From Patterns to Predictions:

A Model for Anticipating Elevated Cancer Rates Across Nations

John Pauline Pineda

November 16, 2023

OUTLINE

EXECUTIVE SUMMARY

- ## Study Objective
  - Conduct data analysis and modeling to investigate the potential drivers for high cancer rates between countries given various world performance indicators (social protection and labor, education, economy and growth, environment, climate change, agricultural and rural development, social development, health, science and technology, urban development) and indices (human development, environmental performance).
- ## Methodology and Tools
  - Data Quality Assessment using Python Pandas and NumPy APIs
  - Data Preprocessing using Python Pandas and Scikit-Learn APIs
  - Data Exploration using Python Matplotlib, Seaborn and SciPy APIs
  - Model Development and Validation using Python Matplotlib, Seaborn and SciPy APIs
- ## Overall Findings
  - Data quality issues were identified and handled with the appropriate data preprocessing methods.
  - Relevant numeric indicators which highly correlated with cancer rates were determined.
  - Relevant categorical indices that effectively differentiated varying levels of cancer rates were determined.
  - Cancer rates were modeled using the key numeric and categorical drivers.

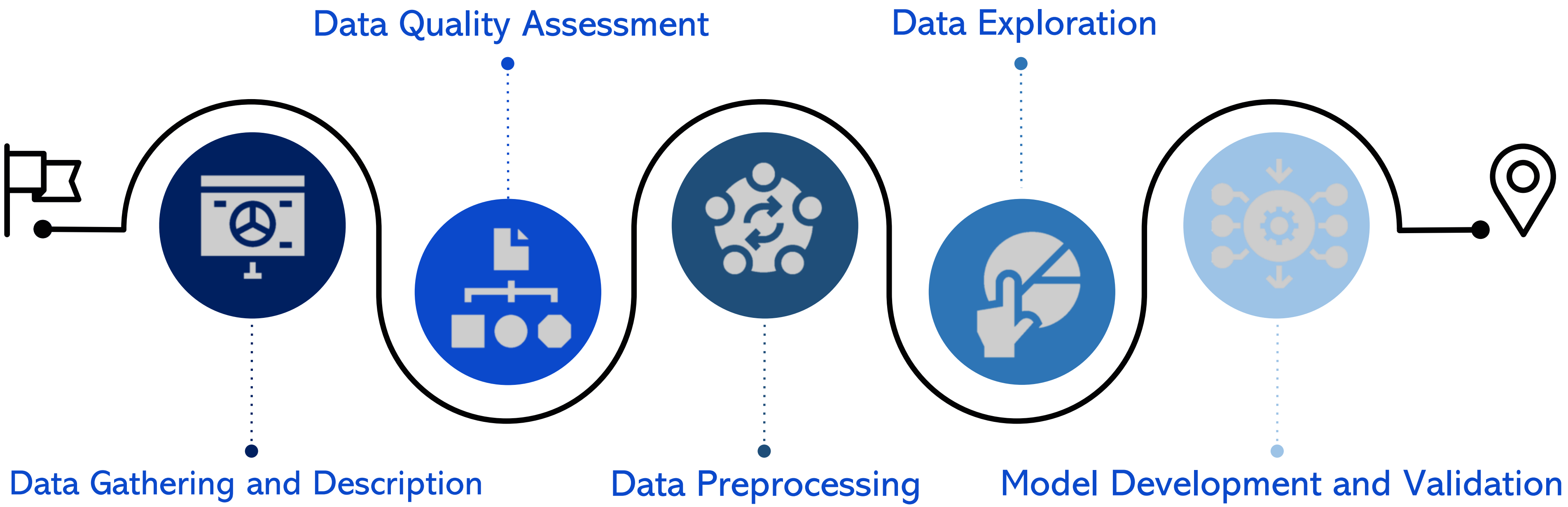
INTRODUCTION

- Cancer – a complex group of diseases characterized by the uncontrolled growth and spread of abnormal cells, has emerged as a leading cause of morbidity and mortality worldwide.
- While the prevalence of cancer varies significantly between countries, anticipating elevated cancer rates given the factors that contribute to these variations is crucial for effective prevention, early detection, and treatment strategies.
- This capstone project generally aims to develop a regression model that could provide robust and reliable estimates of cancer rates from an optimal set of observations and predictors, while delivering accurate predictions when applied to new unseen data.
    - In particular, multiple regression models will be formulated with optimized hyperparameters through internal resampling validation. The final regression model will be selected among candidates based on robust performance estimates. The final model performance and generalization ability will be evaluated through external validation in an independent set.

Section 1

Methodology

METHODOLOGY



- Source assessment
- Data download
- Variable description
- Data dimension
- Column types
- Numeric statistics
- Categorical statistics

- Row duplicates
- Column fill rate
- Row fill rate
- Low variance
- High data skew

- Data cleaning
- Data imputation
- Outlier treatment
- Collinearity
- Transformation
- Centering and scaling
- Data Encoding

- Visual exploration
- Hypothesis testing

- Model development
- Hyperparameter tuning
- Model validation
- Model selection

Section 2

Results Summary

RESULTS – DATA GATHERING

World Performance Indicators

Social Protection and Labor [Source: World Bank]
Education [Source: World Bank]
Economy and Growth [Source: World Bank]
Environment [Source: World Bank]
Climate Change [Source: World Bank]
Agricultural and Rural Development [Source: World Bank]
Social Development [Source: World Bank]
Health [Source: World Bank]
Science and Technology [Source: World Bank]
Urban Development [Source: World Bank]

World Performance Indices

Human Development [Source: Human Development Reports]
Environmental Performance [Source: Yale Center]

- The study hypothesizes that **world performance indicators** (social protection and labor, education, economy and growth, environment, climate change, agricultural and rural development, social development, health, science and technology, urban development) and **indices** (human development, environmental performance) directly influence **cancer rates** across countries.

World Cancer Rates

[Source: World Population Review]

- The objective of the study is to explore and prepare the dataset to determine which among the **world performance indicators** and **indices** are the significant key drivers of **cancer rates** across countries.

RESULTS – DATA DESCRIPTION

Description	Variable	Description	Variable
Country 177 Unique Values	COUNTRY		
GDP Per Person Employed (USD) Min: 1.7K | Mean: 45.3K | Max: 234.6K	GDPPER	Human Development Index Category 4 Categories	HDICAT
Urban Population (% of Total) Min: 13.3 | Mean: 59.8 | Max: 100.0	URBPOP	Environmental Performance Index (Score) Min: 18.9 | Mean: 42.9 | Max: 77.9	EPISCO
Patent Applications by Residents (Count) Min: 1.0 | Mean: 20.6K | Max: 1344.8K	PATRES	GDP per Capita (USD) Min: 0.2K | Mean: 13.9K | Max: 117.3K	GDPCAP
R&D Expenditure (% of GDP) Min: 0.1 | Mean: 1.2 | Max: 5.3	RNDGDP	Tertiary School Enrollment (% Gross) Min: 2.4 | Mean: 50.0 | Max: 143.3	ENRTER
Annual Population Growth (%) Min: -2.1 | Mean: +1.1 | Max: +3.7	POPGRO	Population Density (Persons per Km^2) Min: 2.1 | Mean: 200.8 | Max: 7918.9	POPDEN
Life Expectancy at Birth (Years) Min: 52.7 | Mean: 71.7 | Max: 84.5	LIFEXP	PM2.5 Air Pollution Exposure (% of Total) Min: 0.3 | Mean: 91.9 | Max: 100.0	PM2EXP
Tuberculosis Incidence per 100K (Count) Min: 0.7 | Mean: 105.0 | Max: 592.0	TUBINC	CO2 Emissions (Metric Tons per Capita) Min: 0.0 | Mean: 3.7 | Max: 31.7	CO2EMI
Death by Communicable Disease (% of Total) Min: 1.3 | Mean: 21.3 | Max: 65.2	DTHCMD	Forest Area (% of Land Area) Min: 0.0 | Mean: 32.2 | Max: 97.4	FORARE
Agricultural Land (% of Land Area) Min: 0.5 | Mean: 38.8 | Max: 80.8	AGRLND	Methane Emissions (Kiloton) Min: 0.0K | Mean: 47.8K | Max: 1186.2K	METEMI
Greenhouse Gas Emissions (Kiloton) Min: 0.1K | Mean: 259.5K | Max: 12,942.8K	GHGEMI	Renewable Electricity Output (% of Total) Min: 0.0 | Mean: 39.7 | Max: 100.0	RELOUT

- Original data consisted of:
  - 177 observation rows
  - 1 numeric response column
  - 19 numeric predictor columns
  - 1 categorical predictor column
  - 1 object metadata column
- Numeric data are of different scales.

Target

CANRAT — Cancer Rate Per 100K (%)
Min: 78.4 | Mean: 183.8 | Max: 452.0

21 Predictors + Metadata

177 Observations

- Numeric Target
- Numeric Predictor
- Categorical Predictor
- Object Metadata

RESULTS – DATA QUALITY ASSESSMENT

Description	Variable	Variable	Description
Country	COUNTRY		
GDP Per Person Employed (USD) MIS | MUL | OUT	GDPPER	HDICAT	Human Development Index Category MIS
Urban Population (% of Total) MIS	URBPOP	EPISCO	Environmental Performance Index (Score) MIS | OUT
Patent Applications by Residents (Count) MIS | SKW	PATRES	GDPCAP	GDP per Capita (USD) MIS | MUL | OUT
R&D Expenditure (% of GDP) MIS	RNDGDP	ENRTER	Tertiary School Enrollment (% Gross) MIS
Annual Population Growth (%) MIS	POPGRO	POPDEN	Population Density (Persons per Km^2) MIS | SKW | OUT
Life Expectancy at Birth (Years) MIS	LIFEXP	PM2EXP	PM2.5 Air Pollution Exposure (% of Total) MIS | NZV | SKW | OUT
Tuberculosis Incidence per 100K (Count) MIS | OUT	TUBINC	CO2EMI	CO2 Emissions (Metric Tons per Capita) MIS | OUT
Death by Communicable Disease (% of Total) MIS	DTHCMD	FORARE	Forest Area (% of Land Area) MIS
Agricultural Land (% of Land Area) MIS	AGRLND	METEMI	Methane Emissions (Kiloton) MIS | SKW | MUL | OUT
Greenhouse Gas Emissions (Kiloton) MIS | SKW | MUL | OUT	GHGEMI	RELOUT	Renewable Electricity Output (% of Total) MIS

- Data quality issues identified included:
  - Missing data
  - Skewed distributions
  - Near zero variance
  - High multicollinearity
  - High outlier ratio
- Data preprocessing is needed to address data quality issues.

