
13	Maternal mortality ratio	3.7
14	Out of pocket health expenditure	1.6
15	Physicians per thousand	3.1
16	Population	24.2
17	Population: Labor force participation (%)	1.6
18	Tax revenue (%)	1.4
19	Total tax rate	1.3
20	Unemployment rate	1.4
21	Urban_population	57.8

	variables	VIF
0	Density\n(P/Km2)	1.1
1	Agricultural Land(%)	1.3
2	Land Area(Km2)	1.9
3	Birth Rate	52.6
4	Co2-Emissions	16.5
5	CPI	23.8
6	CPI Change (%)	23.9
7	Fertility Rate	45.8
8	Gasoline Price	1.5
9	GDP	9.6
10	Gross primary education enrollment (%)	1.2
11	Gross tertiary education enrollment (%)	2.8
12	Life expectancy	1.5
13	Maternal mortality ratio	3.7
14	Out of pocket health expenditure	1.6
15	Physicians per thousand	3.1
16	Population	4.5
17	Population: Labor force participation (%)	1.6
18	Tax revenue (%)	1.4
19	Total tax rate	1.3
20	Unemployment rate	1.4

	variables	VIF
0	Density\n(P/Km2)	1.1
1	Agricultural Land(%)	1.2
2	Land Area(Km2)	1.7
3	CPI	1.1
4	Fertility Rate	2.4
5	Gasoline Price	1.4
6	GDP	2.4
7	Gross primary education enrollment (%)	1.1
8	Gross tertiary education enrollment (%)	2.7
9	Life expectancy	1.2
10	Out of pocket health expenditure	1.3
11	Physicians per thousand	2.8
12	Population	2.0
13	Population: Labor force participation (%)	1.3
14	Tax revenue (%)	1.4
15	Total tax rate	1.2
16	Unemployment rate	1.3

MACHINE LEARNING ALGORITHMS



Machine Learning Algorithms

- Multiple Linear Regression
- K-Nearest Neighbors(KNN)
- Decision Tree
- Random Forest
- Bagging
- Boosting
- Support Vector Machine(SVM)

80:20 Train-Test Split Ratio

Algorithms	Model 1 (r^2)	Model 2 (r^2)	Model 1 (MAE)	Model 2 (MAE)
Linear Regression	0.872	0.856	2.256	2.369
KNN	-0.152	-0.152	15.695	15.695
Decision Tree Regressor	0.798	0.808	5.783	5.861
Random Forest	0.890	0.800	4.114	5.592
XG Boost	0.840	0.80	4.924	5.59
Ada Boost	-0.041	-0.041	5.088	5.104
SVM	-0.041	0.866	12.862	5.104
ANN	0.892	0.914	4.215	3.537

***Model 1:** With all features* | ***Model 2:** After removing multi-collinear variables*

R^2 Score:
The **R^2 score** measures how well the independent variables (predictors) explain the variation in the dependent variable (target). It tells you how well your regression model fits the data.

Mean Absolute Error (MAE):
The **MAE** measures the average magnitude of the errors in a regression model. It calculates how far the predicted values are, on average, from the actual values.

75:25 Train-Test Split Ratio

Algorithms	Model 1 (r^2)	Model 2 (r^2)	Model 1 (MAE)	Model 2 (MAE)
Linear Regression	0.849	0.862	2.346	2.383
KNN	0.004	0.004	15.199	15.199
Decision Tree Regressor	0.844	0.823	5.436	5.659
Random Forest	0.892	0.890	4.307	4.503
XG Boost	0.884	0.857	4.777	5.357
Ada Boost	-0.112	-0.111	4.833	5.251
SVM	-0.112	-0.111	14.267	14.267
ANN	0.884	0.861	4.795	4.978

***Model 1:** With all features* | ***Model 2:** After removing multi-collinear variables*

70:30 Train-Test Split Ratio

Algorithms	Model 1 (r^2)	Model 2 (r^2)	Model 1 (MAE)	Model 2 (MAE)
Linear Regression	0.891	0.873	2.241	2.395
KNN	-0.042	-0.042	15.834	15.834
Decision Tree Regressor	0.759	0.689	6.312	7.451
Random Forest	0.889	0.875	4.565	4.999
XG Boost	0.900	0.861	4.667	5.095
Ada Boost	-0.149	-0.149	5.519	5.421
SVM	-0.149	-0.149	15.134	15.133
ANN	0.867	0.859	4.797	5.221

Model 1: *With all features |* ***Model 2:*** *After removing multi-collinear variables*

60:40 Train-Test Split Ratio

Algorithms	Model 1 (r^2)	Model 2 (r^2)	Model 1 (MAE)	Model 2 (MAE)
Linear Regression	0.856	0.788	2.321	2.539
KNN	-0.011	-0.011	14.997	14.997
Decision Tree Regressor	0.766	0.802	6.157	5.784
Random Forest	0.877	0.872	4.467	4.750
XG Boost	0.849	0.844	4.910	5.152
Ada Boost	-0.131	-0.131	5.163	5.449
SVM	-0.131	0.851	5.449	14.272
ANN	0.682	0.677	6.343	0.084

Model 1: *With all features |* ***Model 2:*** *After removing multi-collinear variables*

Algorithm Comparision (80:20)

Model 1

Algorithms	R^ 2 score
Linear Regression	0.872
KNN	-0.152
Decision Tree Regressor	0.798
Random Forest	0.890
XG Boost	0.840
Ada Boost	-0.041
SVM	-0.041
ANN	0.892

Model 2

Algorithms	R^ 2 score
Linear Regression	0.856
KNN	-0.152
Decision Tree Regressor	0.808
Random Forest	0.800
XG Boost	0.80
Ada Boost	-0.041
SVM	0.866
ANN	0.914

Summary

This project explores global country-level macroeconomic, demographic, and social indicators to analyze their impact on infant mortality rates. Using a dataset that includes variables such as GDP, life expectancy, fertility rate, birth rate, and healthcare factors, the goal is to understand the underlying relationships that influence infant mortality across different regions.

