

# Poverty Probability Prediction

The background features several stylized human figures in light green, some with their arms raised. On the right side, there is a cluster of hexagons in various shades of green and teal, arranged in a somewhat geometric pattern.

By:-  
Vaibhav Arora

# Agenda

- Introduction
- About Dataset
- Methodology
- Exploratory Data Analysis
- Target Variable
- Numerical Variables
- Categorical Variables
- Feature Engineering
- Regression
- Gradient Boosting

# Introduction

This project presents an analysis on data concerning the Poverty Probability Index (PPI) of individuals. By exploring the relationships between characteristics of the individuals and their PPI, we extract useful information that would allow us to make prediction of an individual's probability of poverty based on its socioeconomic indicators.

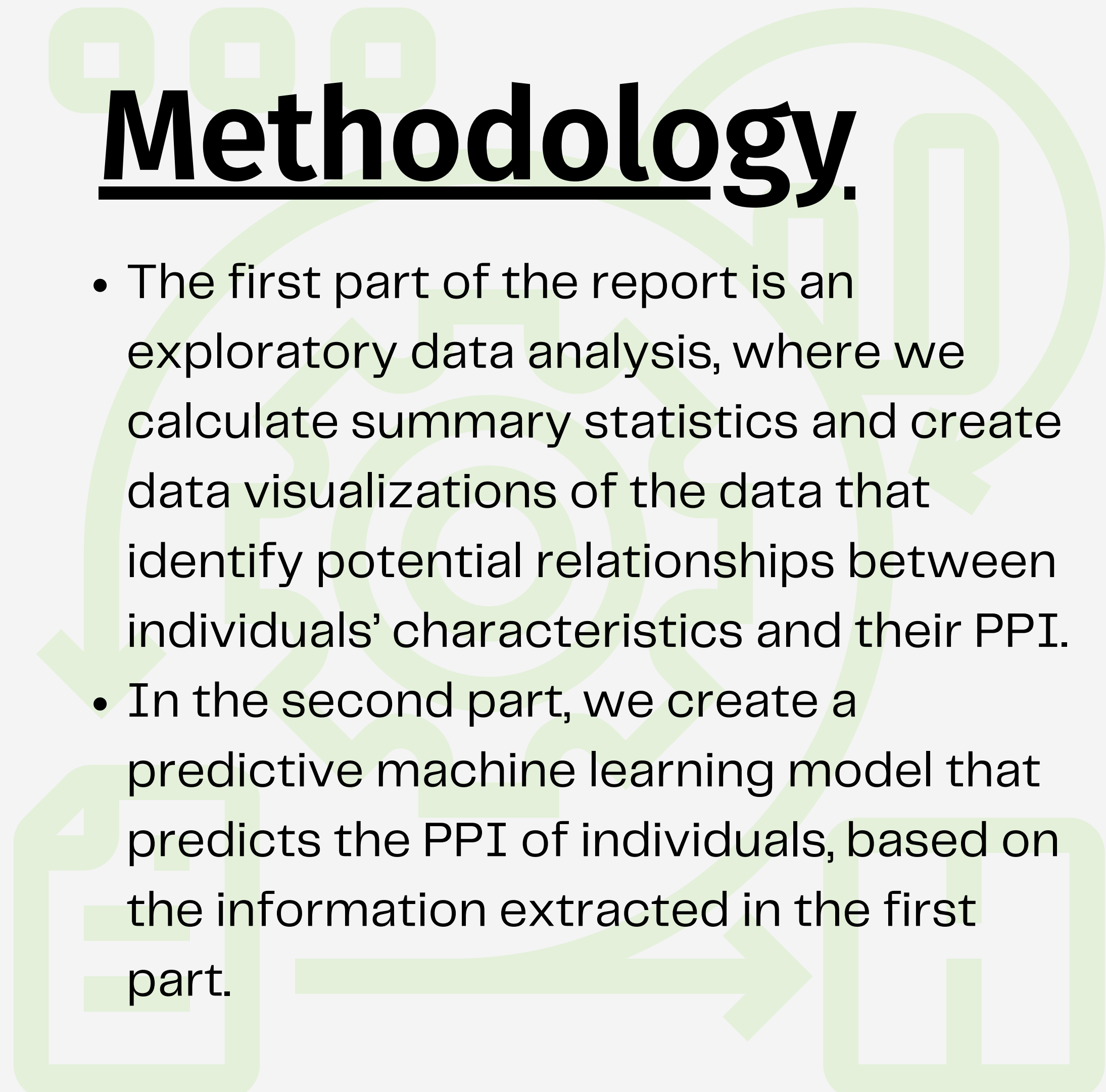


# About Dataset

The dataset retrieved from the PPI website and Financial Inclusion Insights household surveys conducted by InterMedia contains PPI data along with **59** features of 12,600 individuals across 7 different countries. Original sources of the dataset include data from The World Bank. The PPI is a measure to identify an individual's probability of living below the poverty line at the \$2.50/day threshold.



# Methodology.

- The first part of the report is an exploratory data analysis, where we calculate summary statistics and create data visualizations of the data that identify potential relationships between individuals' characteristics and their PPI.
  - In the second part, we create a predictive machine learning model that predicts the PPI of individuals, based on the information extracted in the first part.
- 

