Happiness Score Prediction

ISYE 6740: Spring 2025 - Team 26 Project Report

Ashria Arora, Jonathan Papir, Sadhana Gaddam

1 INTRODUCTION

Happiness is often seen as one of the best ways to measure a country's well-being as it reflects the overall quality of life experienced by its people. The World Population Review publishes World Happiness Rankings each year which rank countries based on factors like GDP per capita, social support, life expectancy, freedom to make life choices, generosity, and perceptions of corruption. While these factors are important and offer a solid foundation for understanding national well-being, they don't always capture the full picture of what truly makes people happy.

1.1 Problem Statement

Previous research focused mostly on basic linear regression models to predict happiness scores using a limited set of variables, and this project goes a step further by leveraging a broader mix of socioeconomic, health, environmental, and technological factors in addition to considering more advanced machine learning models. The goal is to accurately predict a country's happiness score under the assumption that it's a complex variable, and accurate predictions require a holistic view of human nature including more information than what has been used in other analyses. For example, health factors such as birth rate, death rate from suicide, cancer and cardiovascular disease rates; socioeconomic indicators such as GDP per capita, unemployment and inflation rates; technological factors such as electricity and internet access; and environmental factors like CO_2 emissions and material footprint, will be used. Additionally, the project's secondary goal is to identify which features are most predictive of happiness and whether certain feature categories, if any, are more predictive than others.

2 EXPLORATORY DATA ANALYSIS

2.1 Data Source

Data sourced from 2016-2021 Gallup World Polls [1], [2], [3], [4], [5], [6], [7], [8], [9] contain happiness scores representing the target feature predicted by all models built in this project for 149-157 countries. Additional data from the same years compiled by organizations such as the International Labour Organization [12], International Monetary Fund [13], Our World in Data [10], Institute for Health Metrics and Evaluation [11], and more augmented the happiness scores, constituting a rich set of features used to predict them.

2.2 Pre-Processing and Challenges

Since the data came from different sources, pre-processing was essential to ensure it was ready for later steps. First, country names were standardized so they matched between datasets. Next, all datasets were joined together, but each had a different number of observations, corresponding to a different set of countries spanning different time ranges, resulting in a post-join dataset with many rows containing at least one null value. Unfortunately, there was no easy way to source the missing information, and too much was missing for imputation to be a viable strategy.

A decision was made to remove features with > 10% missing observations (5 features in total), and then rows containing at least one missing value were dropped. The end result was a single dataset with 744 observations and 23 features from 164 unique countries between 2016 and 2021 where each row represents a snapshot of a particular country's features and happiness score for a particular year. Although losing a chunk of data was not ideal, more than enough from countries all across the globe remained to paint an accurate picture of happiness.

2.3 Target Feature

The World Population Review rates happiness on a 0-10 scale. Happiness scores, visualized in Figure 1, range from a minimum of 2.52 to a maximum of 7.89 with a mean of 5.45 and standard deviation of 1.15. The scores show significant variability, which likely reflects the diverse socioeconomic, health, environmental, and

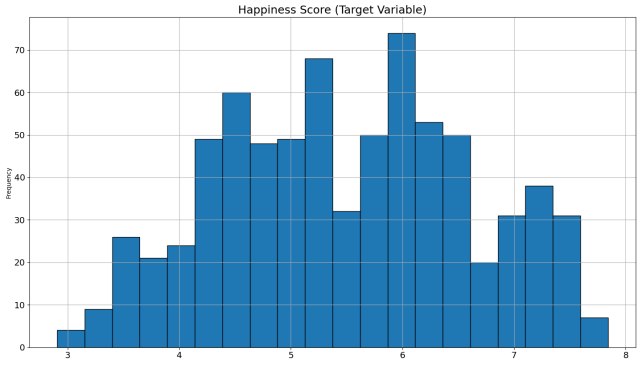

Figure 1: Happiness Score Distribution

technological diversity present across the countries included in the study.

2.4 Data Distribution

All indicators in Figure 2 exhibit a pronounced right skew, showing countries with high GDP and healthcare expenditure per capita, number of internet users, and annual CO_2 emission are a minority. Additionally, health measures such as anxiety, Alzheimer's, cardiovascular disease, and cancer rates are right skewed but to a lesser degree, likely capturing the disparities in health outcomes and healthcare quality between countries (Appendix Figure 17). In contrast, indicators such as the civil liberties index are left skewed – a majority of countries respect the freedom of their citizens, but there are still many with varying degrees of oppression.

2.5 Outlier Detection

The significant right skew in Figure 2 was indicative of outliers, but the boxplots in Appendix Figure 18 confirm it. GDP per capita, healthcare spending, CO_2 emissions, access to electricity, and health measures like HIV and cardiovascular disease rates have a handful of outliers. However, outliers make sense in this context – some countries are simply more developed and have much higher GDP while others are less developed and have much lower access to electricity. Consequently, these outliers are reflective of the true state of the world, and it did not make sense to remove them.

2.6 Log-Transform Features

Given the skewed nature of many variables, a log transformation was applied to stabilize their variance and make their distributions closer to normal. Log transformations help reduce the influence of extreme values and outliers, making the data more suitable for linear modeling techniques by preventing them from overwhelming a model's coefficients. Additionally, checking for correlation between the log-transformed features, rather than the original untransformed features, provides better insight into how a model will perceive relationships between variables and which features are most strongly correlated.

