Predicting Life Expectancy With Machine Learning Techniques

Sarvesh Krishan, Gabriel Jones, Kieran Sullivan, Manroop Samra

STA 160 Fall Quarter 2024

1 Introduction

Public health has been a focal point of many fields of research. With this, it is important for us to consider the progress countries have made, with an important metric to determine this being life expectancy. Life expectancy is a valuable metric for us to delve deeper into as it acts as a general indicator of the health and economic status of a country. This, in turn, may reflect a country's focus on healthcare, nutrition, and socio-economic conditions. A result of Life Expectancy being a great overarching indicator for the wellbeing of a society is that there are a significant number of factors that play into determining life expectancy. While it is difficult to identify at a glance which combination of factors will best determine the life expectancy of a country's people, we expect the main indicators to include information about income, education, and access to healthcare, as these are historically strongly tied to the life expectancy of a population. We also believe that the status of a country (whether it is developed or developing) has an impact on life expectancy and its underlying influencing factors. This is supported by the idea that developing countries have relatively less access to education and healthcare compared to developed countries.

In this study, we aim to answer two questions of interest. First, which factors most contribute to the life expectancy of a given country, and how does this differ, if at all, for countries of differing status? Through this, we hope to provide insight into what countries can focus on to improve the health and well-being of their population.

To revisit our earlier notion on the disparities between developing and developed countries, we constructed yearly box plots for each of our factors split by status.

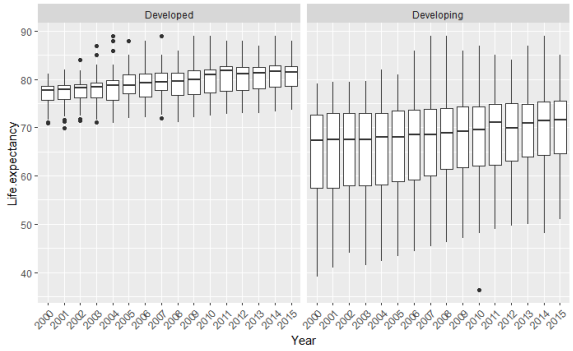

Figure 1. Yearly Boxplot of Life expectancy split by country status. Further boxplots for other variables are given in the supplementary material.

From the result of constructing this plot for life expectancy, we note a considerable difference in the average yearly life expectancy between developed countries and developing countries. We also notice similar trends when constructing plots for schooling and income values. These preliminary plots confirmed the existence of disparities between countries of differing statuses. To further explore the effects of these differences and how they influence the factors in determining life expectancy, we employed Recursive Feature Elimination (RFE) utilizing three machine learning algorithms: Support Vector Machines (SVMs), K-Nearest Neighbors (KNN), and Decision Trees. Our second question of interest aims to understand which one of these three machine learning techniques best predicts life expectancy. This set of algorithms was chosen keeping in mind the

work done by Kasichainula Vydehi et al. Their work was done using the same data set used in this study, the machine learning algorithms employed however were Random Forest, Multiple Linear Regression, and Decision Trees. Through our choice of algorithms, we are able to compare our results with decision trees with the work done by Kasichainula Vydehi et al., as well as provide insight into the performance of novel machine learning algorithms. Additionally, the "status variable" was omitted from the analysis done by Kasichainula Vydehi et al. By including status as a variable in our analysis and splitting the data set by status, we can provide further insight as to how the status of a country impacts which features are most associated with life expectancy. Our second question of interest aims to understand which one of these three models best predicts life expectancy. To determine this, we used each model's MSE and R^2 as performance metrics. After evaluating these models, we find that Support Vector Machines performed best and that our models overall performed better in forecasting the life expectancy of developing countries in comparison to developed countries. With RFE, we observe that Adult Mortality and Income are the most important variables to predict life expectancy. We also observe that key variables for determining life expectancy differed between countries of differing statuses. Most notably, the prevalence of thinness within children ages 5-19 was a uniquely important factor in determining life expectancy for developed countries, while the deaths by HIV/AIDS was a uniquely important factor for developing countries.

2 Literature Review

The usage of machine learning algorithms is not new to health applications, but researchers in the last five years have used Machine Learning techniques to predict life expectancy.

The paper by Kasichainula Vydehi, Keerthi Manchikanti, T. Satya Kumari, and SK Ahmad Shah explores the use of machine learning (ML) techniques to predict life expectancy based on a variety of socioeconomic, health, and demographic indicators. The study contributes to the growing body of literature emphasizing the importance of data-driven approaches in global health and development analytics. The primary goal of their paper is to analyze how various features, such as immunization rates, economic indicators, and health statistics, influence life expectancy. Their research also compares the performance of multiple machine learning models to determine the most effective approach for prediction.

Our study builds on what was done by Kasichainula Vydehi et al., which used the same data set on life expectancy from Kaggle, which originates from the World Health Organization. For classification techniques, life expectancy was split into five groups: above 80, above 70, above 60, above 50, above 40. The regression models used by Vydehil et al. were Multiple Linear Regression, Decision Trees, and Random Forests, with Random Forests having the lowest MSE and greatest R-Squared value. Backward elimination was conducted to determine which features were statistically significant by comparing to an alpha value of 0.05. They found that the most predominant feature that positively affects life expectancy is the "Income composition of resources". The machine learning methods for classification were K-Nearest Neighbor, Decision Trees, and Random Forest with accuracy scores of 81.2%, 87.4% and 89% respectively. This aligns with previous literature that highlights the multifaceted nature of life expectancy,

emphasizing the interactions between health, economic stability, and social well-being. However, what this research paper did not cover is how model results and feature importance towards life expectancy might differ across developed and developing countries, which we aim to answer in our study, alongside studying differing Machine Learning models, such as Support Vector Machines.