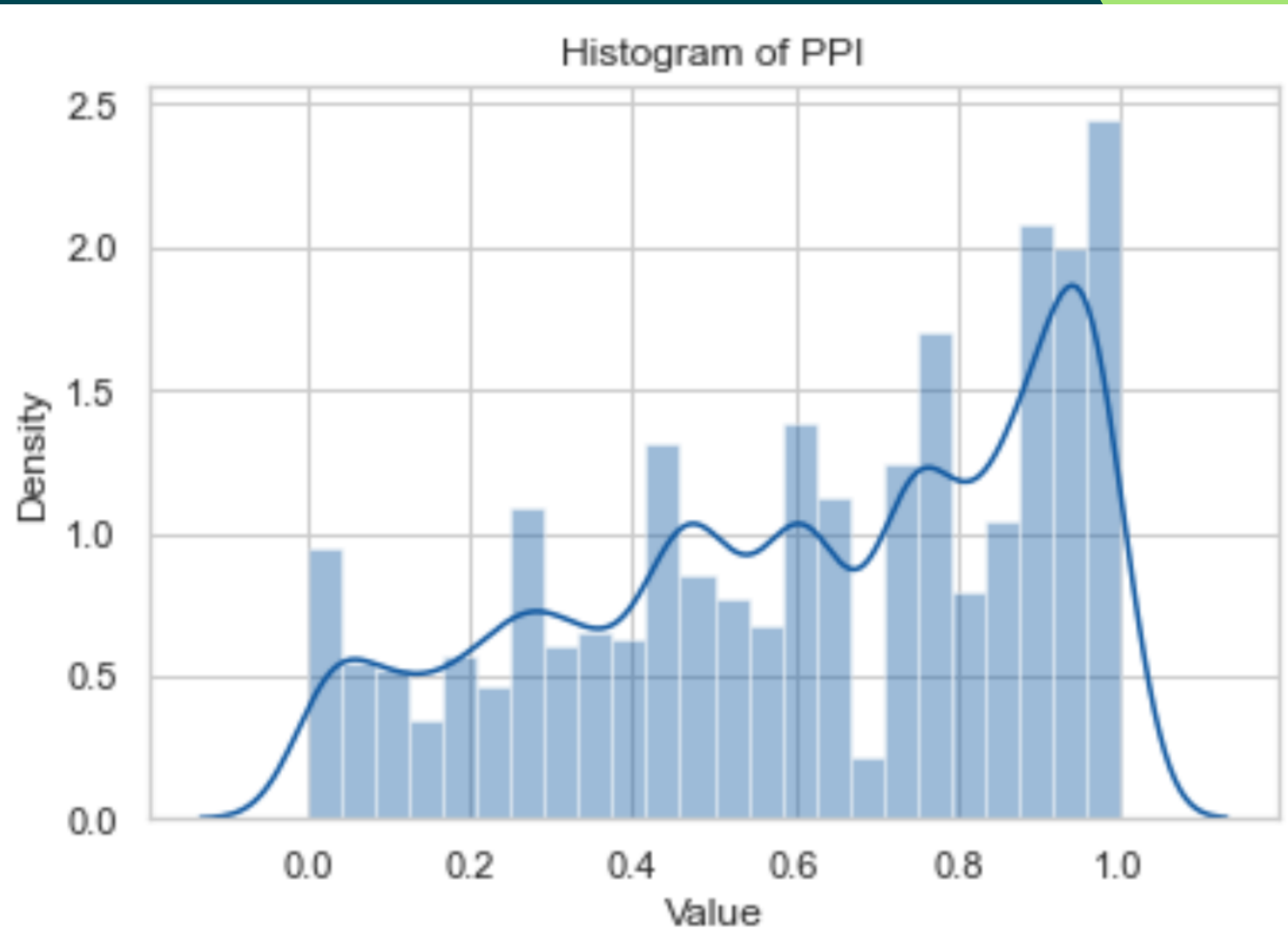


# Exploratory Data Analysis

- Data Pre-processing
- Exploring PPI
- Exploring Numeric Features
- Exploring Categorical Features