Target

CANRAT — Cancer Rate Per 100K (%)
OUT

21 Predictors + Metadata

177 Observations

Numeric Target
Numeric Predictor
Categorical Predictor
Object Metadata

MIS High Missing Data
SKW Skewed Distribution
NZV Near-Zero Variance
MUL High Multicollinearity
OUT High Outlier Ratio

RESULTS – DATA PREPROCESSING

Predictors + Metadata	DATA PREPROCESSING STEPS	Observations
21		177
17	High Column-Wise Missing Data[A]: Column Fill Rate<90% | High Row-Wise Missing Data[B]: Row Missing Rate>20%	163
17	Data Imputation: Iterative Imputation for Low Column-Wise Missing Data	163
15	Multicollinearity Detection: Pairwise Correlation Coefficient>0.90[C]	163
15	Skewness Detection | Outlier Detection: Skewness>|3|, Value>75th Percentile+1.5 IQR, Value<25th Percentile-1.5 IQR	163
15	Shape Transformation: Yeo-Johnson Transformation for Skewness and Outlier Reduction	163
14	Post-Transformation Outlier | Skewness Detection: Outlier Count>25[D], Skewness>|3|[D]	163
17	Centering | Scaling: Standardization for Comparable Range | Data Encoding: One-Hot Encoding for Categorical Column	163

Removed 7 Predictors

[A]RNDGDP [A]PATRES [A]ENRTER [A]RELOUT

[C]GDPPER [C]METEMI [D]PM2EXP

Removed 14 Observations

[B]COUNTRY: Guadeloupe, Martinique, French Guiana, New Caledonia, French Polynesia, Guam, Puerto Rico, North Korea, Somalia, South Sudan, Venezuela, Libya, Eritrea, Yemen

RESULTS – DATA EXPLORATION

Variable	Direction	Description	Category
GDPCAP	↑	GDP per Capita (USD)	Economy and Growth
LIFEXP	↑	Life Expectancy at Birth (Years)	Social Development
EPISCO	↑	Environmental Performance Index (Score)	Environmental Policies
URBPOP	↑	Urban Population (% of Total)	Urban Development
POPGRO	↓	Annual Population Growth (%)	Urban Development
DTHCMD	↓	Death by Communicable Disease (% of Total)	Health
TUBINC	↑	Tuberculosis Incidence per 100K (Count)	Health
CO2EMI	↑	CO2 Emissions (Metric Tons per Capita)	Climate Change
HDICAT-VH	↑	Human Development Index – Very High	Social Development
HDICAT-M	↓	Human Development Index – Medium	Social Development
HDICAT-L	↓	Human Development Index – Low	Social Development

→ CANRAT ↑ Cancer Rate Per 100K (%) — Cancer Rates

Statistically significant associations with predictors were determined.

- Higher cancer rates were generally observed among progressive countries as characterized by the following:
  - Higher economic growth leading to advanced industrialization with increased exposure to pollutants but potentially better environmental policies
  - Better socioeconomic policies to support cancer screening and diagnosis
  - More robust healthcare infrastructure to support the completeness of cancer registries and reporting mechanisms
  - Shifting demographics including a potentially aging population with increased disease incidence trends
- Given the findings, a high level of progressiveness may not inherently imply higher cancer rates; rather, the relationship might have been influenced by a combination of demographic, environmental, and healthcare-related factors associated with developed countries.
- The relationship between a country's level of progressiveness and cancer rates may not be straightforward, and other potential drivers must be considered in the future when exploring this issue including data on lifestyle factors (smoking rates, obesity levels) or genetic diversity (infectious agents).

RESULTS – MODEL DEVELOPMENT

GDPCAP	LIFEXP	EPISCO	URBPOP
POPGRO	DTHCMD	HDICAT-VH	CANRAT

Predictors + Metadata **7** → **7**

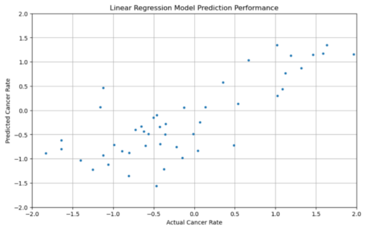
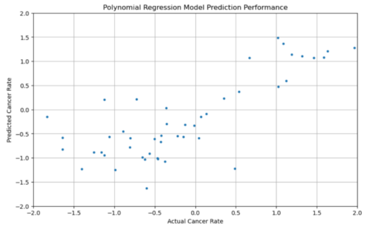
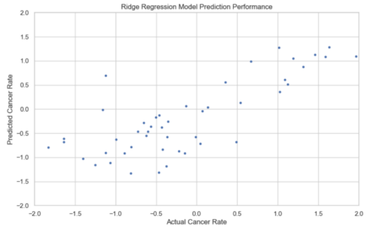
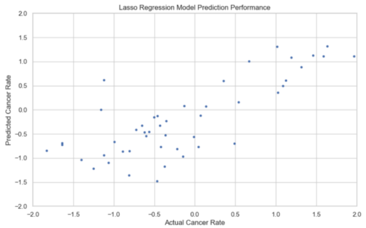
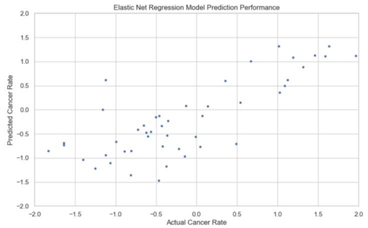
Observations **163** → A **114** → [C]; B **49**

A 70% for Model Training

B 30% for Model External Validation (EV)

C Leave-One-Out Cross-Validation for Model Hyperparameter Tuning
114 Cycles: 113 Folds=Train + 1 Fold=Test

Model Performance: [Y-Axis] Model-Predicted[B] Response versus [X-Axis] Observed Response

Regression Model Algorithm	Linear Regression[D]	Polynomial Regression	Ridge Regression	Lasso Regression	Elastic Net Regression
					
Model Hyperparameters (Number of Hyperparameter Values for Optimization)	degree (1)	degree (1)	alpha (8)	alpha (8)	alpha (8) l1_ratio (9)
Number of Combinations	1 Combination	1 Combination	8 Combinations	8 Combinations	72 Combinations
Optimal[C] Hyperparameter Combination Values	degree=1	degree=2	alpha=10	alpha=0.01	alpha=0.01 l1_ratio=0.90
Apparent[A] | EV[B] R-Squared	0.6332 | 0.6446	0.7908 | 0.6324	0.6220 | 0.6352	0.6295 | 0.6405	0.6298 | 0.6409
Apparent[A] | EV[B] MSE	0.3550 | 0.3716	0.2024 | 0.3844	0.3659 | 0.3815	0.3586 | 0.3760	0.3582 | 0.3755
Apparent[A] | EV[B] MAE	0.4609 | 0.4773	0.3503 | 0.4867	0.4680 | 0.4838	0.4608 | 0.4805	0.4608 | 0.4800

D FINAL MODEL: Demonstrated the best EV R-Squared, MSE, and MAE – with no excessive overfitting comparing the apparent and EV performance.

Section 3

Detailed Findings

RESULTS – DATA QUALITY ASSESSMENT

- ## Findings

1. No duplicated rows observed.
2. Missing data noted for 20 variables with Null.Count>0 and Fill.Rate<1.0.
    - RNDGDP: Null.Count = 103, Fill.Rate = 0.418
    - PATRES: Null.Count = 69, Fill.Rate = 0.610
    - ENRTER: Null.Count = 61, Fill.Rate = 0.655
    - RELOUT: Null.Count = 24, Fill.Rate = 0.864
    - GDPPER: Null.Count = 12, Fill.Rate = 0.932
    - EPISCO: Null.Count = 12, Fill.Rate = 0.932
    - HDICAT: Null.Count = 10, Fill.Rate = 0.943
    - PM2EXP: Null.Count = 10, Fill.Rate = 0.943
    - DTHCMD: Null.Count = 7, Fill.Rate = 0.960
    - METEMI: Null.Count = 7, Fill.Rate = 0.960
    - CO2EMI: Null.Count = 7, Fill.Rate = 0.960
    - GDPCAP: Null.Count = 7, Fill.Rate = 0.960
    - GHGEMI: Null.Count = 7, Fill.Rate = 0.960
    - FORARE: Null.Count = 4, Fill.Rate = 0.977
    - TUBINC: Null.Count = 3, Fill.Rate = 0.983
    - AGRLND: Null.Count = 3, Fill.Rate = 0.983
    - POPGRO: Null.Count = 3, Fill.Rate = 0.983
    - POPDEN: Null.Count = 3, Fill.Rate = 0.983
    - URBPOP: Null.Count = 3, Fill.Rate = 0.983
    - LIFEXP: Null.Count = 3, Fill.Rate = 0.983
3. 120 observations noted with at least 1 missing data. From this number, 14 observations reported high Missing.Rate>0.2.
    - COUNTRY=Guadeloupe: Missing.Rate= 0.909
    - COUNTRY=Martinique: Missing.Rate= 0.909
    - COUNTRY=French Guiana: Missing.Rate= 0.909
    - COUNTRY=New Caledonia: Missing.Rate= 0.500
    - COUNTRY=French Polynesia: Missing.Rate= 0.500
    - COUNTRY=Guam: Missing.Rate= 0.500
    - COUNTRY=Puerto Rico: Missing.Rate= 0.409
    - COUNTRY=North Korea: Missing.Rate= 0.227
    - COUNTRY=Somalia: Missing.Rate= 0.227
    - COUNTRY=South Sudan: Missing.Rate= 0.227
    - COUNTRY=Venezuela: Missing.Rate= 0.227
    - COUNTRY=Libya: Missing.Rate= 0.227
    - COUNTRY=Eritrea: Missing.Rate= 0.227
    - COUNTRY=Yemen: Missing.Rate= 0.227
4. Low variance observed for 1 variable with First.Second.Mode.Ratio>5.
    - PM2EXP: First.Second.Mode.Ratio = 53.000
5. No low variance observed for any variable with Unique.Count.Ratio>10.
6. High skewness observed for 5 variables with Skewness>3 or Skewness<(-3).
    - POPDEN: Skewness = +10.267
    - GHGEMI: Skewness = +9.496
    - PATRES: Skewness = +9.284
    - METEMI: Skewness = +5.801
    - PM2EXP: Skewness = -3.141