2.7 Correlation Analysis

The correlation heatmap in Figure 3 provides a detailed view into the relationships between log transformed variables. Strong positive correlations between average lifespan and GDP per capita are immediately noticeable, as well as between healthcare expenditure per capita, human development index, and electricity access percentage, suggesting countries with higher economic performance and better healthcare also experience higher overall development and quality of life. Conversely, strong negative correlations are visible between birth rate and GDP, lifespan, and human development, indicating that countries with higher birth rates tend to have lower economic performance and developmental indicators. Additionally, the gender inequality

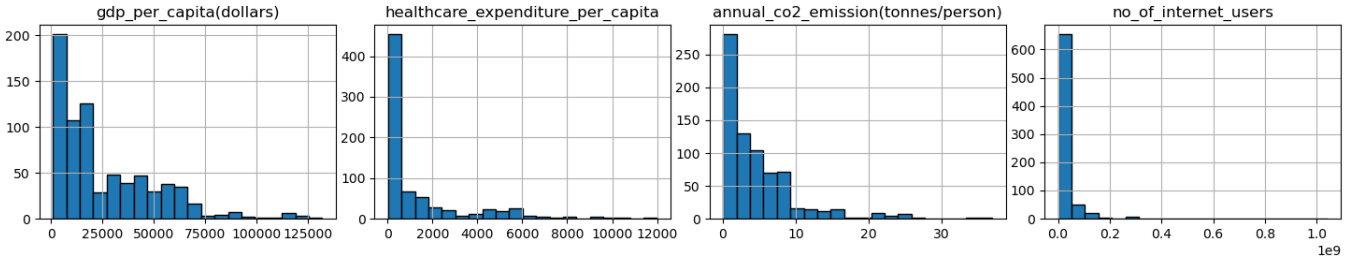

Figure 2: Significant Right Skew in Four Features

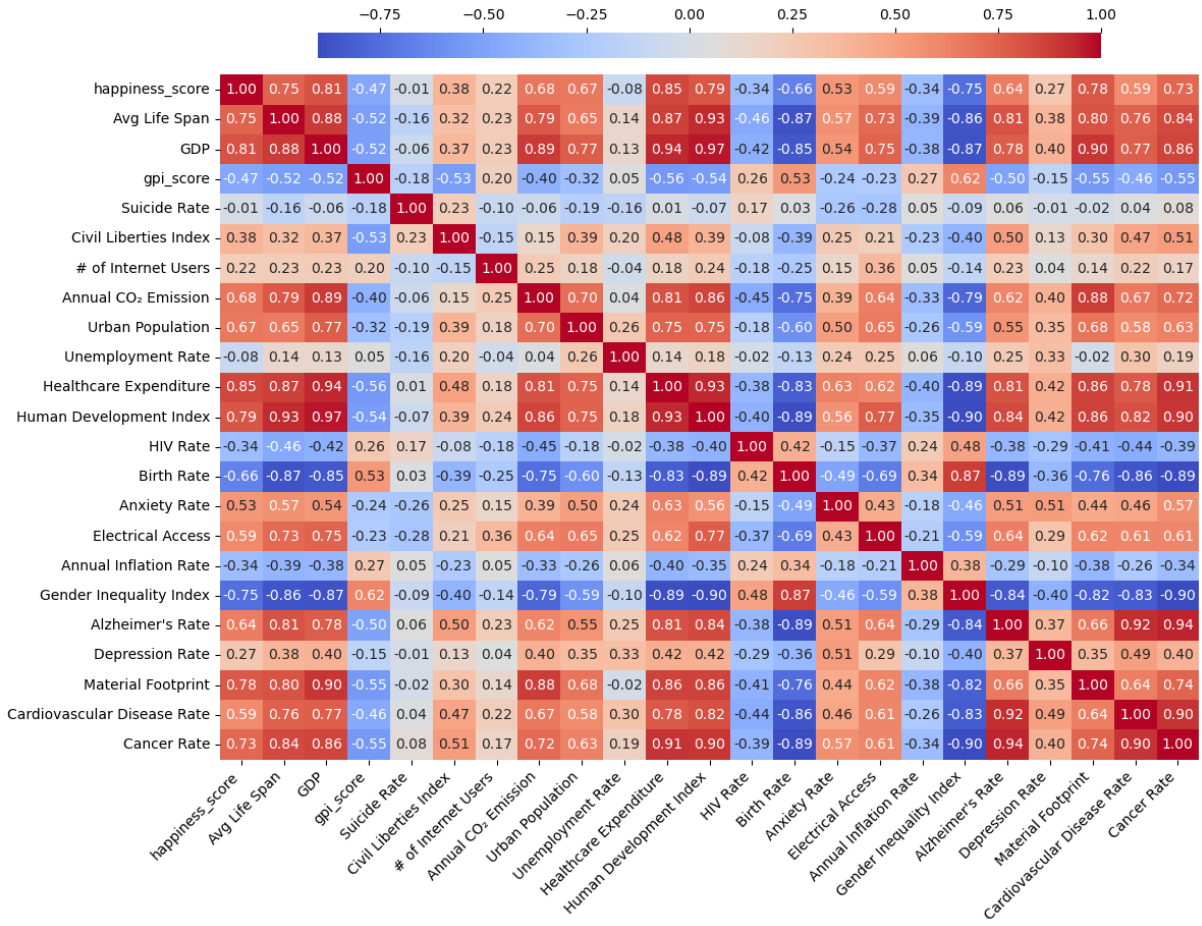

Figure 3: Correlation Matrix of Log Transformed Variables

index shows significant negative correlations with human development and healthcare expenditure, implying that lower gender inequality is associated with better overall living conditions. However, such strong correlations also indicate potential multicollinearity which can affect the reliability of predictive models.

2.7.1 Variance Inflation Factor

The Variance Inflation Factor (VIF) is a metric used to determine whether variables contain overlapping information, otherwise known as multicollinearity, which potentially affects the stability and interpretability of a regression model. VIF scores above 10 are generally considered too high for a variable to be useful, although even VIF scores between 5 and 10 are considered questionable. VIF scores below 5 are ideal, but many of this dataset's variables had scores in the hundreds or thousands, as seen in Table 1.

In an attempt to reduce multicollinearity, the group experimented with dropping variables with high correlations found in Figure 3 and high VIF scores in Table 1. However, some VIF scores were still > 1000 even after removing the most strongly correlated variables, and the group was faced with a decision: continue removing variables until VIF scores dropped to an acceptable level or keep the data as is and work around the multicollinearity. Removing variables went against the main goal of this project – using a new and diverse set of variables to predict happiness scores – and the group decided to overcome the multicollinearity in a different way, but the decision came with a consequence.