3 Materials & Methods

To answer our questions of interest, we needed a thorough understanding of the data set we were working with, as well as a plan on how to create and evaluate our models. Our data set came from Kaggle where it had been originally sourced from the World Health Organization's (WHO) Global Health Observatory (GHO). It contains 22 variables that come from both the WHO and the United Nations (UN). The UN collects data on economic factors like GDP and income composition of resources, and the WHO collects data on health factors like disease rates, alcohol use, and mortality rates. The data was all collected between 2000 and 2015, and comes from 193 countries. Each observation represents the data collected for the various variables for one country in one year, which gives us 2938 observations in total. In regards to our questions of interest relating developed vs. developing countries, we found that there were nearly five times as many observations from developing countries than from developed countries, at 2426 to 512.

Our next step was to clean the data. We first observed a significant amount of missing data, likely due to barriers to collect certain data in some countries. Looking into the missing values, we found that the only variable with over our threshold value of 20% of its values as N/As was population, so this was removed as a variable. Our intent by setting this high threshold value was to not alter our original data set too much, as we wanted to build models for the entire data set, not a smaller, albeit cleaner, subset. Country name was also removed as a predictive variable as this was deemed irrelevant to the modeling we aimed to do. The variable "status", which denotes whether a country was developing or developed, was changed to a numeric variable through one-hot encoding. Finally, all of our data was scaled through centering by subtracting the mean and dividing by the standard deviation. Table 1 of the supplemental materials lists our 17 predictive variables and more detail on what the metric used was.

We began our analysis with building and evaluating three types of regression models: K-Nearest Neighbor, Support Vector Machines, and Decision Trees. KNN works by making regression predictions by analyzing distance between data points, SVM fits the best boundary (hyperplane) that separates points into different classes, maxing the space between classes for classification or lowering regression error, and Decision Trees splits data by variables in order of descending importance and then predicts by the average of the remaining nodes. We believed these three models were distinct enough from one another that they would show meaningful differences in their results. Our models dealt with missing values in different ways: the decision trees used surrogate splits, and the KNN and SVM both used mean values of the variable to fill in any missing values.

To better capture the factors at play in predicting life expectancy, we began splitting data into subsets. We split our full data set into two smaller subsets based on whether the observations were from developed or developing countries, as one of our questions of interest was related to whether there were significant differences in

the most important variables to predict life expectancy on either side of this divide. This meant in total we would create nine models, as we had three types of models and would create one for the full data set as well as the two subsets. We created 80% training, 20% testing samples for each of our 9 models.

We fed the models with all the predictor variables to the Recursive Feature Elimination algorithm, which iteratively removes the least important features of a model by measuring the RMSE to improve its performance. This resulted in finding the predictor variables that had the greatest predictive power for life expectancy. From our full models, we also calculated performance evaluation metrics, such as MSE and R-squared, to interpret which models performed the best, which would answer one of our questions.

4 Results

The results for all nine models, which include the coefficient of determination values and the most important variables computed by the Recursive Feature Elimination algorithm, are displayed in the table below.

Table 2

Model	Data	MSE	R^2
K-NN	Full Data	0.1009	0.9047
SVM	Full Data	0.0744	0.9312
DT	Full Data	0.1896	0.8194
K-NN	Developed	0.0424	0.7528
SVM	Developed	0.0575	0.649
DT	Developed	0.0549	0.673
K-NN	Developing	0.0861	0.9068
SVM	Developing	0.0685	0.9261
DT	Developing	0.1743	0.8057

We can see that for the models that utilized the full data set, Support Vector Machine model performed the best at explaining the variance of life expectancy with an R^2 of 0.93, followed up by with K-Nearest Neighbors model with an R^2 of 0.90, and Decision Tree model performed the worst on the full data set with an R^2 of 0.82.

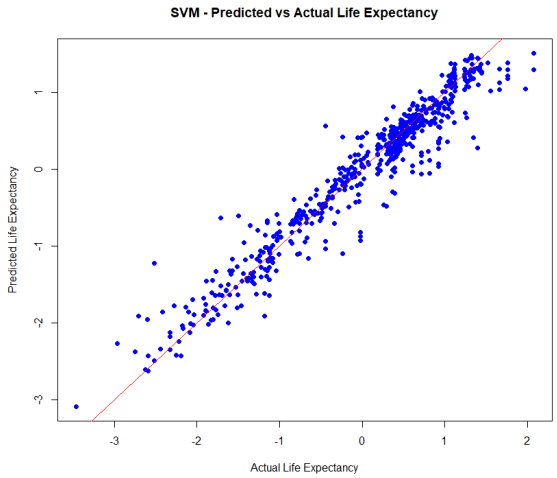

Figure 2. SVM's predictions vs ground truth on the full data. Plots for the other model/data pairings can be found in the Supplemental Materials section.

We see a similar trend with the developing countries subset where the Support Vector machine model explains the variance for life expectancy best with an R^2 of 0.92, followed by K-Nearest Neighbors with an R^2 of 0.90, and Decision Tree model explaining the variance the worst with an R^2 of 0.81. Notice how the R^2 values across the full data versus the developing countries data are very similar for each of the three models. This is not surprising as we discovered that the large majority of the observations in the full data set are from developing countries.

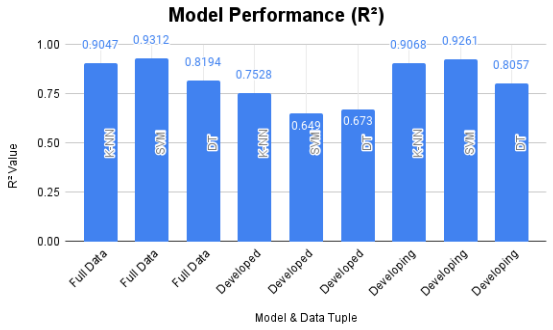

Figure 3. Barplot of R^2 for each model/data pairing.