- Code chunk formulated to generate the assessment tables is presented as a markdown file here.
- Python notebook for the entire capstone project is saved on GitHub and can be accessed here.

RESULTS – DATA CLEANING

- Findings

1. Subsets of rows and columns with high rates of missing data were removed from the dataset:
   - 4 variables with Fill.Rate<0.9 were excluded for subsequent analysis.
     - RNDGDP: Null.Count = 103, Fill.Rate = 0.418
     - PATRES: Null.Count = 69, Fill.Rate = 0.610
     - ENRTER: Null.Count = 61, Fill.Rate = 0.655
     - RELOUT: Null.Count = 24, Fill.Rate = 0.864
   - 14 rows with Missing.Rate>0.2 were exluded for subsequent analysis.
     - COUNTRY=Guadeloupe: Missing.Rate= 0.909
     - COUNTRY=Martinique: Missing.Rate= 0.909
     - COUNTRY=French Guiana: Missing.Rate= 0.909
     - COUNTRY=New Caledonia: Missing.Rate= 0.500
     - COUNTRY=French Polynesia: Missing.Rate= 0.500
     - COUNTRY=Guam: Missing.Rate= 0.500
     - COUNTRY=Puerto Rico: Missing.Rate= 0.409
     - COUNTRY=North Korea: Missing.Rate= 0.227
     - COUNTRY=Somalia: Missing.Rate= 0.227
     - COUNTRY=South Sudan: Missing.Rate= 0.227
     - COUNTRY=Venezuela: Missing.Rate= 0.227
     - COUNTRY=Libya: Missing.Rate= 0.227
     - COUNTRY=Eritrea: Missing.Rate= 0.227
     - COUNTRY=Yemen: Missing.Rate= 0.227
2. No variables were removed due to zero or near-zero variance.
3. The cleaned dataset is comprised of:
   - **163 rows** (observations)
   - **18 columns** (variables)
     - **1/18 metadata** (object)
       - COUNTRY
     - **1/18 target** (numeric)
       - CANRAT
     - **15/18 predictor** (numeric)
       - GDPPER
       - URBPOP
       - POPGRO
       - LIFEXP
       - TUBINC
       - DTHCMD
       - AGRLND
       - GHGEMI
       - METEMI
       - FORARE
       - CO2EMI
       - PM2EXP
       - POPDEN
       - GDPCAP
       - EPISCO
     - **1/18 predictor** (categorical)
       - HDICAT

- Code chunk formulated to generate the assessment tables is presented as a markdown file here.
- Python notebook for the entire capstone project is saved on GitHub and can be accessed here.

RESULTS – MISSING DATA IMPUTATION

- ## Findings

1. Missing data for numeric variables were imputed using the iterative imputer algorithm with a linear regression estimator.
    - GDPPER: Null.Count = 1
    - FORARE: Null.Count = 1
    - PM2EXP: Null.Count = 5
2. Missing data for categorical variables were imputed using the most frequent value.
    - HDICAP: Null.Count = 1

- Code chunk formulated to generate the assessment tables is presented as a markdown file here.
- Python notebook for the entire capstone project is saved on GitHub and can be accessed here.

RESULTS – OUTLIER TREATMENT

- ## Findings

1. High number of outliers observed for 5 numeric variables with Outlier.Ratio>0.10 and marginal to high Skewness.
    - PM2EXP: Outlier.Count = 37, Outlier.Ratio = 0.226, Skewness=-3.061
    - GHGEMI: Outlier.Count = 27, Outlier.Ratio = 0.165, Skewness=+9.299
    - GDPCAP: Outlier.Count = 22, Outlier.Ratio = 0.134, Skewness=+2.311
    - POPDEN: Outlier.Count = 20, Outlier.Ratio = 0.122, Skewness=+9.972
    - METEMI: Outlier.Count = 20, Outlier.Ratio = 0.122, Skewness=+5.688
2. Minimal number of outliers observed for 5 numeric variables with Outlier.Ratio<0.10 and normal Skewness.
    - TUBINC: Outlier.Count = 12, Outlier.Ratio = 0.073, Skewness=+1.747
    - CO2EMI: Outlier.Count = 11, Outlier.Ratio = 0.067, Skewness=+2.693
    - GDPPER: Outlier.Count = 3, Outlier.Ratio = 0.018, Skewness=+1.554
    - EPISCO: Outlier.Count = 3, Outlier.Ratio = 0.018, Skewness=+0.635
    - CANRAT: Outlier.Count = 2, Outlier.Ratio = 0.012, Skewness=+0.910

- Code chunk formulated to generate the assessment tables is presented as a markdown file here.
- Python notebook for the entire capstone project is saved on GitHub and can be accessed here.

RESULTS – COLLINEARITY

- ## Findings

1. Majority of the numeric variables reported moderate to high correlation which were statistically significant.
2. Among pairwise combinations of numeric variables, high Pearson.Correlation.Coefficient values were noted for:
   - GDPPER and GDPCAP: Pearson.Correlation.Coefficient = +0.921
   - GHGEMI and METEMI: Pearson.Correlation.Coefficient = +0.905
3. Among the highly correlated pairs, variables with the lowest correlation against the target variable were removed.
   - GDPPER: Pearson.Correlation.Coefficient = +0.690
   - METEMI: Pearson.Correlation.Coefficient = +0.062
4. The cleaned dataset is comprised of:
   - **163 rows** (observations)
   - **16 columns** (variables)
     - **1/16 metadata** (object)
       - COUNTRY
     - **1/16 target** (numeric)
       - CANRAT
     - **13/16 predictor** (numeric)
       - URBPOP
       - POPGRO
       - LIFEXP
       - TUBINC
       - DTHCMD
       - AGRLND
       - GHGEMI
       - FORARE
       - CO2EMI
       - PM2EXP
       - POPDEN
       - GDPCAP
       - EPISCO
     - **1/16 predictor** (categorical)
       - HDICAT

- Code chunk formulated to generate the assessment tables is presented as a markdown file here.
- Python notebook for the entire capstone project is saved on GitHub and can be accessed here.

RESULTS – SHAPE TRANSFORMATION

- ## Findings

1. A Yeo-Johnson transformation was applied to all numeric variables to improve distributional shape.
2. Most variables achieved symmetrical distributions with minimal outliers after transformation.
3. One variable which remained skewed even after applying shape transformation was removed.
   - PM2EXP
4. The transformed dataset is comprised of:
   - **163 rows** (observations)
   - **15 columns** (variables)
     - **1/15 metadata** (object)
       - COUNTRY
     - **1/15 target** (numeric)
       - CANRAT
     - **12/15 predictor** (numeric)
       - URBPOP
       - POPGRO
       - LIFEXP
       - TUBINC
       - DTHCMD
       - AGRLND
       - GHGEMI
       - FORARE
       - CO2EMI
       - POPDEN
       - GDPCAP
       - EPISCO
     - **1/15 predictor** (categorical)
       - HDICAT

- Code chunk formulated to generate the assessment tables is presented as a markdown file here.
- Python notebook for the entire capstone project is saved on GitHub and can be accessed here.

RESULTS – CENTERING AND SCALING

- ## Findings

1. All numeric variables were transformed using the standardization method to achieve a comparable scale between values.
2. The scaled dataset is comprised of:
   - **163 rows** (observations)
   - **15 columns** (variables)
     - **1/15 metadata** (object)
       - COUNTRY
     - **1/15 target** (numeric)
       - CANRAT
     - **12/15 predictor** (numeric)
       - URBPOP
       - POPGRO
       - LIFEXP
       - TUBINC
       - DTHCMD
       - AGRLND
       - GHGEMI
       - FORARE
       - CO2EMI
       - POPDEN
       - GDPCAP
       - EPISCO
     - **1/15 predictor** (categorical)
       - HDICAT

- Code chunk formulated to generate the assessment tables is presented as a markdown file here.
- Python notebook for the entire capstone project is saved on GitHub and can be accessed here.