Statistical analysis and machine learning models identify key predictors of infant mortality, with visualizations like line plots, scatter plots, and heatmaps highlighting relationships and trends to inform policy interventions.

The Artificial Neural Network (ANN) model proved to be the most effective for predicting infant mortality in both Model 1 and Model 2. It demonstrated superior accuracy by effectively capturing complex relationships between factors such as GDP and healthcare spending. The ANN outperformed other models, providing reliable predictions without overfitting. These results offer valuable insights for formulating policies to reduce infant mortality rates.

Future Scope

- ❖ **Better Models**: Using more advanced machine learning models could improve predictions for infant mortality, making the results more accurate.

❖**Regional Focus**: Analyzing the data separately for different regions could help understand local factors that affect infant mortality better.

❖**Looking at Changes Over Time**: Including historical data could show how infant mortality rates have changed over the years and help identify patterns.

❖**Adding More Data**: Including extra data, like healthcare quality or environmental factors, could give a clearer picture of what impacts infant mortality.

❖**Testing Policies**: Developing a model that tests the effects of policies (like improving healthcare or education) on infant mortality could guide decision-making.

Work Distribution

Team	Work
Lokesh Kumar	Collect information about Global data & Literature Review
K.Manohar	Data Preprocessing
A.Deepak Mudiraj	Exploratory Data Analysis
N.Keerthana Reddy	Implement Machine Learning Algorithms

Thank You

Group 5
N.Keerthana Reddy
A.Deepak Mudiraj
Lokesh Kumar
K.Manohar

APPENDIX.

Loading the Data

```
df = pd.read_csv("/content/world-data-2023.csv")
```

```
[ ] df.head()
```

	Country	Density\n(P/Km2)	Abbreviation	Agricultural Land(%)	Land Area(Km2)	Armed Forces size	Birth Rate	Calling Code	Capital/Major City	Co2-Emissions	...	Out of pocket health expenditure	Physicians per thousand	Population
0	Afghanistan	60	AF	58.10%	652,230	323,000	32.49	93.0	Kabul	8,672	...	78.40%	0.28	38,041,754
1	Albania	105	AL	43.10%	28,748	9,000	11.78	355.0	Tirana	4,536	...	56.90%	1.20	2,854,191
2	Algeria	18	DZ	17.40%	2,381,741	317,000	24.28	213.0	Algiers	150,006	...	28.10%	1.72	43,053,054
3	Andorra	164	AD	40.00%	468	NaN	7.20	376.0	Andorra la Vella	469	...	36.40%	3.33	77,142
4	Angola	26	AO	47.50%	1,246,700	117,000	40.73	244.0	Luanda	34,693	...	33.40%	0.21	31,825,295

5 rows × 35 columns

Population: Labor force participation (%)	Tax revenue (%)	Total tax rate	Unemployment rate	Urban_population	Latitude	Longitude
48.90%	9.30%	71.40%	11.12%	9,797,273	33.939110	67.709953
55.70%	18.60%	36.60%	12.33%	1,747,593	41.153332	20.168331
41.20%	37.20%	66.10%	11.70%	31,510,100	28.033886	1.659626
NaN	NaN	NaN	NaN	67,873	42.506285	1.521801
77.50%	9.20%	49.10%	6.89%	21,061,025	-11.202692	17.873887

Null Values

```
[ ] df.isna().sum()
```

	0
Country	0
Density\n(P/Km2)	0
Abbreviation	7
Agricultural Land(%)	7
Land Area(Km2)	1
Armed Forces size	24
Birth Rate	6
Calling Code	1
Capital/Major City	3
Co2-Emissions	7
CPI	17
CPI Change (%)	16
Currency-Code	15
Fertility Rate	7
Forested Area (%)	7
Gasoline Price	20
GDP	2
Gross primary education enrollment (%)	7
Gross tertiary education enrollment (%)	12
Infant mortality	6
Largest city	6
Life expectancy	8
Maternal mortality ratio	14
Minimum wage	45
Official language	5
Out of pocket health expenditure	7
Physicians per thousand	7
Population	1
Population: Labor force participation (%)	19
Tax revenue (%)	26
Total tax rate	12
Unemployment rate	19
Urban_population	5
Latitude	1
Longitude	1