As for the developed countries data set, our results differ from the previous two sets. The K-Nearest Neighbors model explains the variation the best with an R^2 of 0.75, followed by Decision Tree model with an R^2 of 0.67, and Support Vector Machine model actually explaining the variance worst with an R^2 of 0.65. This is in stark contrast to the other two sets where Support Vector Machines explained the variance in life expectancy best as it had the highest R^2 values. This may be attributed to the developed countries data set likely exhibiting lower variability in life expectancy because developed countries typically have similar levels of healthcare, economic stability, and quality of life. Hence, the relationship between predictors and life expectancy may be more linear or have weaker dependencies, making SVM (a non-linear model) prone to underperforming compared to simpler models like the K-Nearest-Neighbors model. To a lesser extent, the Support Vector Machine model may have also underperformed on the developed countries data set because the model typically performs better with larger data sets where it can leverage its ability to separate data using hyperplanes. Since the developed countries data set is substantially smaller (at 512 observations) compared to the full data set (at 2938 observations) and the developing countries data set (at 2426 observations), Support Vector Machine's ability to separate the data using hyperplanes may have been more inhibited in the developed countries data set.

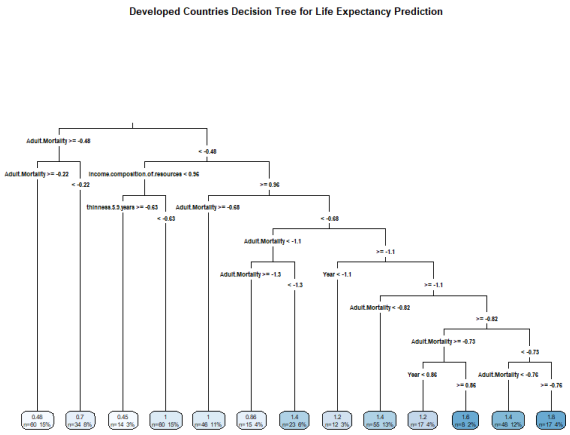

Figure 4: The Decision Tree model for developed countries is visibly unbalanced compared to the trees for the other two sets.

However, as mentioned previously with the R^2 of the models utilizing the developed countries data set, the Support Vector Machine model was not the only model that underperformed. The Decision Tree model was unbalanced, which may suggest that overfitting has occurred. The tree may perform well on training data but poorly on unseen data. Overall, all three models explained the variance of life expectancy worse than their full data set and developing data set counterparts, which was not surprising to see, as we expect the variance of life expectancy to be smaller within the developed countries data set and some predictor variables, such as HIV/AIDs deaths, is less important for explaining the variance in the developed countries data set, compared to the other two sets.

In regards to the Mean Square Error performance metric (which measures the average squared difference between predicted and actual values), we see quite interesting results. The ranking of the three models using MSE is consistent with the rankings using R^2, i.e. in order of best performing, it is Support Vector Machines,

K-Nearest Neighbors, and Decision Tree model for both the full data set and the developing countries data set, and the order is K-Nearest Neighbors, Decision Trees, and Support Vector Machines model for the developed countries data set. In addition, comparing between data sets, the MSE values of the developing countries data set models are slightly lower than the MSE values of the full data set models, which is to be expected given that most of the full data set's observations are from developing countries, so the models on the full data set are skewed towards predicting best for developing countries. Then, building models with the developed countries observations removed, you should expect to see an increase in performance relative to the models utilizing the full data set. However, the MSE values of the models utilizing the developed countries data set are lower than their counterparts from the other two sets across the board. We were expecting the MSE to be greater to indicate poor performance, similar to the results we found with the coefficient of determination. We are considering this finding to be the consequence of the models predicting the average area of life expectancy accurately, but failing to capture the wider variability, which provides reasoning for the low R^2 values with the three models utilizing the developed countries data set. This leads us to reason that, for developed countries in particular, there may exist predictor variables that have greater predictive power for life expectancy than what is available to us from the full data set.

As for the most important variables computed by the Recursive Feature Elimination (REF) algorithm, these are listed in Table 3, which you may find in the Supplemental Materials section. We found that for all nine models, Income and Adult Mortality were very important predictor variables. This was expected as Adult Mortality (the probability of dying between 15 and 60 years per 1000 population) is directly correlated with life expectancy. In the full data set and the developing data set in particular, RFE found that the number of years of schooling and deaths per 1000 to HIV/AIDS were also strong predictors for life expectancy. For developed countries in particular, RFE found that the prevalence of thinness for both ages 5-9 and 10-19 were strong predictors for life expectancy. As a supplemental note, Support Vector Machines model found more variables (avg. 5.33) to be important predictors for life expectancy than K-Nearest Neighbors (avg. 3) and Decision Trees (avg. 3.67). Hence, Support Vector Machine models found other predictor variables important, such as BMI, GDP and Total Expenditure, which the other two models did not find important.

5 Discussion

Despite Support Vector Machines underperforming on the developed countries data set, which we attributed to lower variability in life expectancy in the set and substantially fewer observations, Support Vector Machines was the best performing machine learning model to predict life expectancy. Given that Support Vector Machines is a non-linear model, we reason that it likely handles large-dimensional data and captures underlying relationships better than K-NN (which is reliant on Euclidean distance) and is not as affected by irrelevant features compared to the Decision Trees model. We were not expecting the three Decision Trees models to perform substantially worse than the other two models, given that the paper covered in the Literature Review found that the Random Forests model best predicts life expectancy, with Decision Trees following very closely behind. We infer that the randomness added by randomly selecting a subset of features to

evaluate and training the trees on a bootstrap sample of the data set allows for diverse trees that collectively improve generalization compared to Decision Trees.

Thus, Random Forests are more accurate in predicting Life Expectancy, especially considering the paper omitting five more predictor variables than our models did. Finding that the three models underperformed across the board on the developed data set tells us that the full data set is skewed towards developing countries, and thus, the models derived from the full data set are more accurate for predicting life expectancy for developing countries than developed countries. Hence, the three models for the developing countries data set performed very closely to their full data set model counterparts. This is a consequence of the class imbalance found in the full data set where there were substantially more observations from developing countries than from developed countries.