RESULTS – DATA ENCODING

- ## Findings

1. One-hot encoding was applied to the HDICAP_VH variable resulting to 4 additional columns in the dataset:
   - HDICAP_L
   - HDICAP_M
   - HDICAP_H
   - HDICAP_VH

- Code chunk formulated to generate the assessment tables is presented as a markdown file here.
- Python notebook for the entire capstone project is saved on GitHub and can be accessed here.

RESULTS – EXPLORATORY DATA ANALYSIS

- ## Findings

1. Bivariate analysis identified individual predictors with generally linear relationship to the target variable based on visual inspection.
2. Increasing values for the following predictors correspond to higher CANRAT measurements:
   - URBPOP
   - LIFEXP
   - CO2EMI
   - GDPCAP
   - EPISCO
   - HDICAP_VH
3. Decreasing values for the following predictors correspond to higher CANRAT measurements:
   - POPGRO
   - TUBINC
   - DTHCMD
   - HDICAP_L
   - HDICAP_M
4. Values for the following predictors did not affect CANRAT measurements:
   - AGRLND
   - GHGEMI
   - FORARE
   - POPDEN
   - HDICAP_H

- Code chunk formulated to generate the assessment tables is presented as a markdown file here.
- Python notebook for the entire capstone project is saved on GitHub and can be accessed here.

RESULTS – HYPOTHESIS TESTING

• Findings

1. The relationship between the numeric predictors to the CANRAT target variable was statistically evaluated using the following hypotheses:
   - **Null**: Pearson correlation coefficient is equal to zero
   - **Alternative**: Pearson correlation coefficient is not equal to zero
2. There is sufficient evidence to conclude of a statistically significant linear relationship between the CANRAT target variable and 10 of the 12 numeric predictors given their high Pearson correlation coefficient values with reported low p-values less than the significance level of 0.05.
   - GDPCAP: Pearson.Correlation.Coefficient=+0.735, Correlation.PValue=0.000
   - LIFEXP: Pearson.Correlation.Coefficient=+0.702, Correlation.PValue=0.000
   - DTHCMD: Pearson.Correlation.Coefficient=-0.687, Correlation.PValue=0.000
   - EPISCO: Pearson.Correlation.Coefficient=+0.648, Correlation.PValue=0.000
   - TUBINC: Pearson.Correlation.Coefficient=+0.628, Correlation.PValue=0.000
   - CO2EMI: Pearson.Correlation.Coefficient=+0.585, Correlation.PValue=0.000
   - POPGRO: Pearson.Correlation.Coefficient=-0.498, Correlation.PValue=0.000
   - URBPOP: Pearson.Correlation.Coefficient=+0.479, Correlation.PValue=0.000
   - GHGEMI: Pearson.Correlation.Coefficient=+0.232, Correlation.PValue=0.002
   - FORARE: Pearson.Correlation.Coefficient=+0.165, Correlation.PValue=0.035
3. The relationship between the categorical predictors to the CANRAT target variable was statistically evaluated using the following hypotheses:
   - **Null**: Difference in the means between groups 0 and 1 is equal to zero
   - **Alternative**: Difference in the means between groups 0 and 1 is not equal to zero
4. There is sufficient evidence to conclude of a statistically significant difference between the means of CANRAT measuremens obtained from groups 0 and 1 in 3 of the 4 categorical predictors given their high t-test statistic values with reported low p-values less than the significance level of 0.05.
   - HDICAT_VH: T.Test.Statistic=-10.605, T.Test.PValue=0.000
   - HDICAT_L: T.Test.Statistic=+6.559, T.Test.PValue=0.000
   - HDICAT_M: T.Test.Statistic=+5.104, T.Test.PValue=0.000

- Code chunk formulated to generate the assessment tables is presented as a markdown file here.
- Python notebook for the entire capstone project is saved on GitHub and can be accessed here.

RESULTS – PRE-MODELLING DATA ANALYSIS

- Findings

1. Among the 10 numeric variables determined to have a statistically significant linear relationship between the CANRAT target variable, only 6 were retained with absolute Pearson correlation coefficient values greater than 0.50.
    - GDPCAP: Pearson.Correlation.Coefficient=+0.735, Correlation.PValue=0.000
    - LIFEXP: Pearson.Correlation.Coefficient=+0.702, Correlation.PValue=0.000
    - DTHCMD: Pearson.Correlation.Coefficient=-0.687, Correlation.PValue=0.000
    - EPISCO: Pearson.Correlation.Coefficient=+0.648, Correlation.PValue=0.000
    - TUBINC: Pearson.Correlation.Coefficient=+0.628, Correlation.PValue=0.000
    - CO2EMI: Pearson.Correlation.Coefficient=+0.585, Correlation.PValue=0.000
2. Among the 4 4 categorical predictors determined to have a a statistically significant difference between the means of CANRAT measuremens obtained from groups 0 and 1, only 1 was retained with absolute T-Test statistics greater than 10.
    - HDICAT_VH: T.Test.Statistic=-10.605, T.Test.PValue=0.000

- Code chunk formulated to generate the assessment tables is presented as a markdown file here.
- Python notebook for the entire capstone project is saved on GitHub and can be accessed here.

RESULTS – LINEAR REGRESSION MODEL

- ## Findings

Linear Regression explores the linear relationship between a scalar response and one or more covariates by having the conditional mean of the dependent variable be an affine function of the independent variables. The relationship is modeled through a disturbance term which represents an unobserved random variable that adds noise. The algorithm is typically formulated from the data using the least squares method which seeks to estimate the coefficients by minimizing the squared residual function. The linear equation assigns one scale factor represented by a coefficient to each covariate and an additional coefficient called the intercept or the bias coefficient which gives the line an additional degree of freedom allowing to move up and down a two-dimensional plot.

1. The linear regression model from the **sklearn.linear_model** Python library API was implemented.
2. The model contains 1 hyperparameter:
   - degree = degree held constant at a value of 1
3. No hyperparameter tuning was conducted.
4. The apparent model performance of the optimal model is summarized as follows:
   - **R-Squared** = 0.6332
   - **Mean Squared Error** = 0.3550
   - **Mean Absolute Error** = 0.4609
5. The independent test model performance of the final model is summarized as follows:
   - **R-Squared** = 0.6446
   - **Mean Squared Error** = 0.3716
   - **Mean Absolute Error** = 0.4773
6. Apparent and independent test model performance are relatively comparable, indicative of the absence of model overfitting.

- Code chunk formulated to generate the assessment tables is presented as a markdown file here.
- Python notebook for the entire capstone project is saved on GitHub and can be accessed here.

RESULTS – POLYNOMIAL REGRESSION MODEL

• Findings

Polynomial Regression explores the relationship between one or more covariates and a scalar response which is modelled as an nth degree polynomial of the covariates. The algorithm fits a nonlinear relationship between the value of the independent variables and the corresponding conditional mean of the dependent variable. Although polynomial regression fits a nonlinear model to the data, as a statistical estimation problem it is linear, in the sense that the regression function is linear in the unknown parameters that are estimated from the data. For this reason, polynomial regression is considered to be a special case of multiple linear regression.

1. The polynomial regression model from the **sklearn.linear_model** Python library API was implemented.
2. The model contains 1 hyperparameter:
   - degree = degree held constant at a value of 2
3. No hyperparameter tuning was conducted.
4. The apparent model performance of the optimal model is summarized as follows:
   - **R-Squared** = 0.7908
   - **Mean Squared Error** = 0.2024
   - **Mean Absolute Error** = 0.3503
5. The independent test model performance of the final model is summarized as follows:
   - **R-Squared** = 0.6324
   - **Mean Squared Error** = 0.3844
   - **Mean Absolute Error** = 0.4867
6. Apparent and independent test model performance are relatively different, indicative of the presence of model overfitting.

- Code chunk formulated to generate the assessment tables is presented as a markdown file here.
- Python notebook for the entire capstone project is saved on GitHub and can be accessed here.

RESULTS – RIDGE REGRESSION MODEL

• Findings

Ridge Regression is a regression technique aimed at preventing overfitting in linear regression models when the model is too complex and fits the training data very closely, but performs poorly on new and unseen data. The algorithm adds a penalty term (referred to as the L2 regularization) to the linear regression cost function that is proportional to the square of the magnitude of the coefficients. This approach helps to reduce the magnitude of the coefficients in the model, which in turn can prevent overfitting and is especially helpful where there are many correlated predictor variables in the model. A hyperparameter alpha serving as a constant that multiplies the L2 term thereby controlling regularization strength needs to be optimized through cross-validation.

1. The ridge regression model from the **sklearn.linear_model** Python library API was implemented.
2. The model contains 1 hyperparameter:
   - alpha = alpha made to vary across a range of values equal to 0.0001 to 1000
3. Hyperparameter tuning was conducted using the leave-one-out-cross-validation method with optimal model performance determined for:
   - alpha = 10
4. The apparent model performance of the optimal model is summarized as follows:
   - **R-Squared** = 0.6220
   - **Mean Squared Error** = 0.3659
   - **Mean Absolute Error** = 0.4680
5. The independent test model performance of the final model is summarized as follows:
   - **R-Squared** = 0.6352
   - **Mean Squared Error** = 0.3815
   - **Mean Absolute Error** = 0.4838
6. Apparent and independent test model performance are relatively comparable, indicative of the absence of model overfitting.

- Code chunk formulated to generate the assessment tables is presented as a markdown file here.
- Python notebook for the entire capstone project is saved on GitHub and can be accessed here.

RESULTS – LASSO REGRESSION MODEL

- ## Findings

Least Absolute Shrinkage and Selection Operator Regression is a regression technique aimed at preventing overfitting in linear regression models when the model is too complex and fits the training data very closely, but performs poorly on new and unseen data. The algorithm adds a penalty term (referred to as the L1 regularization) to the linear regression cost function that is proportional to the absolute value of the coefficients. This approach can be useful for feature selection, as it tends to shrink the coefficients of less important predictor variables to zero, which can help simplify the model and improve its interpretability. A hyperparameter alpha serving as a constant that multiplies the L1 term thereby controlling regularization strength needs to be optimized through cross-validation.