Variable	VIF
Avg Life Span(yrs)	5015.406
Cardiovascular Disease Rates	4120.116
Depression Rates	2609.480
GPD Per Capita(USD)	2175.559
Anxiety Rates	2001.141
Human Dev Index	1433.700
HIV Rate	1279.305
Cancer Rates	1118.591
Alzheimer's Rates	859.605
⋮	⋮

Table 1: Top 9 Variance Inflation Factors

Unfortunately, significant multicollinearity renders measures of feature importance meaningless because a change in one variable correlates to a change in the others. Consequently, it is difficult to disentangle each feature's individual impact on happiness in isolation, and any model's attempt to do so will likely be distorted. For example, many of the socioeconomic, health, and environmental variables in this dataset move together — wealthier countries with higher GDP tend to spend more on healthcare and also emit more CO_2 — but it is unclear which feature is really driving the changes in happiness. The resulting ambiguity limits how confidently one can say whether GPD, healthcare spending, or CO_2 emissions plays the bigger role in explaining happiness. As such, the project's secondary goal of identifying which features or feature categories are most predictive of happiness was no longer possible.

3 METHODOLOGY

Since there was no way to remove multicollinearity from the dataset, the group decided to build models or use statistical techniques that weren't susceptible to it. Three models were built in total: a linear regression model trained on the log-transformed features transformed to PCA space, a ridge regression model trained on log-transformed features, and a random forest model trained on the original features. Afterward, XGBoost was applied to each model to improve its predictive accuracy, resulting in six models for consideration. The flowchart in Figure 4 lays out the steps in the model building process.

3.1 PCA - Linear Regression

Principal Component Analysis (PCA) was conducted on log-transformed data to address multicollinearity among the features. However, PCA is sensitive to differences in magnitude, and the data was scaled first to ensure no single variable dominates the process. PCA transforms the original correlated features into a new set of principal components (PCs), where each PC is an orthogonal (i.e. uncorrelated) linear combination of the initial features in the new feature space. Mathematically speaking, this means the covariance between any two PCs is zero, alleviating the multicollinearity issue.

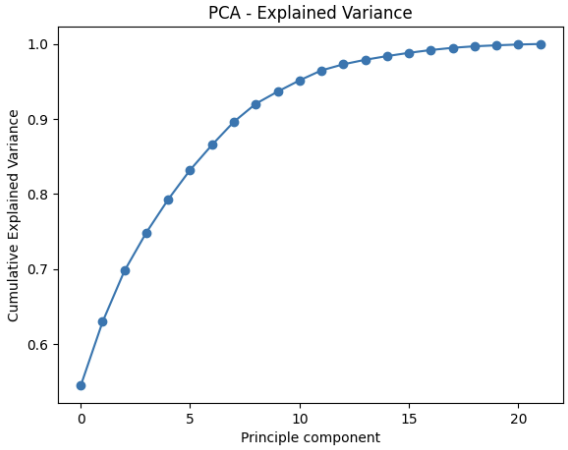

Figure 5: PCA Cumulative Variance Explained

Linear regression was used to model the relationship between the dependent variable, happiness scores, and the independent variables, the top k PCs. Figure 5 represents the cumulative variance explained by the top k PCs, and the first five explain close to 80% of the data's variance.

Linear regression models were built using the top k PCs ranging from those that explained as little as 80% to as much as 100% of the total variance in the data. The models were implemented using the `LinearRegression` class within the `sklearn.linear_model` module, and the optimal number of PCs was determined based on metrics such as Mean Square Error (MSE) and adjusted R^2 value.

Figure 6 shows the validation MSE was lowest when using 17 PCs. Furthermore, the training and validation

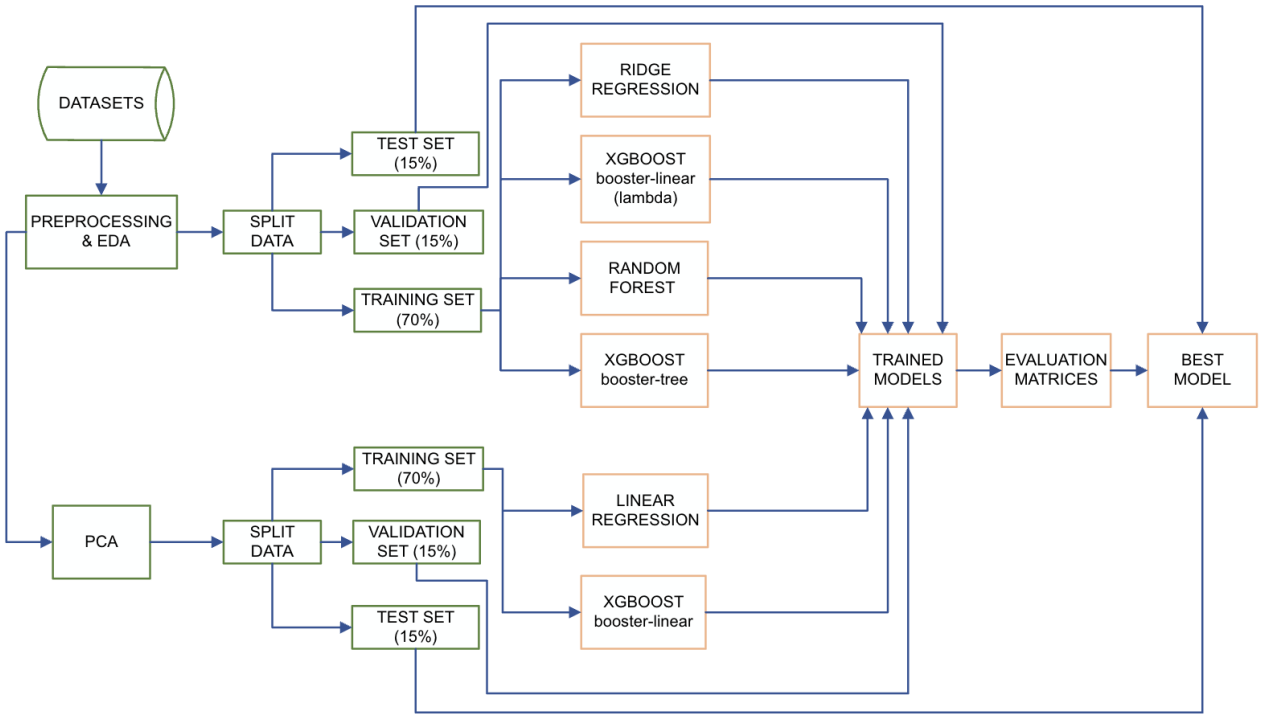

Figure 4: Model Building Process - Training, Validation, and Testing

errors were very close, indicating a well-balanced model — neither underfitting nor overfitting – and the model using the top 17 principal components was selected as best. Evaluating the model on validation data led to the results shown in Table 2.