dtype: int64

Null Values

```
# Fill numeric columns with their mean
numeric_cols = df_cleaned.select_dtypes(include=['number']).columns
df_cleaned[numeric_cols] = df_cleaned[numeric_cols].fillna(df_cleaned[numeric_cols].mean())

# Fill non-numeric (categorical) columns with their mode
non_numeric_cols = df_cleaned.select_dtypes(exclude=['number']).columns
df_cleaned[non_numeric_cols] = df_cleaned[non_numeric_cols].fillna(df_cleaned[non_numeric_cols].mode().iloc[0])

# Display the count of missing values for each column after filling
print("Missing values after filling:")
print(df_cleaned.isna().sum())
```

```
Missing values after filling:
Density\n(P/Km2)                             0
Agricultural Land( %)                        0
Land Area(Km2)                               0
Birth Rate                                   0
Co2-Emissions                                0
CPI                                          0
CPI Change (%)                               0
Fertility Rate                               0
Gasoline Price                               0
GDP                                          0
Gross primary education enrollment (%)       0
Gross tertiary education enrollment (%)      0
Infant mortality                             0
Life expectancy                              0
Maternal mortality ratio                     0
Out of pocket health expenditure             0
Physicians per thousand                      0
Population                                   0
Population: Labor force participation (%)    0
Tax revenue (%)                              0
Total tax rate                               0
Unemployment rate                            0
Urban_population                             0
dtype: int64
```

Checking For the Data type

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 35 columns):
 #   Column                                     Non-Null Count  Dtype
---  ------                                     --------------  -----
 0   Country                                    195 non-null    object
 1   Density
(P/Km2)                     195 non-null    object
 2   Abbreviation                               188 non-null    object
 3   Agricultural Land( %)                      188 non-null    object
 4   Land Area(Km2)                             194 non-null    object
 5   Armed Forces size                          171 non-null    object
 6   Birth Rate                                 189 non-null    float64
 7   Calling Code                               194 non-null    float64
 8   Capital/Major City                         192 non-null    object
 9   Co2-Emissions                              188 non-null    object
 10  CPI                                        178 non-null    object
 11  CPI Change (%)                             179 non-null    object
 12  Currency-Code                              180 non-null    object
 13  Fertility Rate                             188 non-null    float64
 14  Forested Area (%)                          188 non-null    object
```

```
 15  Gasoline Price                             175 non-null    object
 16  GDP                                        193 non-null    object
 17  Gross primary education enrollment (%)     188 non-null    object
 18  Gross tertiary education enrollment (%)    183 non-null    object
 19  Infant mortality                           189 non-null    float64
 20  Largest city                               189 non-null    object
 21  Life expectancy                            187 non-null    float64
 22  Maternal mortality ratio                   181 non-null    float64
 23  Minimum wage                               150 non-null    object
 24  Official language                          190 non-null    object
 25  Out of pocket health expenditure           188 non-null    object
 26  Physicians per thousand                    188 non-null    float64
 27  Population                                 194 non-null    object
 28  Population: Labor force participation (%)  176 non-null    object
 29  Tax revenue (%)                            169 non-null    object
 30  Total tax rate                             183 non-null    object
 31  Unemployment rate                          176 non-null    object
 32  Urban_population                           190 non-null    object
 33  Latitude                                   194 non-null    float64
 34  Longitude                                  194 non-null    float64
dtypes: float64(9), object(26)
memory usage: 53.4+ KB
```

Replacing the symbols

```
df_cleaned = df_cleaned.replace({',': '', '\$': '', '%': ''}, regex=True).apply(pd.to_numeric, errors='coerce')

# Drop any columns that are still non-numeric after conversion
df_cleaned = df_cleaned.dropna(axis=1, how='all')

# Display the cleaned DataFrame
print("\nCleaned DataFrame:")
print(df_cleaned.head())
```

```
Cleaned DataFrame:
   Density\n(P/Km2)  Agricultural Land( %)  Land Area(Km2)  Birth Rate  \
0                60                   58.1          652230       32.49
1               105                   43.1           28748       11.78
2                18                   17.4         2381741       24.28
3               164                   40.0             468        7.20
4                26                   47.5         1246700       40.73

   Co2-Emissions     CPI  CPI Change (%)  Fertility Rate  Gasoline Price  \
0           8672  149.90             2.3            4.47            0.70
1           4536  119.05             1.4            1.62            1.36
2         150006  151.36             2.0            3.02            0.28
3            469  106.58             1.8            1.27            1.51
4          34693  261.73            17.1            5.52            0.97

            GDP  ...  Life expectancy  Maternal mortality ratio  \
0   19101353833  ...        64.500000                638.000000
1   15278077447  ...        78.500000                 15.000000
2  169988236398  ...        76.700000                112.000000
3    3154057987  ...        72.279679                160.392265
4   94635415870  ...        60.800000                241.000000

   Out of pocket health expenditure  Physicians per thousand  Population  \
0                              78.4                     0.28    38041754
1                              56.9                     1.20     2854191
2                              28.1                     1.72    43053054
3                              36.4                     3.33       77142
4                              33.4                     0.21    31825295
```

Dropping The columns

```
# List of columns to drop
columns_to_drop = [
     'Country', 'Abbreviation', 'Armed Forces size',
     'Calling Code', 'Capital/Major City', 'Currency-Code',
     'Forested Area (%)',
     'Largest city','Minimum wage',
      'Official language',
     'Latitude', 'Longitude'
]