1. The lasso regression model from the **sklearn.linear_model** Python library API was implemented.
2. The model contains 1 hyperparameter:
   - alpha = alpha made to vary across a range of values equal to 0.0001 to 1000
3. Hyperparameter tuning was conducted using the leave-one-out-cross-validation method with optimal model performance determined for:
   - alpha = 0.01
4. The apparent model performance of the optimal model is summarized as follows:
   - **R-Squared** = 0.6295
   - **Mean Squared Error** = 0.3586
   - **Mean Absolute Error** = 0.4608
5. The independent test model performance of the final model is summarized as follows:
   - **R-Squared** = 0.6405
   - **Mean Squared Error** = 0.3760
   - **Mean Absolute Error** = 0.4805
6. Apparent and independent test model performance are relatively comparable, indicative of the absence of model overfitting.

- Code chunk formulated to generate the assessment tables is presented as a markdown file here.
- Python notebook for the entire capstone project is saved on GitHub and can be accessed here.

RESULTS – ELASTIC NET REGRESSION MODEL

- ## Findings

Elastic Net Regression is a regression technique aimed at preventing overfitting in linear regression models when the model is too complex and fits the training data very closely, but performs poorly on new and unseen data. The algorithm is a combination of the ridge and LASSO regression methods, by adding both L1 and L2 regularization terms in the cost function. This approach can be useful when there are many predictor variables that are correlated with the response variable, but only a subset of them are truly important for predicting the response. The L1 regularization term can help to select the important variables, while the L2 regularization term can help to reduce the magnitude of the coefficients. Hyperparameters alpha which serves as the constant that multiplies the penalty terms, and l1_ratio that serves as the mixing parameter that penalizes as a combination of L1 and L2 regularization - need to be optimized through cross-validation.

1. The elastic net regression model from the **sklearn.linear_model** Python library API was implemented.
2. The model contains 2 hyperparameters:
   - l1_ratio = l1_ratio made to vary across a range of values equal to 0.1 to 0.9
   - alpha = alpha made to vary across a range of values equal to 0.0001 to 1000
3. Hyperparameter tuning was conducted using the leave-one-out-cross-validation method with optimal model performance determined for:
   - l1_ratio = 0.9
   - alpha = 0.01
4. The apparent model performance of the optimal model is summarized as follows:
   - **R-Squared** = 0.6298
   - **Mean Squared Error** = 0.3582
   - **Mean Absolute Error** = 0.4608
5. The independent test model performance of the final model is summarized as follows:
   - **R-Squared** = 0.6409
   - **Mean Squared Error** = 0.3755
   - **Mean Absolute Error** = 0.4800
6. Apparent and independent test model performance are relatively comparable, indicative of the absence of model overfitting.

- Code chunk formulated to generate the assessment tables is presented as a markdown file here.
- Python notebook for the entire capstone project is saved on GitHub and can be accessed here.

RESULTS – MODEL VALIDATION AND SELECTION

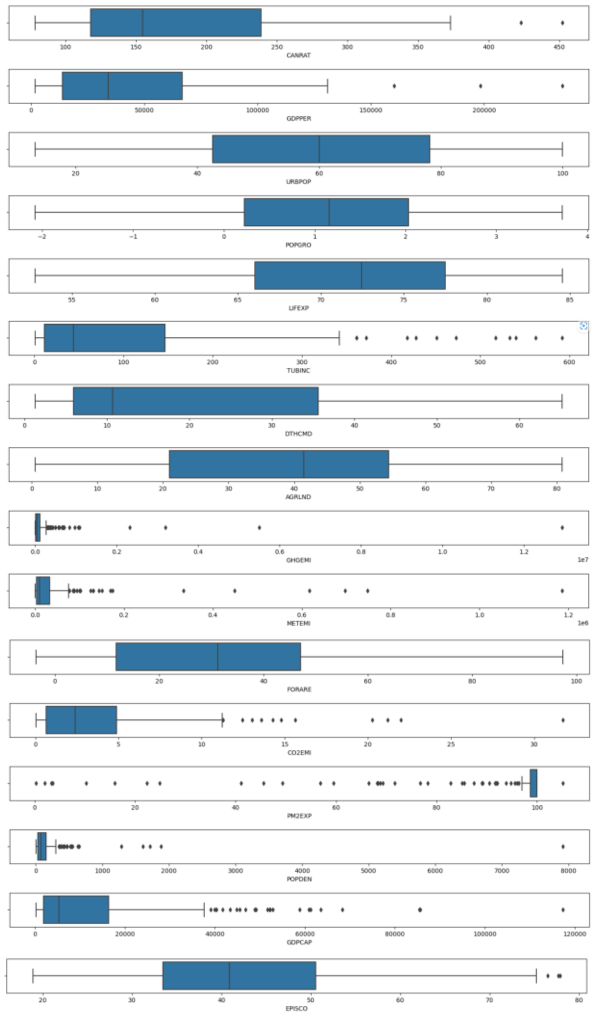
- ## Findings

1. The linear regression model demonstrated the best independent test model performance among the candidate models. This model which did not apply any form of regularization proved to be more superior than the penalized models potentially due to the presence of a smaller subset of equally informative predictors - leading to a minimal over-confidence in the parameter estimates and in effect, minimal impact of the regularization constraints.
   - **R-Squared** = 0.6446
   - **Mean Squared Error** = 0.3716
   - **Mean Absolute Error** = 0.4773
2. Among the penalized models, the optimal elastic net regression model demonstrated the best independent test model performance with the tuned hyperparameter leaning towards a higher L1 regulairzation effect (optimal L1 ratio equals to 0.9 which is reminiscent of lasso where L1 ratio equals to 1.0).
   - **R-Squared** = 0.6409
   - **Mean Squared Error** = 0.3755
   - **Mean Absolute Error** = 0.4800
3. The optimal lasso regression model using an L1 regularization term demonstrated an equally high independent test model performance, confirming the earlier findings.
   - **R-Squared** = 0.6405
   - **Mean Squared Error** = 0.3760
   - **Mean Absolute Error** = 0.4805
4. The optimal ridge regression model using an L2 regularization term equally demonstrated good independent test model performance.
   - **R-Squared** = 0.6352
   - **Mean Squared Error** = 0.3815
   - **Mean Absolute Error** = 0.4838
5. The polynomial regression model demonstrated the worst independent test model performance. In addition, metrics obtained from the apparent and independent test model performance are relatively different, indicative of the presence of model overfitting.
   - **R-Squared** = 0.6324
   - **Mean Squared Error** = 0.3844
   - **Mean Absolute Error** = 0.4867
6. The computed r-squared metrics for the formulated models were relatively low - only ranging from 0.63 to 0.65, which could be further improved by:
   - Considering more informative predictors
   - Considering more complex models other than linear regression and its variants

- Code chunk formulated to generate the assessment tables is presented as a markdown file here.
- Python notebook for the entire capstone project is saved on GitHub and can be accessed here.