	Linear Reg
MSE	0.257
Adjusted R^2	0.790

Table 2: Linear Regression Validation Results

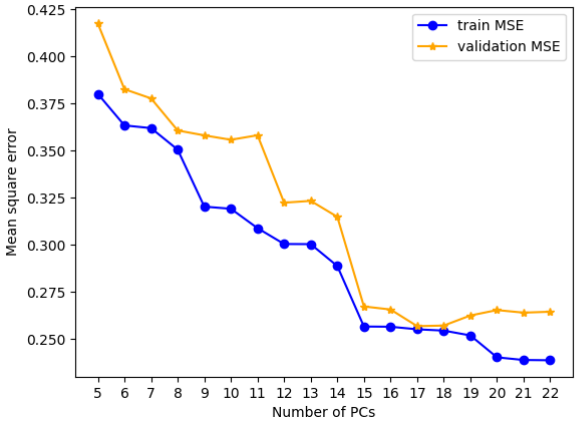

Figure 6: PCA Linear Regression Model

3.2 Ridge Regression

Ridge regression was selected because it has a L_2 penalty it its loss function, helping to address the high multicollinearity among the features. The L_2 penalty shrinks all coefficients towards zero, preventing any single feature from exerting too much influence on the model, which stabilizes coefficient estimates when predictors are highly correlated and improves generalization on unseen data. Prior to fitting, each log transformed feature was standardized so that the regularization penalty applies uniformly across features.

A grid search (`GridSearchCV`) using five-fold cross-validation was conducted to identify the optimal regularization parameter, α, which controls the L_2 penalty.

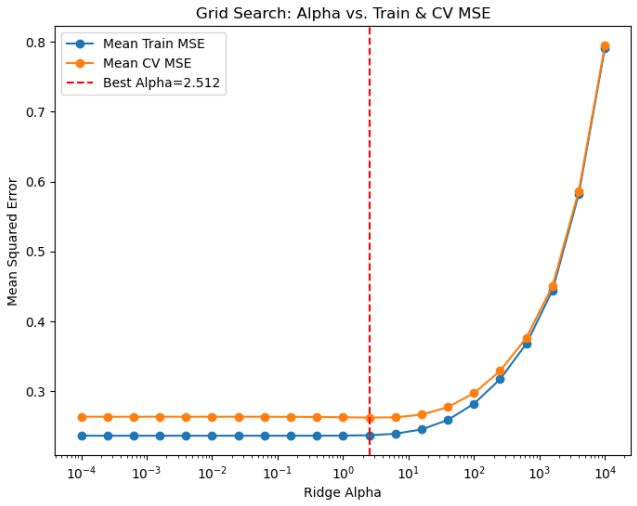

Figure 7: L_2 Penalty: Training & CV Errors

Figure 7 plots the mean training and cross validation MSE against α over values spanning 10^{-4} to 10^4 on a logarithmic scale. When α is small, the model is barely regularized, and the training error sits just below the CV error. Once $\alpha \geq 10$, both the training and the CV errors increase sharply, showing that the model has become too regularized. As α increases, coefficients smoothly shrink toward zero, preserving correlated predictors in the model while reducing their combined variance (Figure 8). The grid search ultimately decided a model with $\alpha = 2.512$ was optimal, and its performance on a held-out validation set is shown in Table 3.

	Ridge Reg
MSE	0.263
Adjusted R^2	0.738

Table 3: Ridge Regression Validation Results

3.3 Random Forest

Random forest regression models are robust to multicollinearity due to their ensemble structure and tunable hyperparameters, making them a natural choice for this dataset. Each decision tree is trained on a different bootstrap sample of the data, which helps mitigate the influence of correlated predictors, and the final prediction is obtained by averaging all tree outputs, tamping down multicollinearity's effect even further. Random forest models also have a `max_features` hyperparameter which limits the number of features considered when splitting at each node, reducing the likelihood

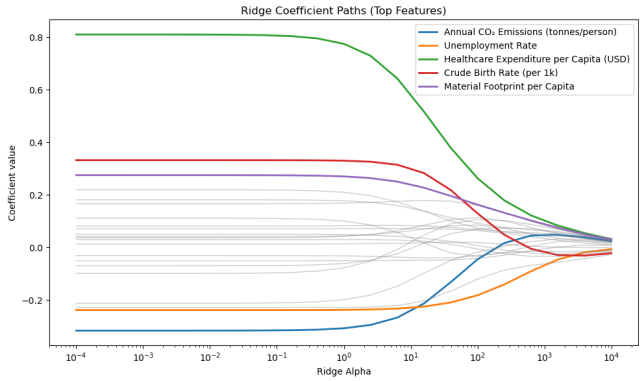

Figure 8: L_2 Penalty's Effect on Coefficient Paths

that highly correlated features are selected together, providing another layer of protection against multicollinearity and minimizing its impact.

Unlike linear or ridge regression, random forest models do not have coefficients and aren't susceptible to outliers or differences in magnitude among features. Consequently, log transformations to remove outliers or feature scaling weren't necessary, and all random forest models were built using the raw features from the original dataset.

Random forest models have more tunable hyperparameters than most machine learning models, and tuning began with `n_estimators`, which controls the number of trees. While increasing the number of trees increases the model's complexity, it doesn't increase its bias, and averaging predictions over more and more trees decreases its variance monotonically toward zero. Consequently, prediction error converges beyond a certain number of trees, and additional trees beyond that point don't diminish the model's performance but are superfluous.

Figure 9 compares out-of-bag (OOB), validation, and cross-validation errors when constructing SKlearn `RandomForestRegressor` models over forests containing 20 to 1,000 trees. All errors converge by roughly 300 to 400 trees, but validation error takes longer to smooth out, finally calming down around 500 or 600 trees and suggesting at least 600 should be more than enough for future random forest models trained on this data.