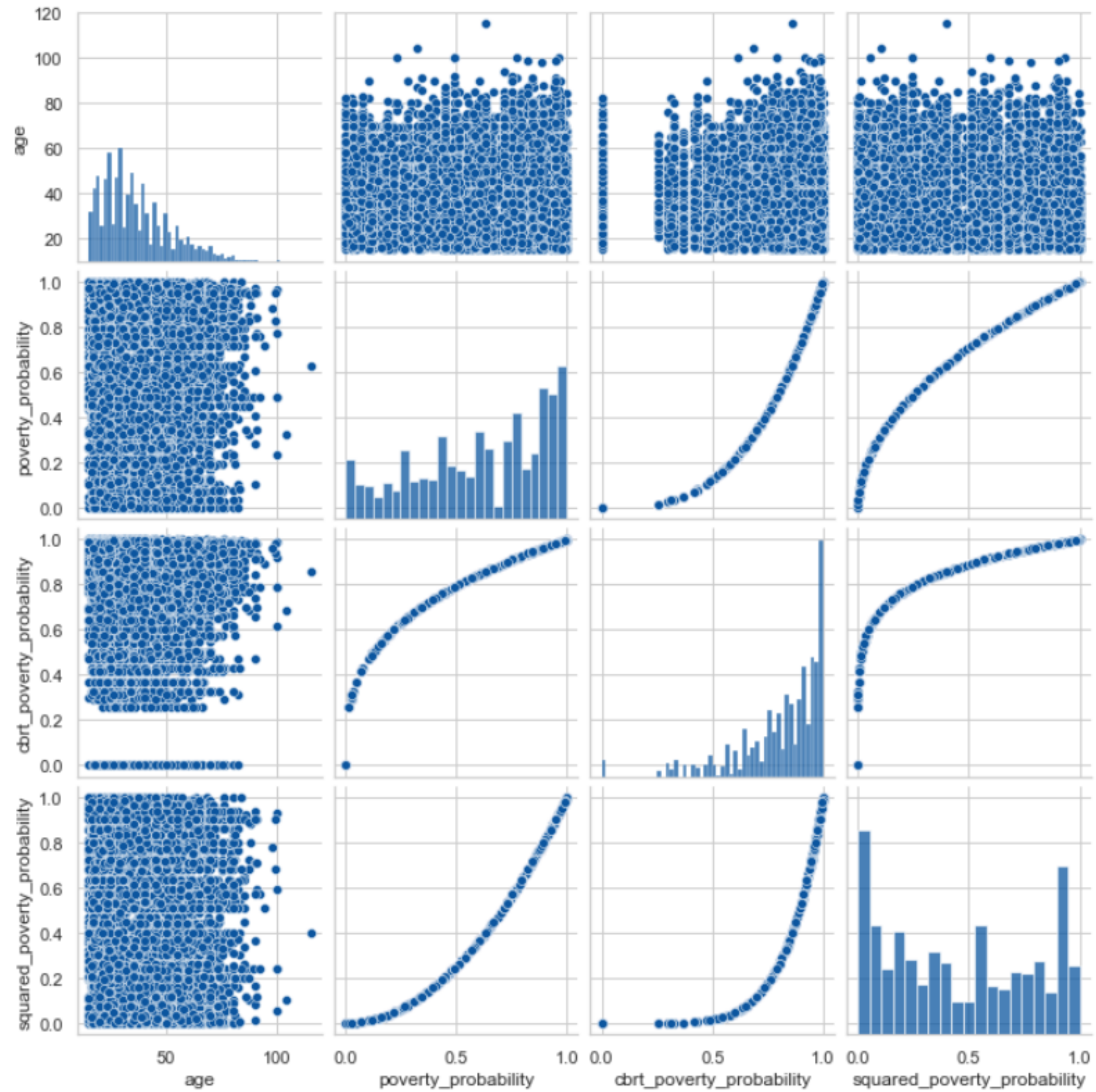
# Target