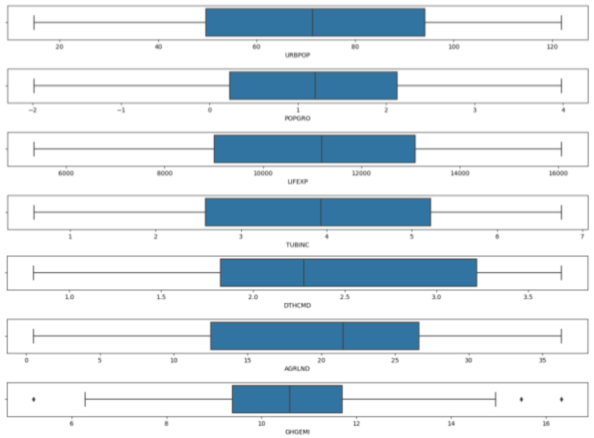
PLOTS – OUTLIER ANALYSIS

- Original Data

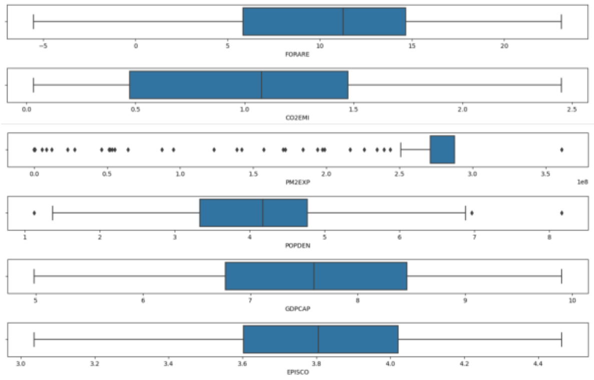


- Transformed Data

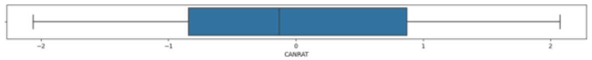
Removed due to high multicollinearity



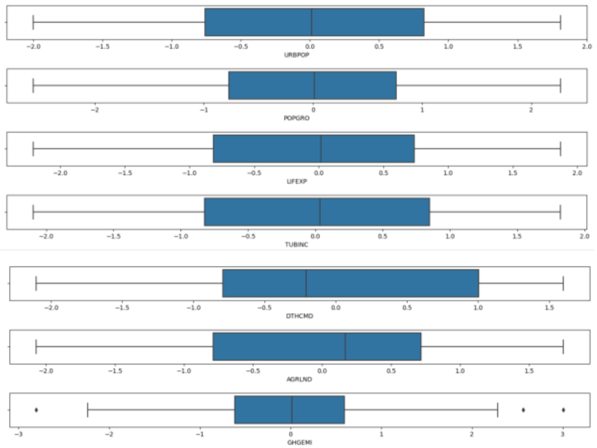
Removed due to high multicollinearity



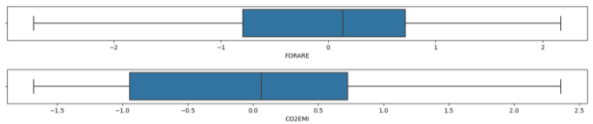
- Centered and Scaled Data



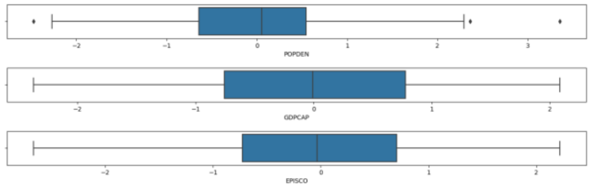
Removed due to high multicollinearity



Removed due to high multicollinearity

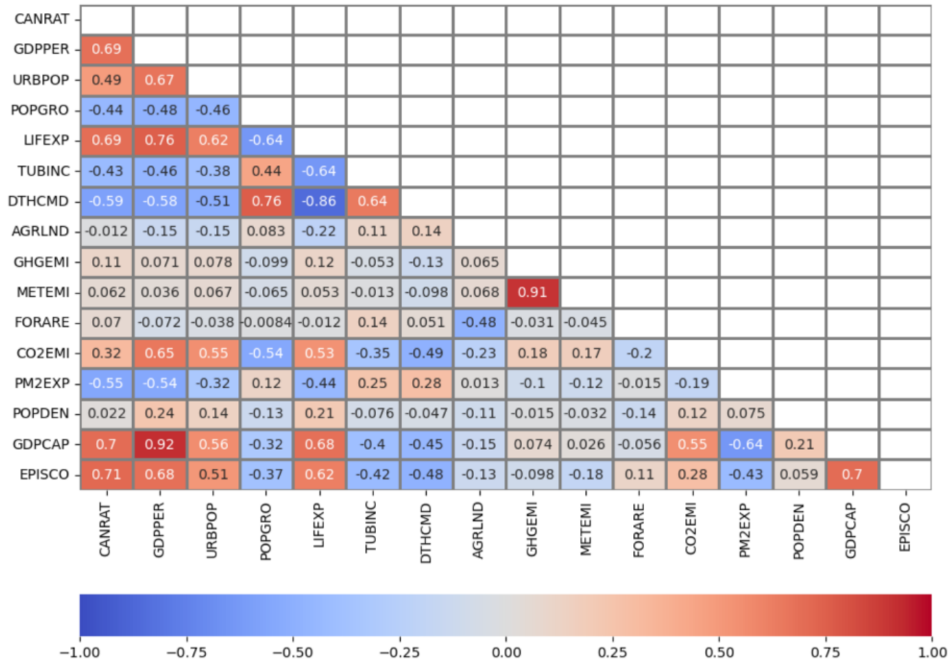


Removed due to persistent skew after transformation

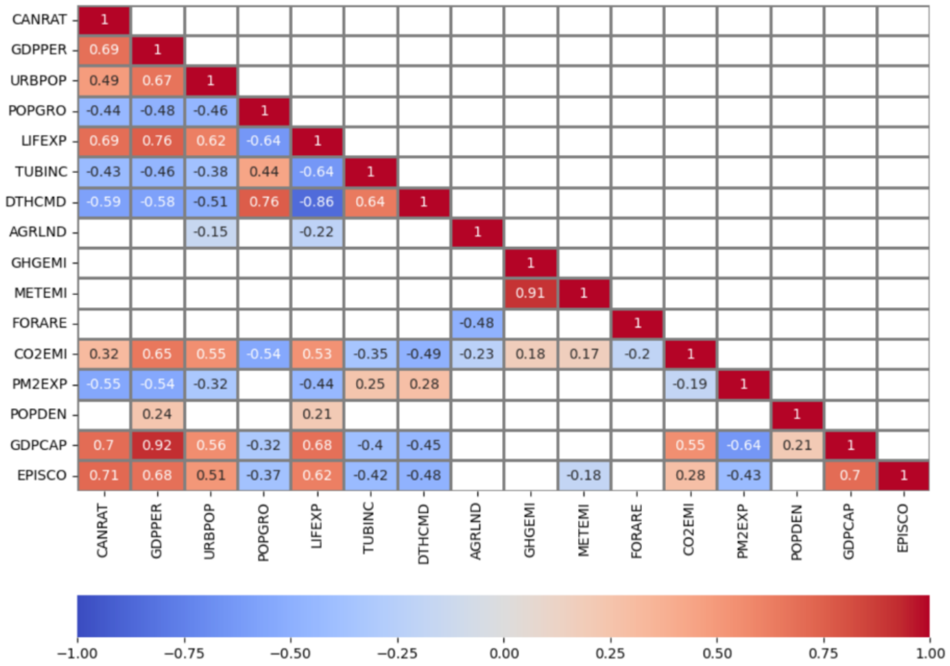


PLOTS – CORRELATION ANALYSIS

- All Correlations

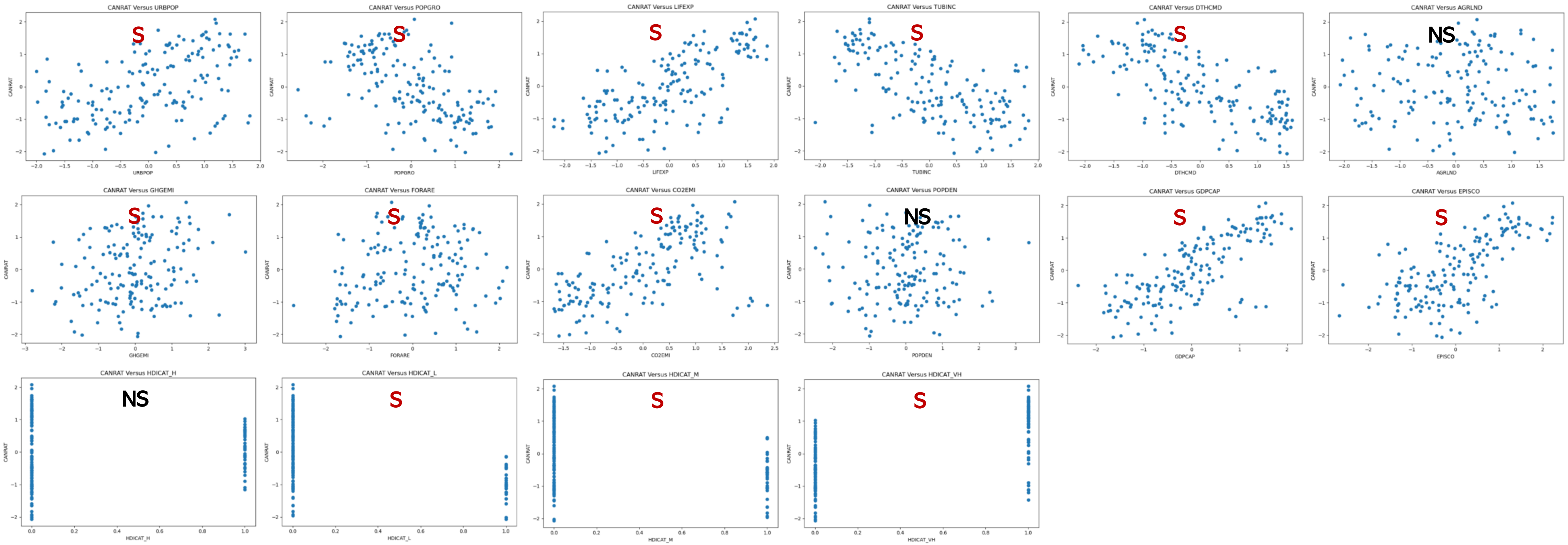


- Statistically Significant Correlations



PLOTS – EXPLORATORY DATA ANALYSIS

- Relationship Scatterplots

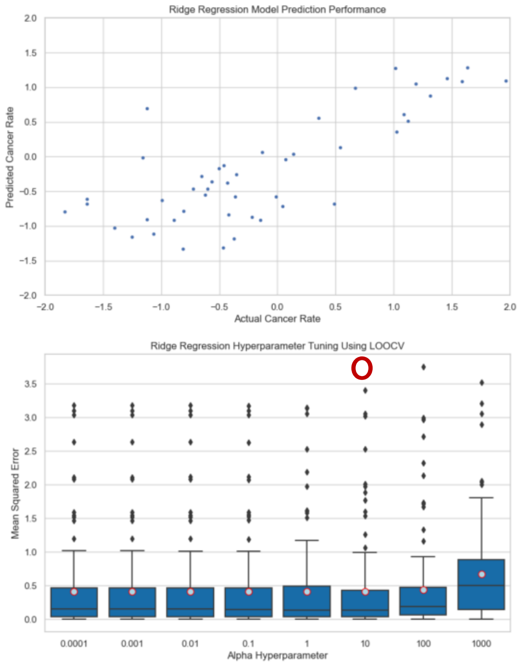


S: Association between predictor and target was statistically significant

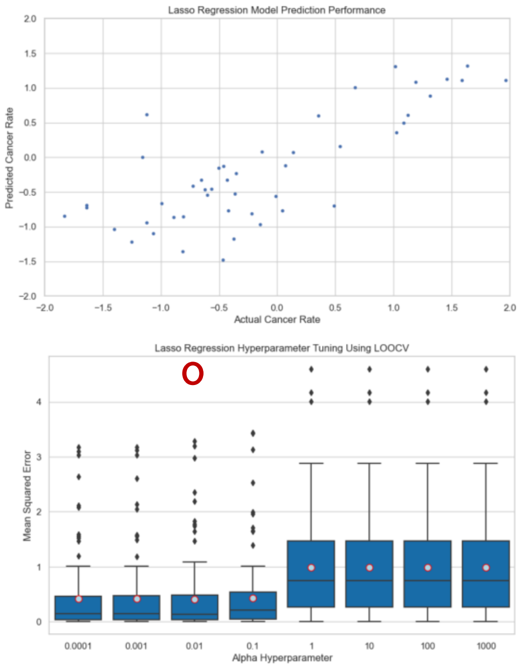
NS: Association between predictor and target was not statistically significant