Regardless of developed or developing status, Recursive Feature Elimination found Adult Mortality and Income to be the most important variables in predicting life expectancy. This is expected, as Income is measured by the income composition of resources portion of the Human Development Index, which is a function of Gross National Income (GNI) per capita, adjusted for inequality. This measure is combined with indicators for education (mean years of schooling and expected years of schooling) and life expectancy to form the composite HDI. As history signifies, high income values indicate greater access to income resources that contribute to human development, such as greater access to healthcare, higher education, and better living standards, which are all characteristics associated with developed countries, who on average have higher life expectancy. On the contrary, lower income values reflect limited income resources, often associated with poorer access to essential services and lower standards of living, which are characteristic of developing countries, who on average have lower life expectancy. The paper we covered corroborated this, as their findings also include Income as being the most important variable in predicting life expectancy. However, while our results indicate Adult Mortality being the most important predictor of life expectancy for all nine models, the paper found Adult Mortality to be an insignificant predictor variable. While we believe its intuitive that Adult Mortality is the most important predictor variable given its inverse relationship with life expectancy, we attribute these differences to the paper omitting the following predictor variables: Year, Status, Hepatitis B, Population, Thinness (10-19 years), Thinness (5-9 years) whereas we only omitted Population.

Given that RFE found both Thinness predictor variables to be important, in addition to Adult Mortality and Income, to predicting life expectancy in developed countries, we advise public policy makers in developed countries to consider nutritional food programs for children in school and raising awareness to parents and children alike on the importance of a healthy diet to prevent the prevalence of thinness amongst children ages 5 to 19. As for developing countries, RFE found deaths by HIV/AIDS to be the most important predictor variable aside from Adult Mortality and Income, so we advise public policy makers in developing countries to raise awareness about HIV transmission, prevention methods, and the importance of regular testing to prevent deaths by HIV/AIDS.

A limitation of our research is the non-trivial amount of missing data. This may hurt the generalizability of our findings. If we can find supplemental data from the World Health Organization or the United

Nations, we can increase the robustness of our models. As covered previously, another limitation of our research was the class imbalance where there were significantly more observations from developing countries vs. developed, which makes our global findings more applicable to developing countries than developed countries.

Two improvements we can consider for our work is to experiment with hyperparameter tuning, i.e. the k in K-Nearest Neighbors or the kernel in Support Vector Machines and to fit reduced models using the important variables found by Recursive Feature Elimination. Both of these improvements can produce models that are more accurate in predicting life expectancy. Some possibilities for future work include considering continental groupings to find differences in feature rankings and considering transforming Life Expectancy into a categorical variable, so we can employ classification algorithms. This future work could also entail splitting data based on time since we have data that spans multiple years and running our models on just a single year to analyze whether the good predictive performance is due to countries not changing much year to year.

6 Conclusion

To recall our questions of interest, we aimed to find the combination of features that best predict a country's life expectancy and whether this differed from developed to developing countries, and we wanted to find a machine learning algorithm that would predict life expectancy with high success.

We were able to answer our two questions of interest by creating, refining, and implementing three prediction models on a large data set and two smaller subsets. Through the use of RFE and assessment of the MSE and R^2 of K-Nearest Neighbors, Support Vector Machines, and Decision Trees, we found that Adult Mortality and Income are the most important variables for predicting life expectancy across the board. A key difference was found when trying to find the most important combination of variables for developed vs. developing countries, as thinness rates in children were additional significant variables in developed countries, and HIV/AIDS deaths of children under four years old were a useful predictive variable in developing countries.

We also found that among our three models, Support Vector Machines generally performed the best, ignoring model complexity and computational run-time. This is likely due to the fact that it handles large-dimensional data better than KNN and not as affected by irrelevant features like DT.

These findings could prove to be useful to any parties interested in improving the life expectancy of a given nation, a key descriptive variable of that country's health and economic success. The implementation and evaluation of these predictive models could give a roadmap on how to best work towards progress in this metric.

7 Supplemental Materials

Kumar, R. (2017, January). Life Expectancy (WHO), Version 1. Retrieved November 4, 2024 from https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who.

Vydehi, K. (2020). Machine Learning Techniques for Life Expectancy Prediction. In International Journal of Advanced Trends in Computer Science and Engineering (Vol. 9, Issue 4, pp. 4503–4507). The World Academy of Research in Science and Engineering. https://doi.org/10.30534/ijatcse/2020/45942020

You may find the plots for the report linked **HERE**.

You may find the code for the report linked **HERE**.

Table 1	
Variable	**Description**
Adult Mortality	Probability of death between 15 and 60 years of age
BMI	Average Body Mass Index of Population
Thinness, 10-19	Prevalence of thinness amongst ages 10-19 (%)
Thinness, 5-9	Prevalence of thinness amongst ages 5-9 (%)
Income	Human Development Index (Income Composition of Resources)
Schooling	Average number of years of school
HIV/AIDS	HIV/AIDS deaths of 0-4 year olds per 1000 live births
Infant Deaths	Infant deaths per 1000 population
Alcohol	Alcohol consumption per capita (liters)
Percentage Expenditure	Health expenditure as a % of GDP
Hepatitis B	Hep B immunization coverage amongst 1 year olds
Measles	Measles cases per 1000 population
Polio	Polio immunization rate amongst 1 year olds
Total Expenditure	Gov’t expenditure on health as a % of total gov’t expenditure

Diphtheria	DTP3 immunization coverage amongst 1 year olds
GDP	GDP per capita in USD
Status	Developed vs. Developing

Table 3

Model	Data	Most Important Vars	Vars Count
K-NN	Full Data	Income, Schooling, Adult Mortality	3
SVM	Full Data	Income, Adult Mortality, Schooling, BMI, GDP, HIV/AIDS	6
DT	Full Data	Income, Adult Mortality, Schooling, HIV/AIDS	4
K-NN	Developed	Income, Adult Mortality	2
SVM	Developed	Adult Mortality, Income, Thinness (5-9), Thinness (10-19), Total Expenditure	5
DT	Developed	Adult Mortality, Thinness (5-9), Income	3
K-NN	Developing	Adult Mortality, Income, Thinness (5-9), Thinness (10-19)	4
SVM	Developing	Income, Adult Mortality, Schooling, BMI, HIV/AIDS	5
DT	Developing	Adult Mortality, Income, Schooling, HIV/AIDS	4