Variable





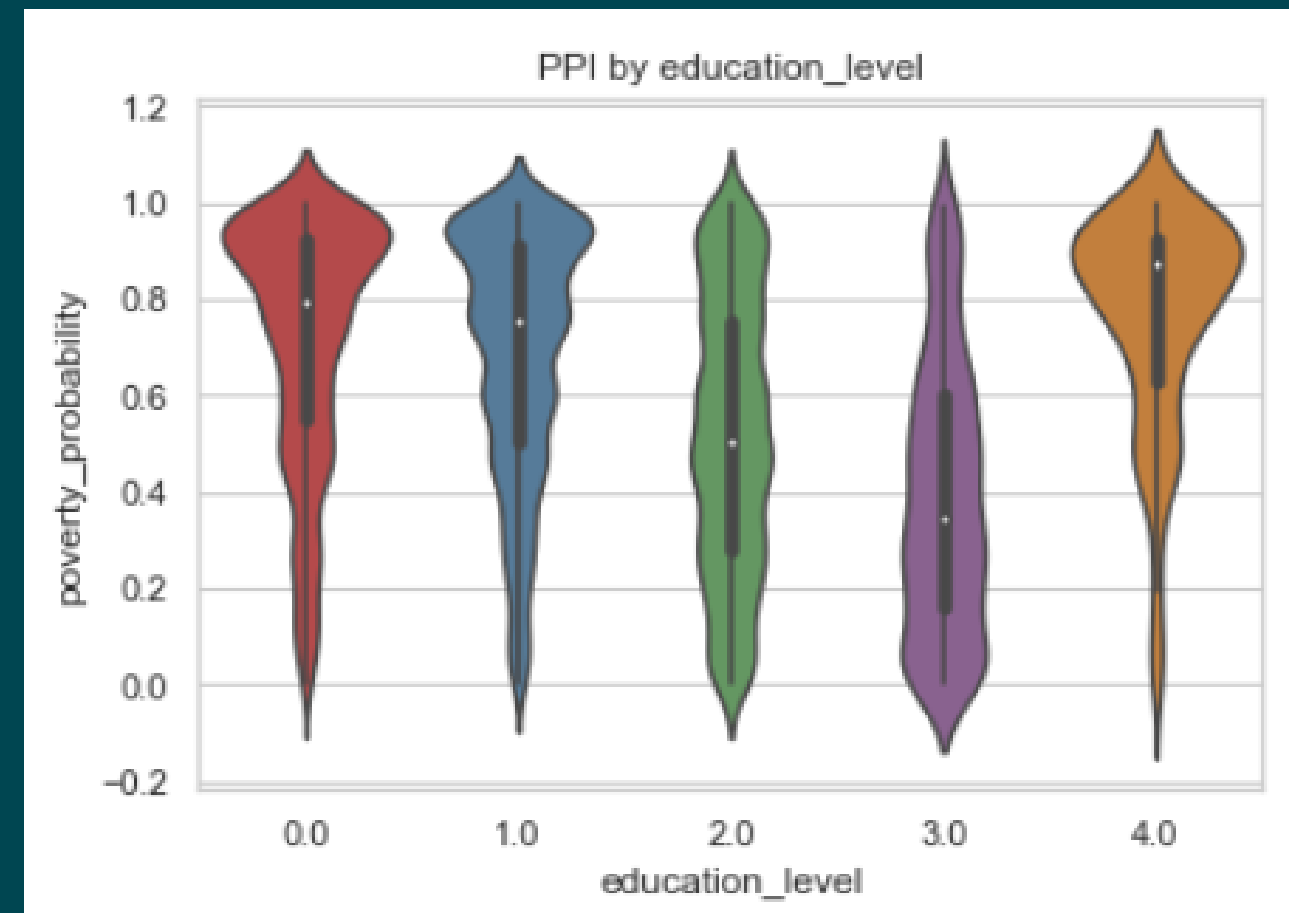