PLOTS – HYPERPARAMETER TUNING BY LOOCV

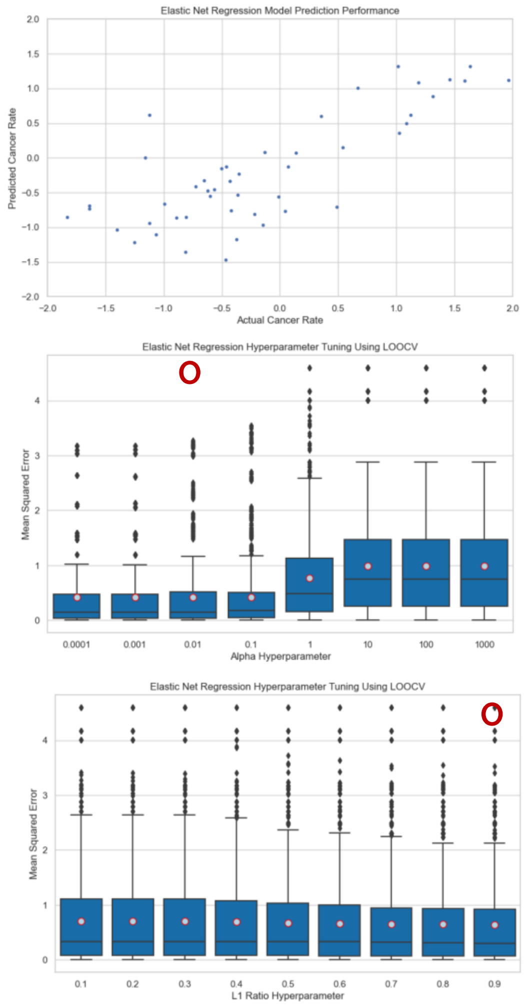
- Penalized – Ridge



- Penalized – Lasso



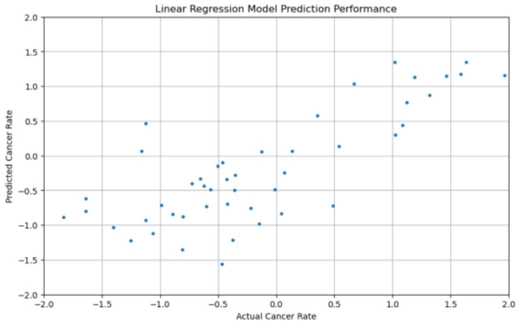
- Penalized – Elastic Net



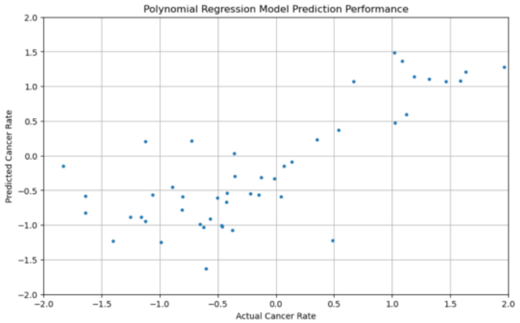
O: Optimal hyperparameter setting determined from LOOCV

PLOTS – MODEL VALIDATION ON TEST DATA

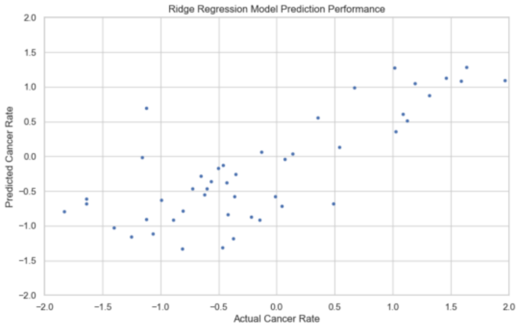
- Linear



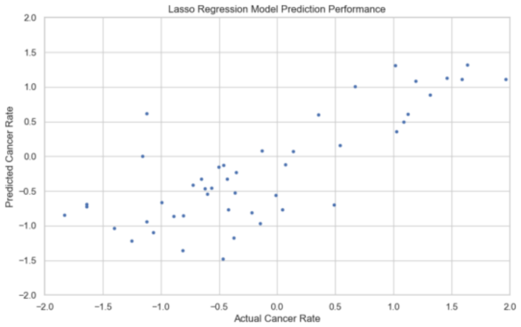
- Polynomial



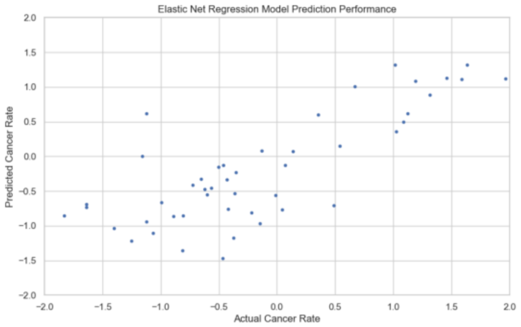
- Penalized – Ridge



- Penalized – Lasso

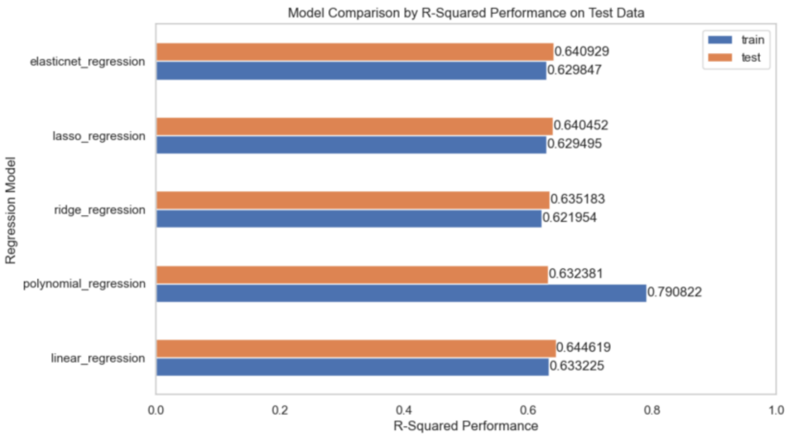


- Penalized – Elastic Net

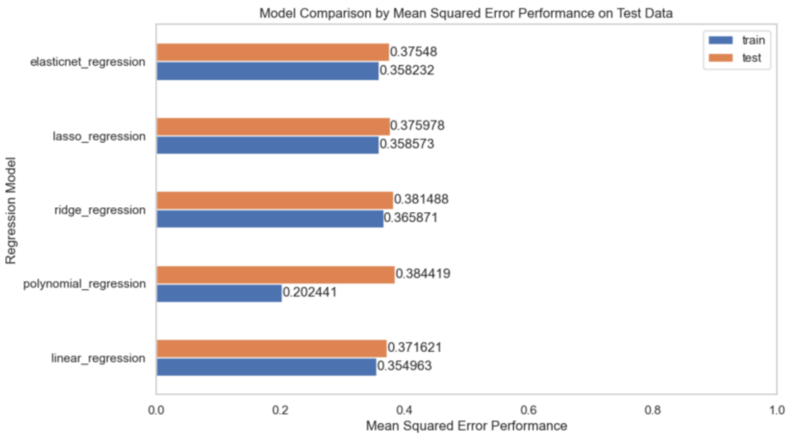


PLOTS – MODEL SELECTION

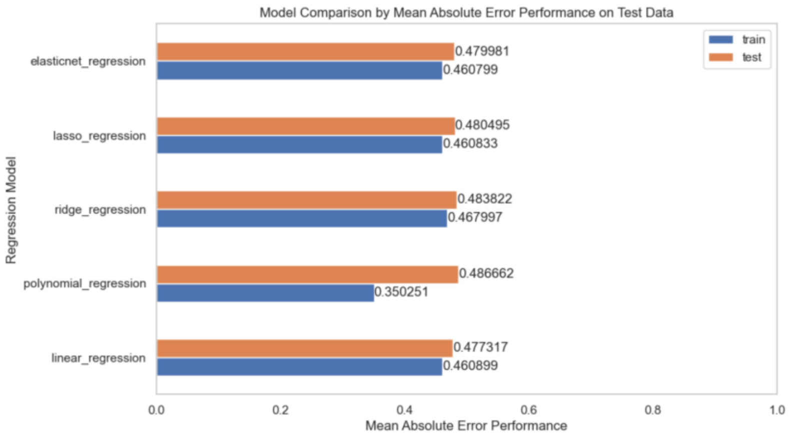
- R-Squared



- MSE



- MAE



Section 4

Summary

OVERALL FINDINGS AND IMPLICATIONS

- Key Findings
  - The linear regression model demonstrated the best independent test model performance among the candidate models. This model which did not apply any form of regularization proved to be superior to the penalized models potentially due to the presence of a smaller subset of equally informative predictors - leading to a minimal over-confidence in the parameter estimates and in effect, the minimal impact of the regularization constraints.
  - Among the penalized models, the optimal elastic net regression model demonstrated the best independent test model performance with the tuned hyperparameter leaning towards a higher L1 regularization effect (optimal L1 ratio equals 0.9 which is reminiscent of lasso where L1 ratio equals 1.0). The optimal lasso regression model using an L1 regularization term demonstrated an equally high independent test model performance, confirming the earlier findings.
- Overall Implications
  - The formulated models could provide robust and reliable estimates of cancer rates.
  - The computed r-squared metrics for the formulated models were relatively low (only ranging from 0.63 to 0.65) which could be further improved by:
    - Considering more informative predictors including data on lifestyle factors (smoking rates, obesity levels) or genetic diversity (infectious agents)
    - Considering more complex models other than linear regression and its variants