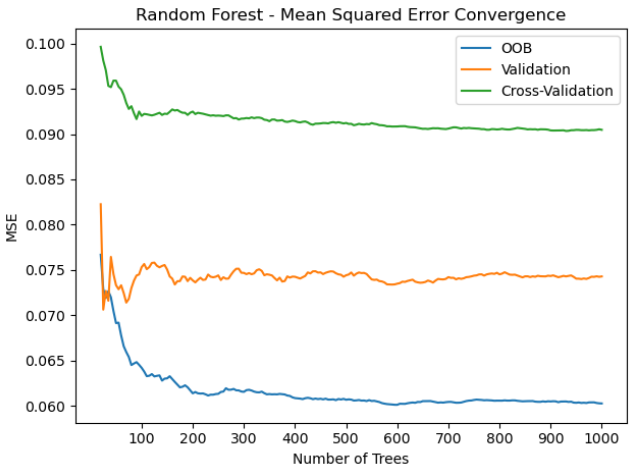

Figure 9: Tuning the Number of Trees

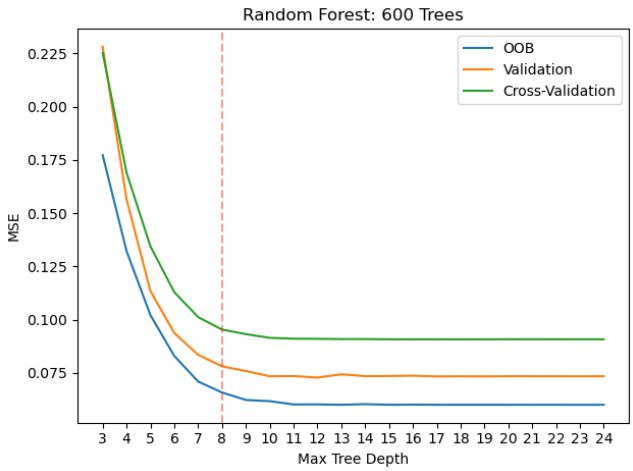

Figure 11: Tuning the Max Tree Depth

All remaining hyperparameters were tuned using five fold cross-validation, starting with `max_features` and `max_depth`, the maximum tree depth, using random forests with 600 trees. As mentioned previously, `max_features` helps mitigate multicollinearity's impact, and the lowest validation, OOB, and cross-validation errors occur at 9, 12, and 15 features, respectively, as seen in Figure 10, while `max_depth` minimizes overfitting by limiting tree depth – deeper trees become highly specific to the training data and sometimes lead to models that don't generalize well. Furthermore, increasing tree depth beyond a certain point leads to diminishing returns. For this dataset, diminishing returns began around a max depth of 8 as seen in Figure 11.

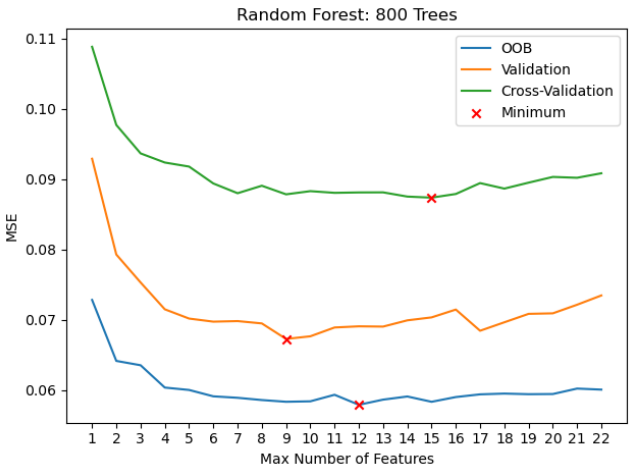

Figure 10: Tuning the Max Number of Features

Using Figure 10 and Figure 11's results as a guideline, cross-validation considering `max_feature` values from 5-20 and `max_depth` values from 5-10 was performed one last time. Additionally, the number of trees was increased to 800 to ensure convergence now that multiple hyperparameters were being tuned simultaneously. The model with the lowest cross-validation error had a maximum depth and max features of 10 and 15, respectively, and was selected as the best random forest model. The optimal model's results on validation data are shown in Table 4.

	Random Forest
MSE	0.071
Adjusted R^2	0.929

Table 4: Random Forest Validation Results

3.4 XGBoost

XGBoost is an efficient implementation of gradient boosting that builds a sequence of decision trees, each one learning to fix the mistakes of the trees that came before it. Rather than fitting one complex tree, XGBoost adds trees sequentially, with each new tree focusing on the observations that the ensemble is getting wrong. XGBoost penalizes overly complex trees to prevent overfitting, charging a "cost" for having too many leaves or excessively large leaf weights, and it uses a learning rate to scale down each tree's impact.

3.4.1 XGBoost - PCA Linear Regression

An XGBoost model using the `gblinear` booster parameter was trained on the training dataset, and no L_1 (Lasso) or L_2 (Ridge) regularization tuning was applied. Performance on the validation set for the linear regression models with or without XGBoost remained about the same, as seen in Table 5.

	Linear Reg	XGBoost Linear Reg
MSE	0.257	0.256
Adjusted R^2	0.790	0.791

Table 5: Linear Regression Validation Comparison

3.4.2 XGBoost - Ridge Regression

Using the same `gblinear` booster but with a L_2 regularization penalty (`reg_lambda`) unlocks access to Ridge Regression in XGBoost. Figure 12 shows how varying the L_2 penalty affects both training and five-fold cross-validation errors. At a very low penalty, the model is almost unregularized and both errors are small. As the penalty increases, the CV errors reach a minimum while training error remains a bit lower. Past $\lambda = 1$, the errors rise and the model becomes too regularized. A model using $\lambda = 0.01$ was selected as best through five-fold cross-validation (`GridSearchCV`), and its performance on the validation set is shown in Table 6.