# Dropping the specified columns
df_cleaned = df.drop(columns=columns_to_drop)

# Display the cleaned DataFrame
print(df_cleaned.head())
```

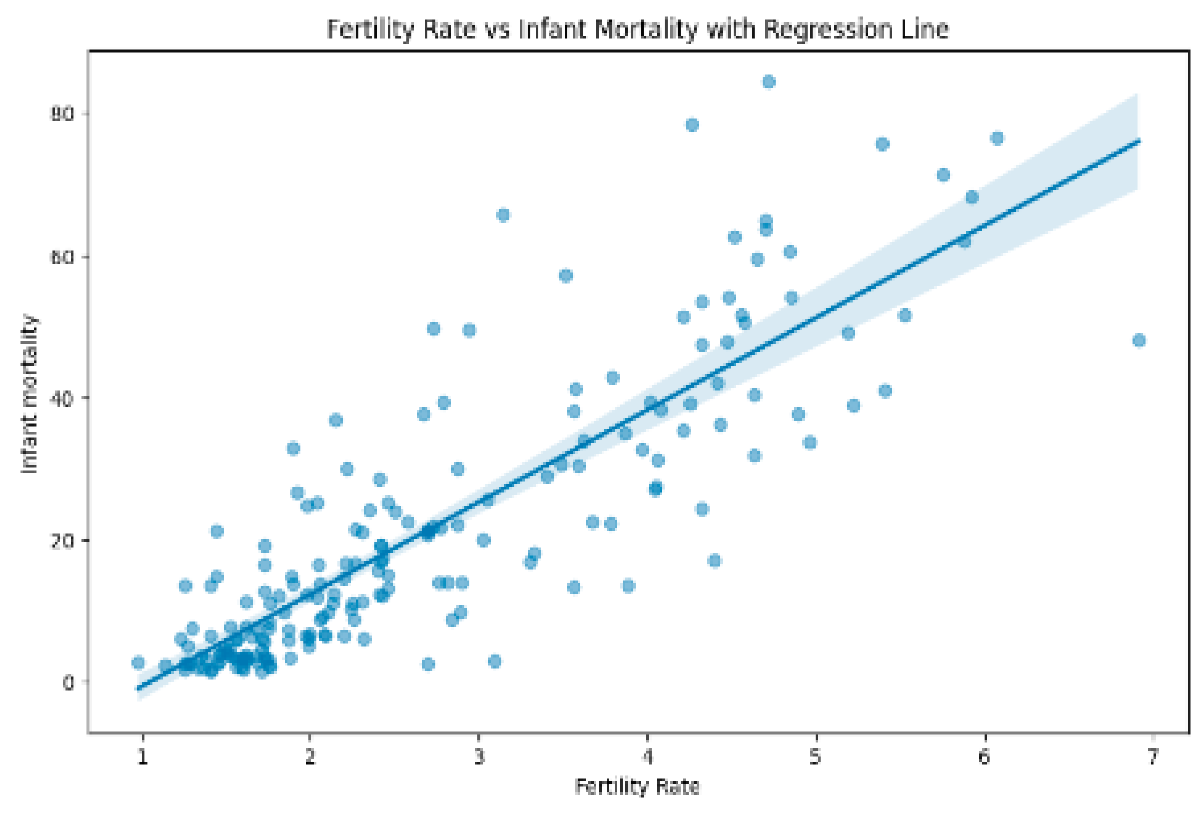
Dropping The columns

```
  Density\n(P/Km2) Agricultural Land( %) Land Area(Km2)  Birth Rate  \
0               60                58.10%        652,230       32.49   
1              105                43.10%         28,748       11.78   
2               18                17.40%      2,381,741       24.28   
3              164                40.00%            468        7.20   
4               26                47.50%      1,246,700       40.73   

  Co2-Emissions     CPI CPI Change (%)  Fertility Rate Gasoline Price
0         8,672   149.9          2.30%            4.47          $0.70
1         4,536  119.05          1.40%            1.62          $1.36
2       150,006  151.36          2.00%            3.02          $0.28
3           469     NaN            NaN            1.27          $1.51
4        34,693  261.73         17.10%            5.52          $0.97