The code for this report is attached below.

```
################################################################################
# EXPLORATORY DATA ANALYSIS
################################################################################

# Install and load packages

list.of.packages <- c("MASS", "dplyr", "tidyverse", "ggplot2", "leaps", "boot",
                      "knitr", "hrbrthemes", "RColorBrewer", "paletteer",
                      "glmnet", "corrplot", "Hmisc", "GGally", "car", "kableExtra")
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()
[,"Package"])]
if(length(new.packages)) install.packages(new.packages)
lapply(list.of.packages, library, character.only = TRUE)

# Read data

data <- read.csv("Life Expectancy Data.csv")
filter(data, if_any(everything(), is.na))

dataNumeric <- data%>%
  select(-"Year", -"Country", -"Status")
data
glimpse(data)

# Yearly data points

data%>%
  ggplot(aes(Year))+
  geom_bar(fill="#1A9988")+
  theme(panel.background = element_rect(fill="#E9EDEE",color="#E9EDEE",
size=0.5,linetype="solid"))+
  labs(title="Yearly Data Points")

# Overall Life Expectancy by Year

data%>%
  mutate(Year=as.factor(Year))%>%
  ggplot(aes(Year, Life.expectancy))+
  geom_boxplot(fill="#1A9988")+
  theme(panel.background = element_rect(fill="#E9EDEE",color="#E9EDEE",
size=0.5,linetype="solid"))+
  labs(title="Life Expectancy Boxplot by Year")

# Find Life Expectancy Outliers

data <- na.omit(data)
dataNumeric <- data%>%
  select(-"Year", -"Country", -"Status")

# Note how many
out.dat <- data[FALSE,]

for (c in unique(data$Country)){
  country.dat <- data$Life.expectancy[data$Country==c]
  lowerq = quantile(data$Life.expectancy[data$Country==c])[2]
  upperq = quantile(data$Life.expectancy[data$Country==c])[4]
  iqr <- IQR(data$Life.expectancy[data$Country==c])
  upper.out = (iqr * 1.5) + upperq
  lower.out = lowerq - (iqr * 1.5)
  for (v in country.dat){
    if (v <= lower.out | v >= upper.out){
      out.dat <- rbind(out.dat, filter(data, Country==c, Life.expectancy==v))
    }
  }
```

```
}
out.dat

# Correlation Matrix

dcor <- cor(dataNumeric)

col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))

corrplot(dcor, method = "shade", shade.col = NA, tl.col = "black", tl.srt = 45,
         col = col(200), addCoef.col = "black", cl.pos = "n", order = "AOE",
         cl.cex = 2)

# Variance Inflation Factor

names <- unique(colnames(select(dataNumeric, -"Life.expectancy")))

suppressWarnings({datmat <- dataNumeric%>%
  select_("-Life.expectancy")
})

datmat <- as.matrix.data.frame(datmat)
outmat <- matrix(data=18, nrow=18,ncol=18)

for (i in 1:18){
  for (j in 1:18){
    # Check for same column
    if (i == j){
      next
    }

    # Make Linear models
    model <- lm(Life.expectancy ~ datmat[,i] + datmat[,j], data=dataNumeric)
    vifvals <- vif(model)
    outmat[i,j] = round(vifvals[1], 3)
    outmat[j,i] = round(vifvals[1], 3)
    rm(model)
    rm(vifvals)
  }
}

rownames(outmat) <- names
colnames(outmat) <- names

kable(outmat,caption = "VIF Values")

# Summary Statistics
sumstat <- data.frame("Country"=c(),
                      "Min."=c(),
                      "X1st.Qu."=c(),
                      "Median"=c(),
                      "Mean"=c(),
                      "X3rd.Qu."=c(),
                      "Max."=c())
for (c in unique(data$Country)){
  country.dat <- data$Life.expectancy[data$Country==c]
  r <- cbind(data.frame("Country"=c(c)), data.frame(as.list(summary(country.dat))))
  sumstat <- rbind(sumstat, r)
}
kable(sumstat)

# Graphical
data%>%
  ggplot(aes(x=Country, y=Life.expectancy))+
  geom_boxplot(fill="#1A9988")+
```

```
  scale_x_discrete(guide = guide_axis(angle = 70))+
  theme(axis.text.x=element_text(size=7),
        panel.background = element_rect(fill="#E9EDEE",color="#E9EDEE",
                                        size=0.5,linetype="solid"))+
  labs(title="Life Expectancy Boxplot by Country")

# QQ Plots for vars of interest
for (i in 1:length(colnames(dataNumeric))){
  qqnorm(dataNumeric[,i], main = paste0("Normal Q-Q Plot ", colnames(dataNumeric)[i]))
}

# Pairplot of scatterplots

dataNumeric%>%
  select(Life.expectancy, Adult.Mortality, HIV.AIDS, Income.composition.of.resources,
Schooling)%>%
  ggpairs(., progress = FALSE)+
  scale_fill_manual(values=c("#1A9988"))+
  scale_color_manual(values=c("#1A9988"))+
  theme(strip.background = element_rect(fill = "#EB5600"),
panel.background=element_rect(fill="#E9EDEE",color="#E9EDEE", size=0.5,linetype="solid"),
        panel.grid.minor = element_blank())+
  labs(title="Scatterplot Matrix of Interest Variables")

# Combine into pairs
for (i in 2:length(colnames(dataNumeric))){
  plot(dataNumeric[,i], dataNumeric[,1],
       main = paste0(colnames(dataNumeric)[i], "Scatterplot"),
       xlab = colnames(dataNumeric)[i],
       ylab = colnames(dataNumeric)[1])
}


# Trying to figure out cause of Adult Mortality Grouping
# Can just note this separation
data%>%