	XGBoost Ridge Reg
MSE	0.262
Adjusted R^2	0.739

Table 6: XGB Ridge Regression Validation Results

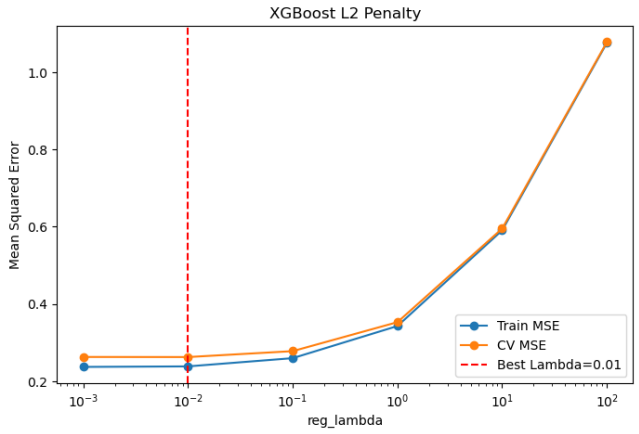

Figure 12: XGBoost L_2 Penalty: Train & CV Errors

3.4.3 XGBoost - Random Forest

XGBoost's `XGBRegressor` function builds tree models similar to `RandomForestRegressor` when its `booster` parameter is changed to `'gbtree'`. `XGBRegressor` also has `n_estimators`, `max_depth`, and `colsample_bynode` hyperparameters that control the number of trees in the forest, as well as the maximum tree depth and number of features used in splitting at each node, but it has many new hyperparameters, too. Of note are `reg_alpha` and `reg_lambda`, which apply L_1 and L_2 regularization penalties to decision tree weights.

Despite many similarities, `XGBRegressor` models do not perform bootstrap sampling. Instead, they sample *without replacement* using a `colsample` parameter, which ranges from 0 to 1 and dictates the proportion of data sampled for training. Furthermore, `XBGRegessor` models don't track which observations are or aren't present in the subsample, making it difficult to calculate OOB error.

The path to tuning the tree based `XGBRegressor` was similar to tuning the random forest model but with different results. Cross-validation error converged much more quickly, by roughly 100 trees, and shallower trees of depths between 3-6 performed the best, as shown in in Figure 13 and 14, respectively.

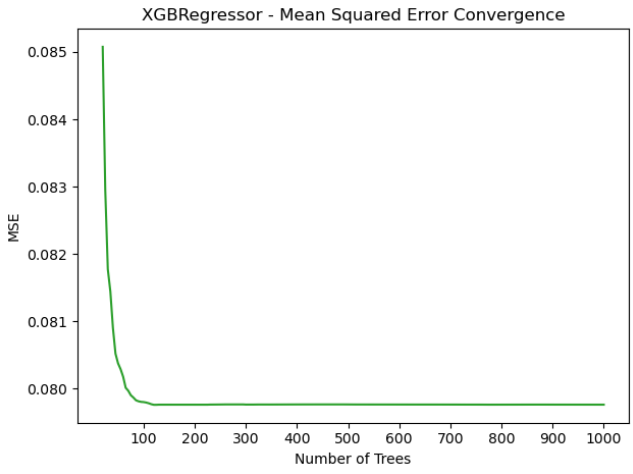

Figure 13: Tuning the Number of Trees

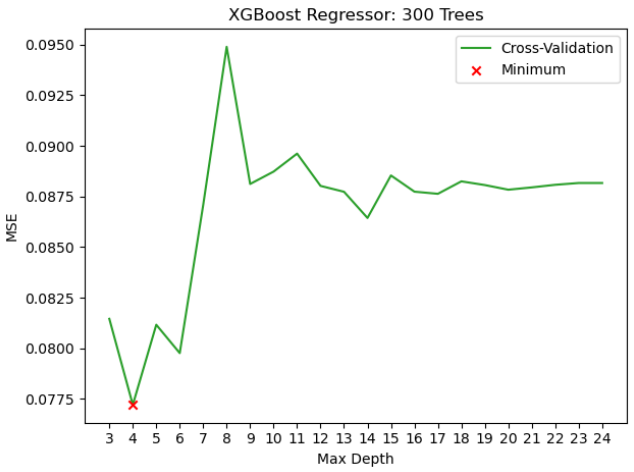

Figure 14: Tuning Max Tree Depth

Subsequent tuning focused on the new `colsample_bynode`, `subsample`, `reg_alpha`, and `reg_lambda` parameters. `colsample_bynode` works the same as `max_features` but takes on a value from 0-1, representing the maximum proportion of features available, instead of the maximum number of features, when splitting at each node. `XGBRegressor` favored a much lower number of features than before – 35% to be exact or roughly 8 features – and cross-validation error was lowest when `subsample = 0.8`, meaning only 80% of the training data was sampled for each tree, as seen in Figure 15. Finally, Figure 16 shows cross-validation error grows as L_1 and L_2 penalties increase, indicating the dataset doesn't benefit much from regularization penalties on decision tree weights and little to no regularization is needed to perform well.

One final cross-validation round used a grid-search to examine `max_depth` ranging from 3-7, `colsample_bynode` from 0.25 - 0.70 in increments of 0.05, and five equally spaced values of `reg_alpha` between 10^{-2} and 10^{0} and `reg_lambda` between from 10^{-1} to 10^{0} in log space. A grid search found a model with `colsample_bytree=0.65`, `max_depth=4`, `reg_alpha=0.316`, `reg_lambda=0.316`, and `subsample=0.8` was best, and its results on validation data are shown in Table 7.

	XGBoost Random Forest
MSE	0.062
Adjusted R^2	0.938

Table 7: XGB Random Forest Validation Results

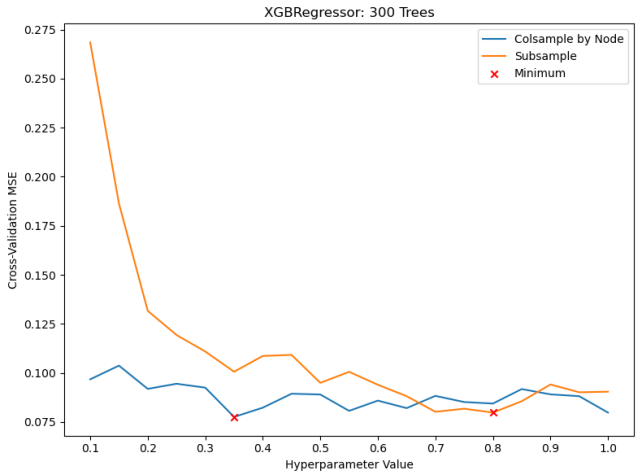

Figure 15: Tuning Max Proportion of Features in Each Split and Max Proportion of Data Sampled