                GDP  ... Life expectancy Maternal mortality ratio  \
0   $19,101,353,833  ...            64.5                    638.0   
1   $15,278,077,447  ...            78.5                     15.0   
2  $169,988,236,398  ...            76.7                    112.0   
3    $3,154,057,987  ...             NaN                      NaN   
4   $94,635,415,870  ...            60.8                    241.0   
```

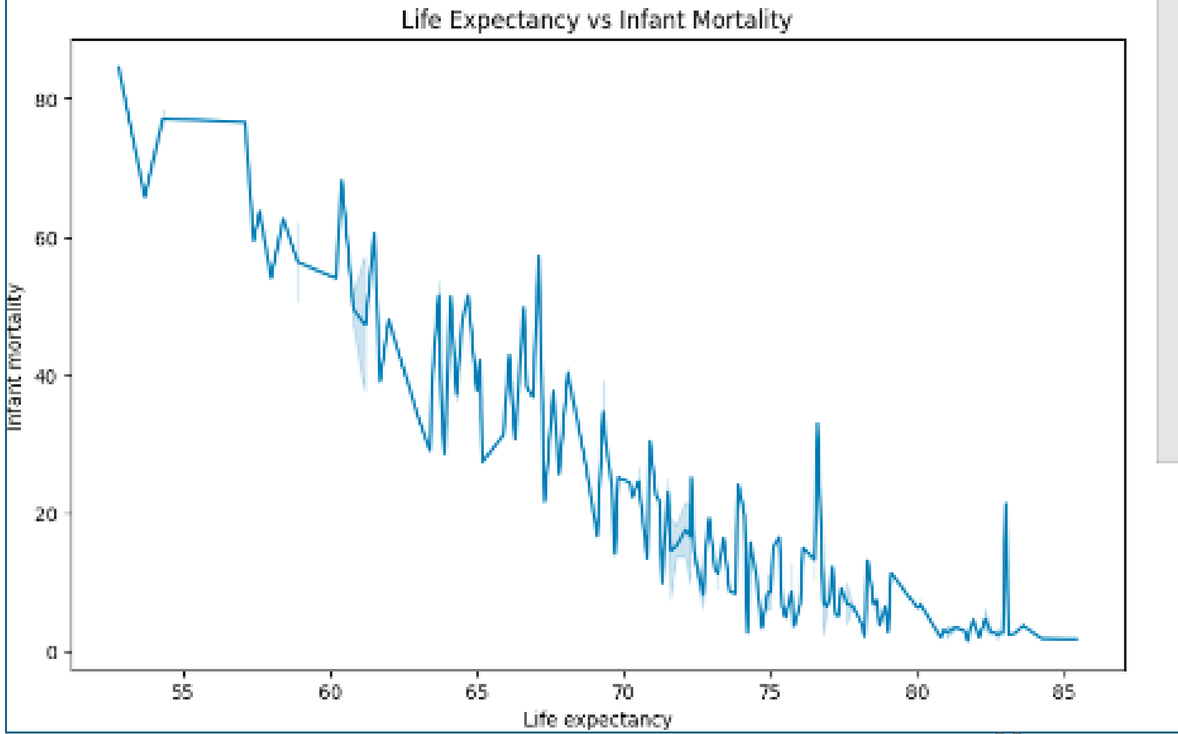
```
  Out of pocket health expenditure  Physicians per thousand  Population  \
0                           78.40%                     0.28  38,041,754   
1                           56.90%                     1.20   2,854,191   
2                           28.10%                     1.72  43,053,054   
3                           36.40%                     3.33      77,142   
4                           33.40%                     0.21  31,825,295   

  Population: Labor force participation (%) Tax revenue (%) Total tax rate  \
0                                    48.90%           9.30%         71.40%   
1                                    55.70%          18.60%         36.60%   
2                                    41.20%          37.20%         66.10%   
3                                       NaN             NaN            NaN   
4                                    77.50%           9.20%         49.10%   

  Unemployment rate Urban_population
0            11.12%        9,797,273
1            12.33%        1,747,593
2            11.70%       31,510,100
3               NaN           67,873
4             6.89%       21,061,025
```

Data Visualization

```python
plt.figure(figsize=(10, 6))
sns.regplot(x='Fertility Rate', y='Infant mortality', data=df_cleaned, scatter_kws={'alpha':0.5})
plt.title('Fertility Rate vs Infant Mortality with Regression Line')
plt.show()
```




```python
plt.figure(figsize=(10, 6))
sns.lineplot(x='Life expectancy', y='Infant mortality', data=df_cleaned)
plt.title('Life Expectancy vs Infant Mortality')
plt.show()
```



Multicollinearity Check

```
[ ] import warnings
    warnings.filterwarnings("ignore")
```

```
[ ] # Split Data into Training and Test Sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

```
[ ] # Import library for VIF
    from statsmodels.stats.outliers_influence import variance_inflation_factor

    def calc_vif(X):

        # Calculating VIF
        vif = pd.DataFrame()
        vif["variables"] = X.columns
        vif["VIF"] = [variance_inflation_factor(X.values, i).round(1) for i in range(X.shape[1])]

        return(vif)

    calc_vif(X)
```

Multicollinearity Check

	variables	VIF
0	Density\n(P/Km2)	1.4
1	Agricultural Land(%)	1.2
2	Land Area(Km2)	1.8
3	Birth Rate	50.6
4	Co2-Emissions	25.1
5	CPI	5.9
6	CPI Change (%)	5.6
7	Fertility Rate	41.5
8	Gasoline Price	1.5
9	GDP	9.0
10	Gross primary education enrollment (%)	1.2
11	Gross tertiary education enrollment (%)	2.9
12	Life expectancy	1.3
13	Maternal mortality ratio	3.6

	variables	VIF
0	Density\n(P/Km2)	1.1
1	Agricultural Land(%)	1.3
2	Land Area(Km2)	1.9
3	Birth Rate	52.6
4	Co2-Emissions	16.5
5	CPI	23.8
6	CPI Change (%)	23.9
7	Fertility Rate	45.8
8	Gasoline Price	1.5
9	GDP	9.6
10	Gross primary education enrollment (%)	1.2
11	Gross tertiary education enrollment (%)	2.8
12	Life expectancy	1.5
13	Maternal mortality ratio	3.7
14	Out of pocket health expenditure	1.6
15	Physicians per thousand	3.1
16	Population	4.5
17	Population: Labor force participation (%)	1.6
18	Tax revenue (%)	1.4
19	Total tax rate	1.3
20	Unemployment rate	1.4

	variables	VIF
0	Density\n(P/Km2)	1.1
1	Agricultural Land(%)	1.2
2	Land Area(Km2)	1.7
3	CPI	1.1
4	Fertility Rate	2.4
5	Gasoline Price	1.4
6	GDP	2.4
7	Gross primary education enrollment (%)	1.1
8	Gross tertiary education enrollment (%)	2.7
9	Life expectancy	1.2
10	Out of pocket health expenditure	1.3
11	Physicians per thousand	2.8
12	Population	2.0
13	Population: Labor force participation (%)	1.3
14	Tax revenue (%)	1.4
15	Total tax rate	1.2
16	Unemployment rate	1.3