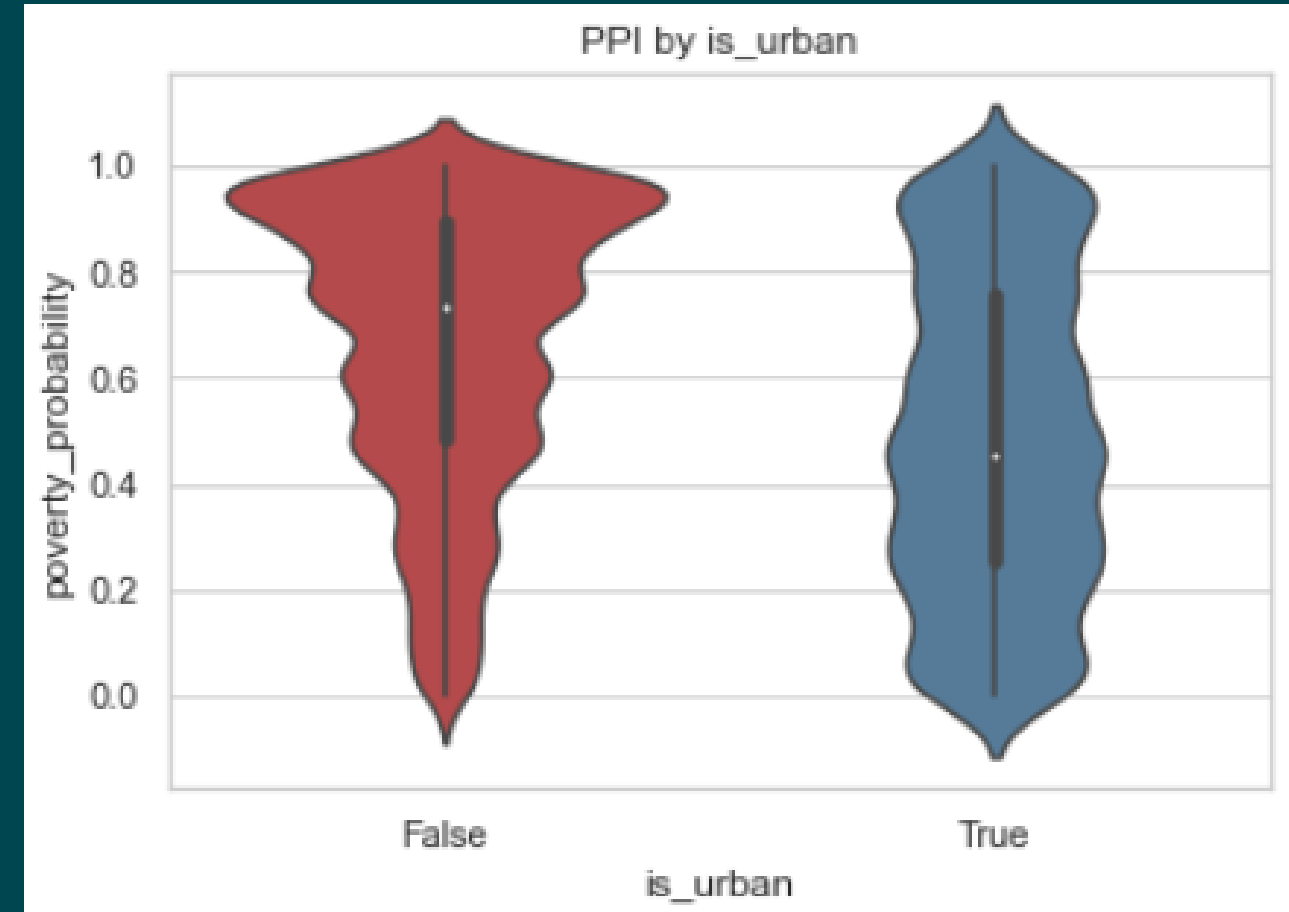