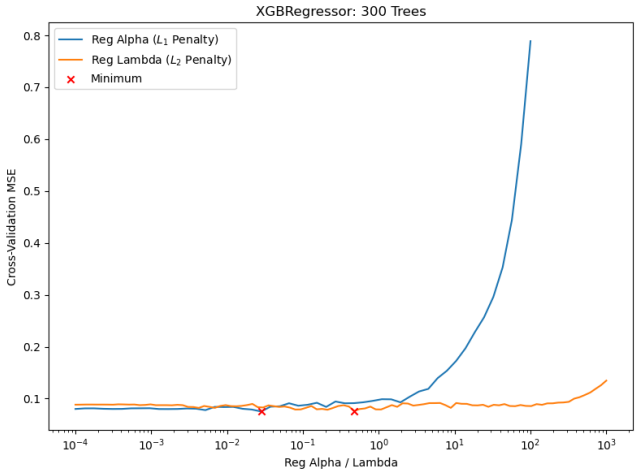

Figure 16: Tuning Regularization Parameters

4 EVALUATION AND FINAL RESULTS

All models used the same 70/15/15 data split for training, validation, and testing, respectively. However, a log-transformation was applied to the partitions prior to use in the two ridge regression models, and a log-transformation as well as PCA were applied prior to use in the two linear regression models. No transformations were needed for the two random forest models.

Two metrics were calculated to evaluate the models' performance:

(1) Mean square error (MSE) explains how close the predicted values are from true/original values. The lower the MSE, the better the model.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

(2) Adjusted R^2 measures the proportion of variance explained by the features. However, it adjusts for R^2's tendency to increase as more features are added by penalizing models with more features, providing a more accurate assessment of model fit.

$$Adjusted\ R^2 = 1 - \frac{(1-R^2)(n-1)}{n-p-1}$$

y_i = actual value
$\hat{y}_i$ = predicted value
p = number of features
n = number of data points

Cross-validation MSE using the training partition determined the optimal hyperparameters for all ridge regression and random forest models, but the two linear regression models had no tunable hyperparameters, and cross-validation wasn't necessary. Table 8 shows the MSE and adjusted R^2 results for all six models on validation data. The XGBoost-random forest model scored highest on both metrics, and it was chosen as best.

Interestingly, XGBoost-random forest's MSE dropped from 0.062 for validation data to 0.048 for testing data, and adjusted R^2 increased from 0.938 to 0.957, which was not to be expected. MSE is typically higher and adjusted R^2 typically lower when calculated on testing data, so the group performed another 70/15/15 split using a different random seed for comparison.

In the new split, the testing error increased slightly from 0.053 to 0.068 and adjusted R^2 decreased from 0.936 to 0.930. The reversal suggests the difference is due to randomness in the sampling process. Certain seeds produce training partitions that are more similar to testing partitions and less similar to validation partitions, leading to a lower testing error and a higher validation error in some cases. However, all metrics were within the same order of magnitude regardless of the seed, suggesting that the training process is stable, and the model is generalizing well. Overall, the results are promising, and although one's definition of happiness can vary significantly from person to person, it seems the right combination of predictors and model can predict a nation's collective happiness with high accuracy.

Metric		**Seed 6740**	**Seed 42**
MSE	Validation	0.062	0.053
	Test	0.048	0.068
Adjusted R^2	Validation	0.938	0.936
	Test	0.957	0.930

Table 9: XGB-Random Forest Results by Random Seed

	Linear Reg	XGBoost Linear Reg	Ridge Reg	XGBoost Ridge Reg	Random Forest	XGBoost Tree
MSE	0.257	0.256	0.263	0.262	0.071	0.062
Adjusted R^2	0.790	0.791	0.738	0.739	0.929	0.938

Table 8: Validation Results: All Models

5 DISTRIBUTION OF WORK

- Data Collection - Ashria Arora, Jonathan Papir, Sadhana Gaddam
- Data Preprocessing - Ashria Arora, Jonathan Papir, Sadhana Gaddam
- Data Cleaning - Sadhana Gaddam
- Exploratory Data Analysis - Ashria Arora
- Model Building
  - PCA Linear Regression / XGB-Linear Regression: Sadhana Gaddam
  - Ridge Regression / XGB-Ridge Regression: Ashria Arora
  - Random Forest / XGB-Random Forest: Jonathan Papir
- Report Writing: Ashria Arora, Jonathan Papir, Sadhana Gaddam
- Report Editing: Jonathan Papir

6 SOURCES

- Random Forests
  - Overfitting
    - https://medium.com/@alexmolasmartin/can-random-forests-overfit-a743755251b4
- XGBoost Random Forest
  - Tuning
    - https://gabrieltseng.github.io/posts/2018-02-25-XGB/#: :text=reg_alpha%20and%20reg_lambda&text=reg_-alpha%20and%20reg_lambda%20control%20the,at%20the%20leaves%20can%20become.
    - https://machinelearningmastery.com/random-forest-ensembles-with-xgboost/
    - https://machinelearningmastery.com/xgboost-for-regression/
    - https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/
  - Cross-Validation
    - https://xgboost.readthedocs.io/en/stable/python/examples/cross_validation.html
- Scientific Paper
  - https://scispace.com/pdf/machine-learning-for-the-analysis-of-quality-of-life-using-3mb5u57i.pdf

REFERENCES

[1] Helliwell, J. F., Layard, R., Sachs, J. D., De Neve, J.-E., Aknin, L. B., & Wang, S. (Eds.). (2024). World Happiness Report 2024. University of Oxford: Wellbeing Research Centre.

[2] Helliwell, J. F., Layard, R., Sachs, J. D., Aknin, L. B., De Neve, J.-E., & Wang, S. (Eds.). (2023). World Happiness Report 2023 (11th ed.). Sustainable Development Solutions Network.

[3] Helliwell, J. F., Layard, R., Sachs, J. D., De Neve, J.-E., Aknin, L. B., & Wang, S. (Eds.). (2022). World Happiness Report 2022. New York: Sustainable Development Solutions Network.

[4] Helliwell, John F., Richard Layard, Jeffrey Sachs, and Jan-Emmanuel De Neve, eds. 2021. World Happiness Report 2021. New York: Sustainable Development Solutions Network.

[5] Helliwell, John F., Richard Layard, Jeffrey Sachs, and Jan-Emmanuel De Neve, eds. 2020. World Happiness Report 2020. New York: Sustainable Development Solutions Network

[6] Helliwell, J., Layard, R., & Sachs, J. (2019). World Happiness Report 2019, New York: Sustainable Development Solutions Network.