Ordinary Least Square

```python
import statsmodels.api as sm

model = sm.OLS(y, X).fit()

# Print the summary to get p-values
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       Infant mortality   R-squared:                       0.825
Model:                            OLS   Adj. R-squared:                  0.805
Method:                 Least Squares   F-statistic:                     41.04
Date:                Wed, 06 Nov 2024   Prob (F-statistic):           1.34e-44
Time:                        15:02:48   Log-Likelihood:                -532.59
No. Observations:                 156   AIC:                             1099.
Df Residuals:                     139   BIC:                             1151.
Df Model:                          16
Covariance Type:            nonrobust
==============================================================================================================
                                                coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------------------------------
Density
(P/Km2)                                  0.0004      0.001      0.434      0.665      -0.001      0.002
Agricultural Land( %)                         0.0395      0.031      1.268      0.207      -0.022       0.101
Land Area(Km2)                             1.433e-07   3.88e-07      0.370      0.712   -6.23e-07     9.1e-07
CPI                                           0.0026      0.003      0.824      0.411      -0.004       0.009
Fertility Rate                                7.6369      0.807      9.468      0.000       6.042       9.232
Gasoline Price                                0.9851      2.086      0.472      0.637      -3.138       5.109
GDP                                       -6.273e-14   3.92e-13     -0.160      0.873   -8.38e-13    7.12e-13
Gross primary education enrollment (%)        0.0210      0.057      0.366      0.715      -0.093       0.135
Gross tertiary education enrollment (%)      -0.1129      0.035     -3.252      0.001      -0.182      -0.044
Life expectancy                              -0.3353      0.097     -3.459      0.001      -0.527      -0.144
Out of pocket health expenditure              0.2117      0.039      5.466      0.000       0.135       0.288
Physicians per thousand                      -0.5107      0.652     -0.784      0.435      -1.799       0.778
Population                                 1.832e-09   5.54e-09      0.331      0.741   -9.12e-09    1.28e-08
Population: Labor force participation (%)     0.1996      0.073      2.729      0.007       0.055       0.344
Tax revenue (%)                               0.0409      0.115      0.357      0.722      -0.186       0.267
Total tax rate                                0.0474      0.033      1.426      0.156      -0.018       0.113
Unemployment rate                             0.3562      0.160      2.233      0.027       0.041       0.672
==============================================================================
Omnibus:                       22.394   Durbin-Watson:                   1.837
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               30.289
Skew:                           0.826   Prob(JB):                     2.65e-07
Kurtosis:                       4.390   Cond. No.                     8.20e+12
==============================================================================
```

Linear Regression

```
[ ] from sklearn.linear_model import LinearRegression
    from sklearn.metrics import confusion_matrix
    from sklearn.metrics import accuracy_score
    from sklearn.metrics import classification_report
```

80-20

```
[ ] linreg = LinearRegression()
    linreg.fit(X_train1_nomulti, y_train1_nomulti)
    y_pred1 = linreg.predict(X_test1_nomulti)
    print(np.sqrt(metrics.mean_absolute_error(y_test1_nomulti, y_pred1)))

    2.369363817805726

    from sklearn.metrics import r2_score
    r2_score(y_test1_nomulti,y_pred1)

    0.8562569235847044
```

75-25

```
[ ] linreg = LinearRegression()
    linreg.fit(X_train2_nomulti, y_train2_nomulti)
    y_pred2 = linreg.predict(X_test2_nomulti)
    print(np.sqrt(metrics.mean_absolute_error(y_test2_nomulti, y_pred2)))

    2.3831576897652957

    from sklearn.metrics import r2_score
    r2_score(y_test2_nomulti,y_pred2)

    0.8625554682820821
```

70-30

```
    linreg = LinearRegression()
    linreg.fit(X_train3_nomulti, y_train3_nomulti)
    y_pred3 = linreg.predict(X_test3_nomulti)
    print(np.sqrt(metrics.mean_absolute_error(y_test3_nomulti, y_pred3)))

    2.3950590456006586

[ ] from sklearn.metrics import r2_score
    r2_score(y_test3_nomulti,y_pred3)

    0.8735595477870415
```

60-40

```
    linreg = LinearRegression()
    linreg.fit(X_train4_nomulti, y_train4_nomulti)
    y_pred4 = linreg.predict(X_test4_nomulti)
    print(np.sqrt(metrics.mean_absolute_error(y_test4_nomulti, y_pred4)))

    2.539115603296015

[ ] from sklearn.metrics import r2_score
    r2_score(y_test4_nomulti,y_pred4)

    0.7884406041449054
```

K-Nearest Neighbors

```
[ ] from sklearn.neighbors import KNeighborsRegressor
```

```
[ ] model=KNeighborsRegressor(n_neighbors=5)
```