CONCLUSION

- Overall Summary
  - Data collection for the analysis involved world performance indicators and indices hypothesized to be directly influencing cancer rates across countries.
  - The quality of gathered data was assessed and potential issues were identified.
  - Appropriate pre-processing methods including remedial procedures to address duplicate, missing, outlying, and non-normalized data were applied to prepare the data for subsequent analysis. Additional data scaling and transformation were implemented.
  - EDA using visualization presented the various distributions and comparisons between the indicators and indices as evaluated against cancer rates – eventually identifying the key drivers of higher cancer rates. Statistical hypothesis testing identified the individual drivers that were significantly associated with cancer rates.
  - A final regression model was selected among candidate regression models that could provide robust and reliable estimates of cancer rates from an optimal set of observations and predictors.
  - Overall analysis findings were discussed and their practical implications were highlighted.

Section 5

Appendix

APPENDIX

- Source Data
    - Cancer Rates: World Population Review
    - Social Protection and Labor Indicator: World Bank
    - Education Indicator: World Bank
    - Economy and Growth Indicator: World Bank
    - Environment Indicator: World Bank
    - Climate Change Indicator: World Bank
    - Agricultural and Rural Development Indicator: World Bank
    - Social Development Indicator: World Bank
    - Health Indicator: World Bank
    - Science and Technology Indicator: World Bank
    - Urban Development Indicator: World Bank
    - Human Development Indices: Human Development Reports
    - Environmental Performance Indices: Yale Center for Environmental Law and Policy

APPENDIX

- Python Notebooks | Codes
  - GitHub URL: Data Background
  - GitHub URL: Data Description
  - GitHub URL: Data Quality Assessment
  - GitHub URL: Data Preprocessing
    - GitHub URL: Data Cleaning
    - GitHub URL: Missing Data Imputation
    - GitHub URL: Outlier Treatment
    - GitHub URL: Collinearity
    - GitHub URL: Shape Transformation
    - GitHub URL: Centering and Scaling
    - GitHub URL: Data Encoding
    - GitHub URL: Preprocessed Data Description
  - GitHub URL: Data Exploration
    - GitHub URL: Exploratory Data Analysis
    - GitHub URL: Hypothesis Testing

APPENDIX

- Python Notebooks | Codes
  - GitHub URL: Model Development
    - GitHub URL: Premodelling Data Description
    - GitHub URL: Linear Regression
    - GitHub URL: Polynomial Regression
    - GitHub URL: Ridge Regression
    - GitHub URL: Least Absolute Shrinkage and Selection Operator Regression
    - GitHub URL: Elastic Net Regression
  - GitHub URL: Consolidated Findings

APPENDIX

- ## References

  - [Book] Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python by Jason Brownlee
  - [Book] Feature Engineering and Selection: A Practical Approach for Predictive Models by Max Kuhn and Kjell Johnson
  - [Book] Feature Engineering for Machine Learning by Alice Zheng and Amanda Casari
  - [Book] Applied Predictive Modeling by Max Kuhn and Kjell Johnson
  - [Book] Data Mining: Practical Machine Learning Tools and Techniques by Ian Witten, Eibe Frank, Mark Hall and Christopher Pal
  - [Book] Data Cleaning by Ihab Ilyas and Xu Chu
  - [Book] Data Wrangling with Python by Jacqueline Kazil and Katharine Jarmul
  - [Python Library API] NumPy by NumPy Team
  - [Python Library API] pandas by Pandas Team
  - [Python Library API] seaborn by Seaborn Team
  - [Python Library API] matplotlib.pyplot by MatPlotLib Team
  - [Python Library API] itertools by Python Team
  - [Python Library API] operator by Python Team
  - [Python Library API] sklearn.experimental by Scikit-Learn Team
  - [Python Library API] sklearn.impute by Scikit-Learn Team
  - [Python Library API] sklearn.linear_model by Scikit-Learn Team
  - [Python Library API] sklearn.preprocessing by Scikit-Learn Team
  - [Python Library API] sklearn.metrics by Scikit-Learn Team
  - [Python Library API] sklearn.model_selection by Scikit-Learn Team
  - [Python Library API] sklearn.pipeline by Scikit-Learn Team
  - [Python Library API] scipy by SciPy Team

APPENDIX

- ## References
  - **[Article]** Step-by-Step Exploratory Data Analysis (EDA) using Python by Malamahadevan Mahadevan (Analytics Vidhya)
  - **[Article]** Exploratory Data Analysis in Python — A Step-by-Step Process by Andrea D'Agostino (Towards Data Science)
  - **[Article]** Exploratory Data Analysis with Python by Douglas Rocha (Medium)
  - **[Article]** 4 Ways to Automate Exploratory Data Analysis (EDA) in Python by Abdishakur Hassan (BuiltIn)
  - **[Article]** 10 Things To Do When Conducting Your Exploratory Data Analysis (EDA) by Alifia Harmadi (Medium)
  - **[Article]** How to Handle Missing Data with Python by Jason Brownlee (Machine Learning Mastery)
  - **[Article]** Statistical Imputation for Missing Values in Machine Learning by Jason Brownlee (Machine Learning Mastery)
  - **[Article]** Imputing Missing Data with Simple and Advanced Techniques by Idil Ismiguzel (Towards Data Science)
  - **[Article]** Missing Data Imputation Approaches | How to handle missing values in Python by Selva Prabhakaran (Machine Learning +)
  - **[Article]** Master The Skills Of Missing Data Imputation Techniques In Python(2022) And Be Successful by Mrinal Walia (Analytics Vidhya)
  - **[Article]** How to Preprocess Data in Python by Afroz Chakure (BuiltIn)
  - **[Article]** Easy Guide To Data Preprocessing In Python by Ahmad Anis (KDNuggets)
  - **[Article]** Data Preprocessing in Python by Tarun Gupta (Towards Data Science)
  - **[Article]** Data Preprocessing using Python by Suneet Jain (Medium)
  - **[Article]** Data Preprocessing in Python by Abonia Sojasingarayar (Medium)
  - **[Article]** Data Preprocessing in Python by Afroz Chakure (Medium)
  - **[Article]** Detecting and Treating Outliers | Treating the Odd One Out! by Harika Bonthu (Analytics Vidhya)
  - **[Article]** Outlier Treatment with Python by Sangita Yemulwar (Analytics Vidhya)
  - **[Article]** A Guide to Outlier Detection in Python by Sadrach Pierre (BuiltIn)
  - **[Article]** How To Find Outliers in Data Using Python (and How To Handle Them) by Eric Kleppen (Career Foundry)

APPENDIX

- ## References

  - **[Article]** Statistics in Python — Collinearity and Multicollinearity by Wei-Meng Lee (Towards Data Science)
  - **[Article]** Understanding Multicollinearity and How to Detect it in Python by Terence Shin (Towards Data Science)
  - **[Article]** A Python Library to Remove Collinearity by Gianluca Malato (Your Data Teacher)
  - **[Article]** 8 Best Data Transformation in Pandas by Tirendaz AI (Medium)
  - **[Article]** Data Transformation Techniques with Python: Elevate Your Data Game! by Siddharth Verma (Medium)
  - **[Article]** Data Scaling with Python by Benjamin Obi Tayo (KDNuggets)
  - **[Article]** How to Use StandardScaler and MinMaxScaler Transforms in Python by Jason Brownlee (Machine Learning Mastery)
  - **[Article]** Feature Engineering: Scaling, Normalization, and Standardization by Aniruddha Bhandari (Analytics Vidhya)
  - **[Article]** How to Normalize Data Using scikit-learn in Python by Jayant Verma (Digital Ocean)
  - **[Article]** What are Categorical Data Encoding Methods | Binary Encoding by Shipra Saxena (Analytics Vidhya)
  - **[Article]** Guide to Encoding Categorical Values in Python by Chris Moffitt (Practical Business Python)
  - **[Article]** Categorical Data Encoding Techniques in Python: A Complete Guide by Soumen Atta (Medium)
  - **[Article]** Categorical Feature Encoding Techniques by Tara Boyle (Medium)
  - **[Article]** Ordinal and One-Hot Encodings for Categorical Data by Jason Brownlee (Machine Learning Mastery)
  - **[Article]** Hypothesis Testing with Python: Step by Step Hands-On Tutorial with Practical Examples by Ece Polat (Towards Data Science)
  - **[Article]** 17 Statistical Hypothesis Tests in Python (Cheat Sheet) by Jason Brownlee (Machine Learning Mastery)
  - **[Article]** A Step-by-Step Guide to Hypothesis Testing in Python using Scipy by Gabriel Rennó (Medium)
  - **[Publication]** Ridge Regression: Biased Estimation for Nonorthogonal Problems by Arthur Hoerl and Robert Kennard (Technometrics)
  - **[Publication]** Regression Shrinkage and Selection Via the Lasso by Rob Tibshirani (Royal Statistical Society Journal)
  - **[Publication]** Regularization and Variable Selection via the Elastic Net by Hui Zou and Trevor Hastie (Royal Statistical Society Journal)

Thank You!