  ggplot(aes(Adult.Mortality, Life.expectancy, color=Status))+
  geom_point()

# Residual Plots

# Include Vars of interest, note anything else interesting
for (i in 2:length(colnames(dataNumeric))){
  model <- lm(dataNumeric[,1] ~ dataNumeric[,i])
  res <- resid(model)
  plot(fitted(model), res,
       main = paste0(colnames(dataNumeric)[i], " Residual Plot"),
       xlab = "Residuals",
       ylab = "Fitted Values")
  abline(0,0)
}


# Life Expectancy grouped by Status

# Reset Data
data <- read.csv("Life Expectancy Data.csv")

# Include all these to show impact of developed and developing
data%>%
  ggplot(aes(Status, Life.expectancy))+
  geom_boxplot()

data%>%
  mutate(Year=as.factor(Year))%>%
```

```
  ggplot(aes(Year, Life.expectancy))+
  geom_boxplot()+
  facet_wrap(vars(Status))+
  scale_x_discrete(guide = guide_axis(angle = 45))


# Adult Mortality grouped by Status

data%>%
  ggplot(aes(Status, Adult.Mortality))+
  geom_boxplot()

data%>%
  mutate(Year=as.factor(Year))%>%
  ggplot(aes(Year, Adult.Mortality))+
  geom_boxplot()+
  facet_wrap(vars(Status))+
  scale_x_discrete(guide = guide_axis(angle = 45))


# HIV/AIDS grouped by Status

data%>%
  ggplot(aes(Status, HIV.AIDS))+
  geom_boxplot()

data%>%
  mutate(Year=as.factor(Year))%>%
  ggplot(aes(Year, HIV.AIDS))+
  geom_boxplot()+
  facet_wrap(vars(Status))+
  scale_x_discrete(guide = guide_axis(angle = 45))


# Schooling grouped by Status

data%>%
  ggplot(aes(Status, Schooling))+
  geom_boxplot()

data%>%
  mutate(Year=as.factor(Year))%>%
  ggplot(aes(Year, Schooling))+
  geom_boxplot()+
  facet_wrap(vars(Status))+
  scale_x_discrete(guide = guide_axis(angle = 45))


# Income grouped by Status

data%>%
  ggplot(aes(Status, Income.composition.of.resources))+
  geom_boxplot()

data%>%
  mutate(Year=as.factor(Year))%>%
  ggplot(aes(Year, Income.composition.of.resources))+
  geom_boxplot()+
  facet_wrap(vars(Status))+
  scale_x_discrete(guide = guide_axis(angle = 45))



##############################################################################
# MODELING & FEATURE ELIMINATION
```

```
################################################################################

library(class)
library(FNN)
library(caret)
library(randomForest)
library(e1071)
library(rpart)
library(rpart.plot)

################################################################################

df = read.csv("Life Expectancy Data.csv")

# Calculate the percentage of NA entries per column

missing_percentage <- sapply(df, function(col) sum(is.na(col)) / nrow(df) * 100)
print(missing_percentage)

# Remove variables with missing data over 20% i.e. population

df$Population <- NULL

# Remove non-metric variables

df$Country <- NULL

# Replace NA entries with the column mean

df[] <- lapply(df, function(col) {
  if (is.numeric(col)) {
    replace(col, is.na(col), mean(col, na.rm = TRUE))
  } else {
    col
  }
})

# Count the number of NA entries per column, should now be 0

na_count <- sapply(df, function(col) sum(is.na(col)))

# Verify the result

print(na_count)

# Scale numeric columns
df_scaled <- df
numeric_cols <- sapply(df, is.numeric)
df_scaled[numeric_cols] <- scale(df[numeric_cols])  # Scale numeric columns

# Convert 'Status' to a factor and one-hot encode using model.matrix
status_one_hot <- model.matrix(~ Status - 1, data = df_scaled)  # '-1' removes the
intercept column

# Combine the one-hot encoded columns with the rest of the dataset
df_scaled <- cbind(df_scaled[, -2], status_one_hot)

# Subset of data for developed countries
df_scaled_dev <- df_scaled[df_scaled$StatusDeveloped == 1, ]
df_scaled_dev <- df_scaled_dev[, -c(20, 21)]
nrow(df_scaled_dev)

# Subset of data for developing countries
df_scaled_undev <- df_scaled[df_scaled$StatusDeveloping == 1, ]
df_scaled_undev <- df_scaled_undev[, -c(20, 21)]
```

```
nrow(df_scaled_undev)

# Set seed for reproducibility
set.seed(160)

# Split ratio is 80% training, 20% testing
trainIndex <- sample(1:nrow(df_scaled), size = 0.8 * nrow(df_scaled))
trainIndexDev <- sample(1:nrow(df_scaled_dev), size = 0.8 * nrow(df_scaled_dev))
trainIndexUndev <- sample(1:nrow(df_scaled_undev), size = 0.8 * nrow(df_scaled_undev))

# Split the data
trainData <- df_scaled[trainIndex, ]
nrow(trainData)
testData <- df_scaled[-trainIndex, ]
nrow(testData)

trainDataDev <- df_scaled_dev[trainIndexDev, ]
nrow(trainDataDev)
testDataDev <- df_scaled_dev[-trainIndexDev, ]
nrow(testDataDev)

trainDataUndev <- df_scaled_undev[trainIndexUndev, ]
nrow(trainDataUndev)
testDataUndev <- df_scaled_undev[-trainIndexUndev, ]
nrow(testDataUndev)

# Extract predictors and target variable
train_x <- trainData[, -2]  # All columns except Life.expectancy
train_y <- trainData$Life.expectancy
test_x <- testData[, -2]
test_y <- testData$Life.expectancy

train_x_dev <- trainDataDev[, -2]
train_y_dev <- trainDataDev$Life.expectancy
test_x_dev <- testDataDev[, -2]
test_y_dev <- testDataDev$Life.expectancy