[7] Helliwell, J., Layard, R., & Sachs, J. (2018). World Happiness Report 2018, New York: Sustainable Development Solutions Network.

[8] Helliwell, J., Layard, R., & Sachs, J. (2017). World Happiness Report 2017, New York: Sustainable Development Solutions Network.

[9] Helliwell, J., Layard, R., & Sachs, J. (2016). World Happiness Report 2016, Update (Vol. I). New York: Sustainable Development Solutions Network.

[10] Our World in Data Retrieved from: 'https://ourworldindata.org/' [Online Resource]

[11] Institute for Health Metrics and Evaluation (IHME). (2021). *GBD Compare Data Visualization Tool.* Seattle, WA: IHME, University of Washington.

[12] International Labor Organization : 'https://www.ilo.org/data-and-statistics' [Online Resource]

[13] International Monetary Fund: 'https://www.imf.org/en/Data' [Online Resource]

7 APPENDIX

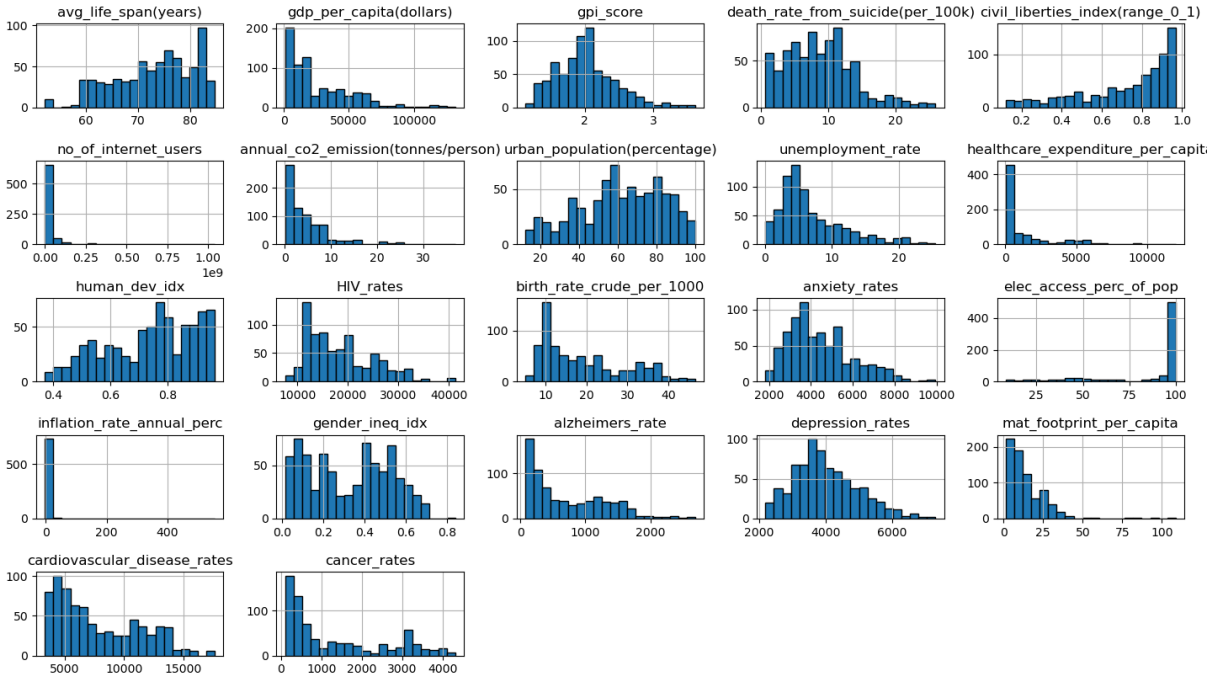

Figure 17: Distribution of Raw Features

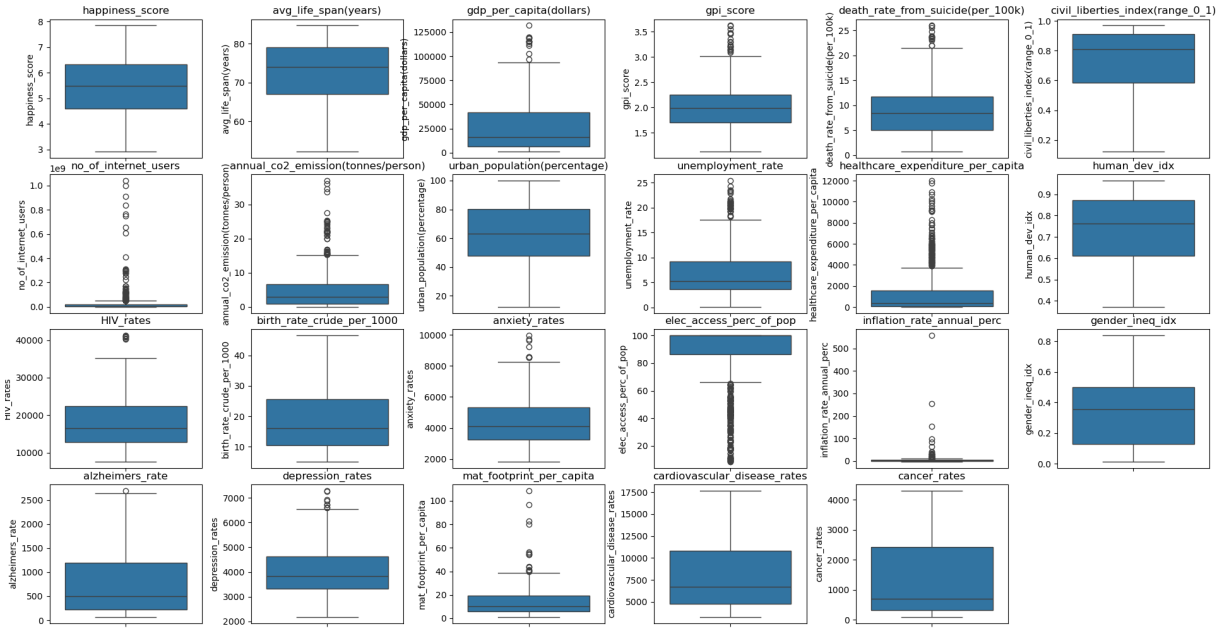

Figure 18: Boxplots of Raw Features