80-20

```
[ ] model.fit(X_train1_nomulti, y_train1_nomulti)
```

```
KNeighborsRegressor
KNeighborsRegressor()
```

```
[ ] y_pred1 = model.predict(X_test1_nomulti)
    y_pred1
```

```
array([29.08656085, 19.66      , 25.18      , 22.92      ,  9.12      ,
       25.97312169, 33.24      , 43.06      , 29.08656085, 15.68      ,
       10.46      , 20.2       , 41.92      , 13.26      , 11.8       ,
       20.2       , 14.66      , 18.44      , 20.2       , 33.34      ,
       23.42      , 29.08656085, 17.32      , 32.04      , 14.24      ,
        5.42      , 10.36      , 36.18      , 20.68      , 46.54      ,
       12.38      , 10.46      , 33.01312169, 20.4      , 12.38      ,
       48.48      , 11.86      , 29.08656085,  5.42      ])
```

R^2 error

```
[ ] from sklearn.metrics import r2_score
    r2_score(y_test1_nomulti,y_pred1)
```

```
-0.1524838918222453
```

Mean Absolute Error

```
[ ] from sklearn import metrics
    metrics.mean_absolute_error(y_test1_nomulti,y_pred1)
```

```
15.695376475376476
```

```
knn = pd.DataFrame({'Predicted':y_pred1,'Actual':y_test1_nomulti})
knn
```

	Predicted	Actual
132	29.086561	16.600000
16	19.660000	2.900000
18	25.180000	60.500000
51	22.920000	18.100000
164	9.120000	2.500000
145	25.973122	9.800000
11	33.240000	8.300000
27	43.060000	41.000000
176	29.086561	13.400000
118	15.680000	36.800000
28	10.460000	59.400000
170	20.200000	14.000000
56	41.920000	21.332804
116	13.260000	19.200000
103	11.800000	6.700000
31	20.200000	50.600000
59	14.660000	1.400000
40	18.440000	7.600000
136	20.200000	17.200000
95	33.340000	65.700000
35	23.420000	6.200000
107	29.086561	27.400000

Support Vector Machine

```
[ ] from sklearn.svm import SVR
```

```
model = SVR(kernel='rbf')
```

80-20

```
[ ] model.fit(X_train1_nomulti, y_train1_nomulti)
```

```
SVR
SVR()
```

```
[ ] y_pred1 = model.predict(X_test1_nomulti)
    y_pred1
```

```
array([15.64245242, 13.394925  , 15.59346725, 14.4486608 ,  8.95492719,
       15.63980709, 15.59893168, 15.63303307, 15.64187883, 15.37113382,
       15.43476572, 15.50130963, 15.63033892, 15.21000428, 14.17383466,
       15.50720491, 14.59811034, 15.42386631, 15.50940777, 15.63494183,
       14.5396159 , 15.64266863, 15.59626544, 15.58622962, 15.36780501,
       15.45142283, 14.85838838, 15.63659383, 15.30177063, 15.6296678 ,
       14.03465474, 15.43794359, 15.63919571, 13.62375911, 14.06413723,
       15.60197367, 14.69311065, 15.64326844, 15.44477781])
```

R^2 score

```
from sklearn.metrics import r2_score
r2_score(y_test1,y_pred1)
```

```
-0.04112710818970866
```

Mean Absolute Error

```
[ ] from sklearn import metrics
    metrics.mean_absolute_error(y_test1,y_pred1)
```

```
12.862058926026647
```

```
svm = pd.DataFrame({'Predicted':y_pred1,'Actual':y_test1_nomulti})
svm
```

	Predicted	Actual
132	15.642452	16.600000
16	13.394925	2.900000
18	15.593467	60.500000
51	14.448661	18.100000
164	8.954927	2.500000
145	15.639807	9.800000
11	15.598932	8.300000
27	15.633033	41.000000
176	15.641879	13.400000
118	15.371134	36.800000
28	15.434766	59.400000
170	15.501310	14.000000
56	15.630339	21.332804
116	15.210004	19.200000
103	14.173835	6.700000
31	15.507205	50.600000
59	14.598110	1.400000
40	15.423866	7.600000
136	15.509408	17.200000
95	15.634942	65.700000
35	14.539616	6.200000
107	15.642669	27.400000

Decision Tree

```
[ ] # Create Decision Tree classifer object
    reg1 = DecisionTreeRegressor()

    # Train Decision Tree Classifer
    reg1 = reg1.fit(X_train1,y_train1)
```

```
[ ] DecisionTreeRegressor?
```

```
[ ] #Predict the response for test dataset
    y_pred1 = reg1.predict(X_test1)
```

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

# Calculate regression metrics
print("Mean Absolute Error:", mean_absolute_error(y_test1, y_pred1))
print("R-squared:", r2_score(y_test1, y_pred1))
```

```
Mean Absolute Error: 5.998277031610365
R-squared: 0.7755995316044647
```

XG Boost

```
[ ] # Initialize the XGBoost Regressor models
    model1 = xgb.XGBRegressor()
```

80-20

```
[ ] # Fit the models on the training data
    train_model1 = model1.fit(X_train1, y_train1)
```

```
[ ] # Make predictions
    pred1 = train_model1.predict(X_test1)
```

```
# Calculate and print evaluation metrics
print("Model 1 XGBoost - Mean Absolute Error: %.5f" % mean_absolute_error(y_test1, pred1))
print("Model 1 XGBoost - R^2 Score: %.5f" % r2_score(y_test1, pred1))
```