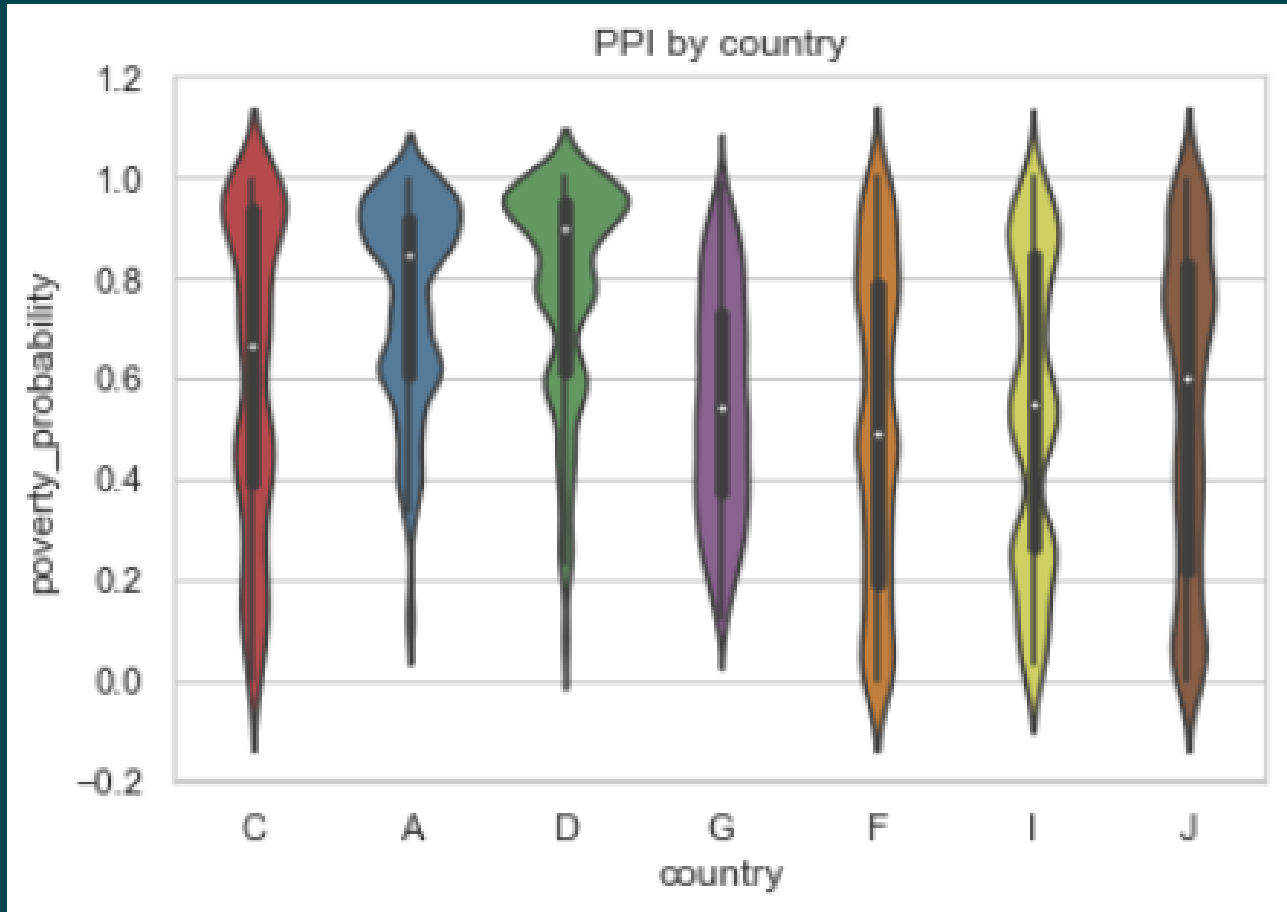
# Numerical Variables