train_x_undev <- trainDataUndev[, -2]
train_y_undev <- trainDataUndev$Life.expectancy
test_x_undev <- testDataUndev[, -2]
test_y_undev <- testDataUndev$Life.expectancy

##############################################################################

# Train KNN model
predictions <- knn.reg(train = train_x, test = test_x, y = train_y, k = 5)$pred

# Evaluate the model
mae <- mean(abs(predictions - test_y))  # Mean Absolute Error
mse <- mean((predictions - test_y)^2)   # Mean Squared Error
rmse <- sqrt(mse)                       # Root Mean Squared Error
rsq <- cor(predictions, test_y)^2       # R-squared

# Print results
cat("MAE: ", round(mae, 4), "\n")
cat("MSE: ", round(mse, 4), "\n")
cat("RMSE: ", round(rmse, 4), "\n")
cat("R-squared: ", round(rsq, 4), "\n")

# Plot predicted vs actual values
plot(test_y, predictions, main = "KNN - Predicted vs Actual Life Expectancy",
     xlab = "Actual Life Expectancy", ylab = "Predicted Life Expectancy",
     col = "blue", pch = 16)
abline(a = 0, b = 1, col = "red")  # Add 45-degree reference line
```

```
# Train KNN model for developed countries
predictions <- knn.reg(train = train_x_dev, test = test_x_dev, y = train_y_dev, k =
5)$pred

# Evaluate the model
mae <- mean(abs(predictions - test_y_dev))
mse <- mean((predictions - test_y_dev)^2)
rmse <- sqrt(mse)
rsq <- cor(predictions, test_y_dev)^2

# Print results
cat("MAE: ", round(mae, 4), "\n")
cat("MSE: ", round(mse, 4), "\n")
cat("RMSE: ", round(rmse, 4), "\n")
cat("R-squared: ", round(rsq, 4), "\n")

# Plot predicted vs actual values
plot(test_y_dev, predictions, main = "KNN - Predicted vs Actual Life Expectancy for
Developed Countries",
     xlab = "Actual Life Expectancy", ylab = "Predicted Life Expectancy",
     col = "blue", pch = 16)
abline(a = 0, b = 1, col = "red")  # Add 45-degree reference line



# Train KNN model for developing countries
predictions <- knn.reg(train = train_x_undev, test = test_x_undev, y = train_y_undev, k =
5)$pred

# Evaluate the model
mae <- mean(abs(predictions - test_y_undev))
mse <- mean((predictions - test_y_undev)^2)
rmse <- sqrt(mse)
rsq <- cor(predictions, test_y_undev)^2

# Print results
cat("MAE: ", round(mae, 4), "\n")
cat("MSE: ", round(mse, 4), "\n")
cat("RMSE: ", round(rmse, 4), "\n")
cat("R-squared: ", round(rsq, 4), "\n")

# Plot predicted vs actual values
plot(test_y_undev, predictions, main = "KNN - Predicted vs Actual Life Expectancy for
Developing Countries",
     xlab = "Actual Life Expectancy", ylab = "Predicted Life Expectancy",
     col = "blue", pch = 16)
abline(a = 0, b = 1, col = "red")  # Add 45-degree reference line



# FEATURE ELIM FULL KNN

# Train a KNN model using the caret package
ctrl <- rfeControl(functions = caretFuncs, method = "cv", number = 10)

# Perform RFE
knn_rfe_results <- rfe(x = train_x, y = train_y, sizes = c(1:ncol(train_x)), rfeControl =
ctrl, method = "knn")

# View the results
print(knn_rfe_results)

# Plot the results
```

```
plot(knn_rfe_results, type = c("g", "o"))

# FEATURE ELIM DEVELOPED KNN

knn_rfe_results_dev <- rfe(x = train_x_dev, y = train_y_dev, sizes =
c(1:ncol(train_x_dev)), rfeControl = ctrl, method = "knn")
print(knn_rfe_results_dev)
plot(knn_rfe_results_dev, type = c("g", "o"))

# FEATURE ELIM DEVELOPING KNN

knn_rfe_results_undev <- rfe(x = train_x_undev, y = train_y_undev, sizes =
c(1:ncol(train_x_undev)), rfeControl = ctrl, method = "knn")
print(knn_rfe_results_undev)
plot(knn_rfe_results_undev, type = c("g", "o"))

##############################################################################

# Train SVM model
svm_model <- svm(Life.expectancy ~ ., data = trainData, kernel = "radial")

# Make predictions on the test set
svm_predictions <- predict(svm_model, testData)

# Evaluate the model
mse <- mean((svm_predictions - test_y)^2)
rmse <- sqrt(mse)
mae <- mean(abs(svm_predictions - test_y))
rsq <- cor(svm_predictions, test_y)^2

# Print results
cat("MSE: ", round(mse, 4), "\n")
cat("RMSE: ", round(rmse, 4), "\n")
cat("MAE: ", round(mae, 4), "\n")
cat("R-squared: ", round(rsq, 4), "\n")

# Plot predicted vs actual values
plot(test_y, svm_predictions, main = "SVM - Predicted vs Actual Life Expectancy",
     xlab = "Actual Life Expectancy", ylab = "Predicted Life Expectancy",
     col = "blue", pch = 16)
abline(a = 0, b = 1, col = "red")



# Train SVM model for developed countries
svm_model <- svm(Life.expectancy ~ ., data = trainDataDev, kernel = "radial")

# Make predictions on the test set
svm_predictions <- predict(svm_model, testDataDev)

# Evaluate the model
mse <- mean((svm_predictions - test_y_dev)^2)
rmse <- sqrt(mse)
mae <- mean(abs(svm_predictions - test_y_dev))
rsq <- cor(svm_predictions, test_y_dev)^2

# Print results
cat("MSE: ", round(mse, 4), "\n")
cat("RMSE: ", round(rmse, 4), "\n")
cat("MAE: ", round(mae, 4), "\n")
cat("R-squared: ", round(rsq, 4), "\n")

# Plot predicted vs actual values
plot(test_y_dev, svm_predictions, main = "SVM - Predicted vs Actual Life Expectancy for