```
Model 1 XGBoost - Mean Absolute Error: 4.92472
Model 1 XGBoost - R^2 Score: 0.84043
```

75-25

```
[ ] # Fit the models on the training data
    train_model1 = model1.fit(X_train2, y_train2)
```

```
[ ] # Make predictions
    pred1 = train_model1.predict(X_test2)
```

```
[ ] # Calculate and print evaluation metrics
    print("Model 1 XGBoost - Mean Absolute Error: %.5f" % mean_absolute_error(y_test2, pred1))
    print("Model 1 XGBoost - R^2 Score: %.5f" % r2_score(y_test2, pred1))
```

```
Model 1 XGBoost - Mean Absolute Error: 4.77716
Model 1 XGBoost - R^2 Score: 0.88414
```

70-30

+ Code + Text

```
[ ] # Fit the models on the training data
    train_model1 = model1.fit(X_train3, y_train3)
```

```
[ ] # Make predictions
    pred1 = train_model1.predict(X_test3)
```

```
[ ] # Calculate and print evaluation metrics
    print("Model 1 XGBoost - Mean Absolute Error: %.5f" % mean_absolute_error(y_test3, pred1))
    print("Model 1 XGBoost - R^2 Score: %.5f" % r2_score(y_test3, pred1))
```

```
Model 1 XGBoost - Mean Absolute Error: 4.66719
Model 1 XGBoost - R^2 Score: 0.90009
```

60-40

```
# Fit the models on the training data
train_model1 = model1.fit(X_train4, y_train4)
```

```
[ ] # Make predictions
    pred1 = train_model1.predict(X_test4)
```

```
[ ] # Calculate and print evaluation metrics
    print("Model 1 XGBoost - Mean Absolute Error: %.5f" % mean_absolute_error(y_test4, pred1))
    print("Model 1 XGBoost - R^2 Score: %.5f" % r2_score(y_test4, pred1))
```

```
Model 1 XGBoost - Mean Absolute Error: 4.91023
Model 1 XGBoost - R^2 Score: 0.84903
```

ADA Boost

```
from sklearn.ensemble import AdaBoostRegressor
```

```
# Initialize base estimator for regression
base_estimator = DecisionTreeRegressor(max_depth=3)

# Initialize AdaBoost model for regression
adaboost_regressor = AdaBoostRegressor(estimator=base_estimator, n_estimators=50, learning_rate=1.0, random_state=42)
```

80-20

```
from sklearn.metrics import r2_score
r2_score(y_test1,y_pred1)
```

```
-0.04112710818970866
```

```
# Train the model on the training data
adaboost_regressor.fit(X_train1, y_train1)

# Predict on the test set
y_pred1 = adaboost_regressor.predict(X_test1)

# Calculate mean squared error
mse = mean_absolute_error(y_test1, y_pred1)
print("Mean Absolute Error:", mse)
```

```
Mean Absolute Error: 5.08887940318661
```

75-25

```
from sklearn.metrics import r2_score
r2_score(y_test2,y_pred2)
```

```
-0.11160589279975563
```

```
# Train the model on the training data
adaboost_regressor.fit(X_train2, y_train2)

# Predict on the test set
y_pred2 = adaboost_regressor.predict(X_test2)

# Calculate mean squared error
mse = mean_absolute_error(y_test2, y_pred2)
print("Mean Absolute Error:", mse)
```

```
Mean Absolute Error: 4.833365204714036
```

70-30

```
from sklearn.metrics import r2_score
r2_score(y_test3,y_pred3)
```

```
-0.1490283527929619
```

```
# Train the model on the training data
adaboost_regressor.fit(X_train3, y_train3)

# Predict on the test set
y_pred3 = adaboost_regressor.predict(X_test3)

# Calculate mean squared error
mse = mean_absolute_error(y_test3, y_pred3)
print("Mean Absolute Error:", mse)
```

```
Mean Absolute Error: 5.519318141674682
```

60-40

```
from sklearn.metrics import r2_score
r2_score(y_test4,y_pred4)
```

```
-0.13160793971459128
```

```
# Train the model on the training data
adaboost_regressor.fit(X_train4, y_train4)

# Predict on the test set
y_pred4 = adaboost_regressor.predict(X_test4)

# Calculate mean squared error
mse = mean_absolute_error(y_test4, y_pred4)
print("Mean Absolute Error:", mse)
```

```
Mean Absolute Error: 5.16314541546549
```

Bagging(Random Forest)

```
[ ]  # Random Forest Regressor
     rf = RandomForestRegressor()
```

80-20

```
# Fit the model
rf.fit(X_train1, y_train1)
```

```
▾ RandomForestRegressor
RandomForestRegressor()
```

```
[ ]  # Predict on the test set
     y_pred_rf1 = rf.predict(X_test1)
```

```
[ ]  # Evaluate the model
     print("Model 1 Random Forest Regressor - Mean Absolute Error: %.5f" % mean_absolute_error(y_test1, y_pred_rf1))
     print("Random Forest Regressor - R^2 Score:", r2_score(y_test1, y_pred_rf1))
```

```
Model 1 Random Forest Regressor - Mean Absolute Error: 4.34654
Random Forest Regressor - R^2 Score: 0.8724363115203904
```