Scatter plot to  
check Linear  
relationship  
with Target  
Variable

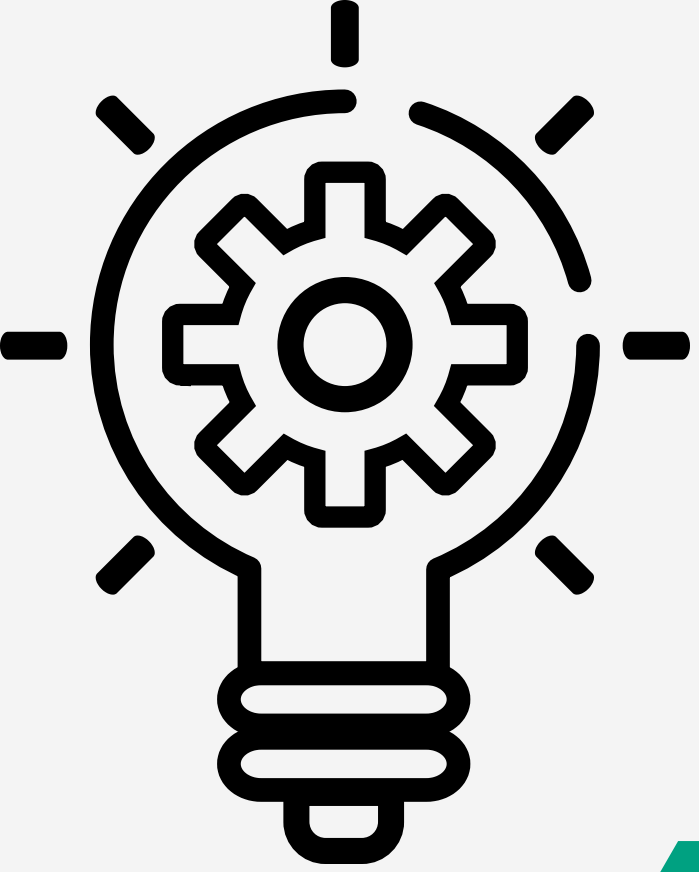




# Categorical Variables



Violin plot to  
check spread  
within the  
variable



# Feature Engineering

- Variables Encoding
- Eliminating Low variance features
- Recursive Feature Elimination
- Rescaling Numeric Features

# Regression

## OLS Regression Results

```
=====
Dep. Variable:      poverty_probability    R-squared:                0.187
Model:              OLS                   Adj. R-squared:           0.187
Method:             Least Squares         F-statistic:             362.9
Date:               Tue, 31 Jan 2023       Prob (F-statistic):      0.00
Time:               14:30:53              Log-Likelihood:          -1037.8
No. Observations:   12600                 AIC:                     2094.
Df Residuals:       12591                 BIC:                     2161.
Df Model:           8
Covariance Type:    nonrobust
=====
```

```
=====
               coef      std err          t      P>|t|      [0.025      0.975]
-----
const          0.7766      0.007    113.068      0.000      0.763      0.790
education_level -0.0479      0.003   -16.415      0.000     -0.054     -0.042
is_urban       -0.1254      0.005   -24.073      0.000     -0.136     -0.115
phone_technology -0.0334      0.003   -13.323      0.000     -0.038     -0.029
formal_savings -0.0746      0.006   -13.200      0.000     -0.086     -0.064
literacy       -0.0020      0.006    -0.355      0.722     -0.013      0.009
has_investment -0.0452      0.005    -8.545      0.000     -0.056     -0.035
married        0.0305      0.005     6.151      0.000      0.021      0.040
female        -0.0018      0.005    -0.362      0.717     -0.011      0.008
=====
```



# Gradient Boosting

```
GBoost = GradientBoostingRegressor(objective='regression', num_leaves = 32,  
                                   learning_rate=0.01)
```

## Hyperparameter Tuning

```
GBoost = GradientBoostingRegressor(n_estimators= 2000, learning_rate=0.01,  
                                   max_depth=5, max_features='sqrt',  
                                   min_samples_leaf=7, min_samples_split=15,  
                                   loss='ls', random_state = 1)
```

## Result

Mean Square Error	=	0.04660316097887786
Root Mean Square Error	=	0.21587765280102028
Mean Absolute Error	=	0.17159834294081766
Median Absolute Error	=	0.1416639805849445
R^2	=	0.4357342204782534
Adjusted R^2	=	0.4114345761427413