Developed Countries",
```

```
     xlab = "Actual Life Expectancy", ylab = "Predicted Life Expectancy",
     col = "blue", pch = 16)
abline(a = 0, b = 1, col = "red")



# Train SVM model for developing countries
svm_model <- svm(Life.expectancy ~ ., data = trainDataUndev, kernel = "radial")

# Make predictions on the test set
svm_predictions <- predict(svm_model, testDataUndev)

# Evaluate the model
mse <- mean((svm_predictions - test_y_undev)^2)
rmse <- sqrt(mse)
mae <- mean(abs(svm_predictions - test_y_undev))
rsq <- cor(svm_predictions, test_y_undev)^2

# Print results
cat("MSE: ", round(mse, 4), "\n")
cat("RMSE: ", round(rmse, 4), "\n")
cat("MAE: ", round(mae, 4), "\n")
cat("R-squared: ", round(rsq, 4), "\n")

# Plot predicted vs actual values
plot(test_y_undev, svm_predictions, main = "SVM - Predicted vs Actual Life Expectancy for
Developing Countries",
     xlab = "Actual Life Expectancy", ylab = "Predicted Life Expectancy",
     col = "blue", pch = 16)
abline(a = 0, b = 1, col = "red")



# FEATURE ELIM FULL SVM

# Train a SVM model using the caret package
ctrl <- rfeControl(functions = caretFuncs, method = "cv", number = 5)

svm_rfe_results <- rfe(x = train_x, y = train_y, sizes = c(1:ncol(train_x)), rfeControl =
ctrl, method = "svmRadial")
print(svm_rfe_results)
plot(svm_rfe_results, type = c("g", "o"))

# FEATURE ELIM DEVELOPED SVM

svm_rfe_results_dev <- rfe(x = train_x_dev, y = train_y_dev, sizes =
c(1:ncol(train_x_dev)), rfeControl = ctrl, method = "svmRadial")
print(svm_rfe_results_dev)
plot(svm_rfe_results_dev, type = c("g", "o"))

# FEATURE ELIM DEVELOPING SVM

svm_rfe_results_undev <- rfe(x = train_x_undev, y = train_y_undev, sizes =
c(1:ncol(train_x_undev)), rfeControl = ctrl, method = "svmRadial")
print(svm_rfe_results_undev)
plot(svm_rfe_results_undev, type = c("g", "o"))

##############################################################################

# Train a decision tree model
tree_model <- rpart(Life.expectancy ~ ., data = trainData, method = "anova")

# Make predictions on the test set
dt_predictions <- predict(tree_model, testData)
```

```
# Calculate performance metrics
mae <- mean(abs(dt_predictions - test_y))
mse <- mean((dt_predictions - test_y)^2)
rmse <- sqrt(mse)
rsq <- cor(dt_predictions, test_y)^2

# Print the results
cat("MAE: ", round(mae, 4), "\n")
cat("MSE: ", round(mse, 4), "\n")
cat("RMSE: ", round(rmse, 4), "\n")
cat("R-squared: ", round(rsq, 4), "\n")

# Plot the decision tree
rpart.plot(tree_model,
           type = 3,
           extra = 101,
           main = "Decision Tree for Life Expectancy Prediction")



# Train a decision tree model for developed countries
tree_model <- rpart(Life.expectancy ~ ., data = trainDataDev, method = "anova")

# Make predictions on the test set
dt_predictions <- predict(tree_model, testDataDev)

# Calculate performance metrics
mae <- mean(abs(dt_predictions - test_y_dev))
mse <- mean((dt_predictions - test_y_dev)^2)
rmse <- sqrt(mse)
rsq <- cor(dt_predictions, test_y_dev)^2

# Print the results
cat("MAE: ", round(mae, 4), "\n")
cat("MSE: ", round(mse, 4), "\n")
cat("RMSE: ", round(rmse, 4), "\n")
cat("R-squared: ", round(rsq, 4), "\n")

# Plot the decision tree
rpart.plot(tree_model,
           type = 3,
           extra = 101,
           main = "Developed Countries Decision Tree for Life Expectancy Prediction")



# Train a decision tree model for developing countries
tree_model <- rpart(Life.expectancy ~ ., data = trainDataUndev, method = "anova")

# Make predictions on the test set
dt_predictions <- predict(tree_model, testDataUndev)

# Calculate performance metrics
mae <- mean(abs(dt_predictions - test_y_undev))
mse <- mean((dt_predictions - test_y_undev)^2)
rmse <- sqrt(mse)
rsq <- cor(dt_predictions, test_y_undev)^2

# Print the results
cat("MAE: ", round(mae, 4), "\n")
cat("MSE: ", round(mse, 4), "\n")
cat("RMSE: ", round(rmse, 4), "\n")
cat("R-squared: ", round(rsq, 4), "\n")

# Plot the decision tree
```

```
rpart.plot(tree_model,
           type = 3,
           extra = 101,
           main = "Developing Countries Decision Tree for Life Expectancy Prediction")



# FEATURE ELIM FULL DT

# Train a DT model using the caret package
ctrl <- rfeControl(functions = caretFuncs, method = "cv", number = 10)

dt_rfe_results <- rfe(x = train_x, y = train_y, sizes = c(1:ncol(train_x)), rfeControl =
ctrl, method = "rpart")
print(dt_rfe_results)
plot(dt_rfe_results, type = c("g", "o"))

# FEATURE ELIM DEVELOPED DT

dt_rfe_results_dev <- rfe(x = train_x_dev, y = train_y_dev, sizes =
c(1:ncol(train_x_dev)), rfeControl = ctrl, method = "rpart")
print(dt_rfe_results_dev)
plot(dt_rfe_results_dev, type = c("g", "o"))

# FEATURE ELIM DEVELOPING DT

dt_rfe_results_undev <- rfe(x = train_x_undev, y = train_y_undev, sizes =
c(1:ncol(train_x_undev)), rfeControl = ctrl, method = "rpart")
print(dt_rfe_results_undev)
plot(dt_rfe_results_undev, type = c("g", "o"))
```