World Happiness Report | Machine Learning Project

HarvardX: PH125.9x Data Science Capstone

Ian Mathers | February 26, 2021

Introduction

The World Happiness Report ranks 156 countries based on their citizens' happiness levels. It is a publication of the Sustainable Development Solutions Network with data collected by Gallup World Poll. It is a survey that combines a number of economic and social factors into a total score. The purpose of this project is to analyze this data, visualize it and apply some basic machine learning prediction models.

Dataset

The dataset was obtained on Kaggle. Reports from 2015 and 2019 are used. For simplicity the two files are automatically downloaded during the loading process below.

Data Loading

```
# if required packages
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")

# libraries
library(tidyverse)
library(caret)

# csv file downloads from GitHub
dat15 <- read.csv("https://raw.githubusercontent.com/Airborne737/World_Happiness/master/2015.csv")
dat19 <- read.csv("https://raw.githubusercontent.com/Airborne737/World_Happiness/master/2019.csv")

dat15 <- dat15 %>%
  rename(country = Country, score = Happiness.Score, GDP_capita = Economy..GDP.per.Capita., healthy_lif
  select(country, score, GDP_capita, healthy_life_expectancy, freedom, generosity, corruption)

dat19 <- dat19 %>%
  rename(country = Country.or.region, score = Score, GDP_capita = GDP.per.capita, healthy_life_expectanc
  select(country, score, GDP_capita, healthy_life_expectancy, freedom, generosity, corruption)
```

Data Preparation, Training and Testing

The datasets are small. The 2015 set contains 158 observations, one for each country. 2019 has 156. Due to the small sample sizes the 2015 material will be divided into two and used for training/testing of several algorithms. Final validation of the best model will use the 2019 set. Only matching data of the two years have been kept with the columns renamed. They have been verified for consistency. Accuracy will be compared using RMSE. Residual mean squared error is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}(\hat{x}_i - x_i)^2}$$

Where N is the number of observations, x_i the actual observations for variable i and $\hat{x}_i$ the predicted values for variable i. The RMSE is a commonly used loss function that simply measures the differences between predicted and observed values. It can be interpreted similarly to a standard deviation.

The following columns will be used to predict the happiness scores: GDP per capita, healthy life expectancy, perception of freedom, giving and generosity and trust in government which is listed as corruption.

```
# create training and testing sets
set.seed(1, sample.kind="Rounding")
test_index <- createDataPartition(y = dat15$score, times = 1, p = 0.5, list = FALSE)
train_set <- dat15[-test_index,]
test_set <- dat15[test_index,]

# RMSE defined
RMSE <- function(true_ratings, predicted_ratings){
  sqrt(mean((true_ratings - predicted_ratings)^2))
}
```

Exploratory Data Analysis

We start by analyzing the data structure from 2015 (dat15). It shows 158 observations, each row is a country, and the 7 renamed columns. All classes are numeric aside from country which is comprised of characters.

```
str(dat15)
```

```
## 'data.frame':    158 obs. of  7 variables:
##  $ country                : chr  "Switzerland" "Iceland" "Denmark" "Norway" ...
##  $ score                  : num  7.59 7.56 7.53 7.52 7.43 ...
##  $ GDP_capita             : num  1.4 1.3 1.33 1.46 1.33 ...
##  $ healthy_life_expectancy: num  0.941 0.948 0.875 0.885 0.906 ...
##  $ freedom                : num  0.666 0.629 0.649 0.67 0.633 ...
##  $ generosity             : num  0.297 0.436 0.341 0.347 0.458 ...
##  $ corruption             : num  0.42 0.141 0.484 0.365 0.33 ...
```

dat19 is structured the same except that 156 countries were ranked that year.

```
str(dat19)
```

```
## 'data.frame':    156 obs. of  7 variables:
##  $ country                : chr  "Finland" "Denmark" "Norway" "Iceland" ...
##  $ score                  : num  7.77 7.6 7.55 7.49 7.49 ...
##  $ GDP_capita             : num  1.34 1.38 1.49 1.38 1.4 ...
##  $ healthy_life_expectancy: num  0.986 0.996 1.028 1.026 0.999 ...
##  $ freedom                : num  0.596 0.592 0.603 0.591 0.557 0.572 0.574 0.585 0.584 0.532 ...
##  $ generosity             : num  0.153 0.252 0.271 0.354 0.322 0.263 0.267 0.33 0.285 0.244 ...
##  $ corruption             : num  0.393 0.41 0.341 0.118 0.298 0.343 0.373 0.38 0.308 0.226 ...
```

The summary function provides statistical summaries for each column. The data is consistent across both years.

```r
summary(dat15)
```

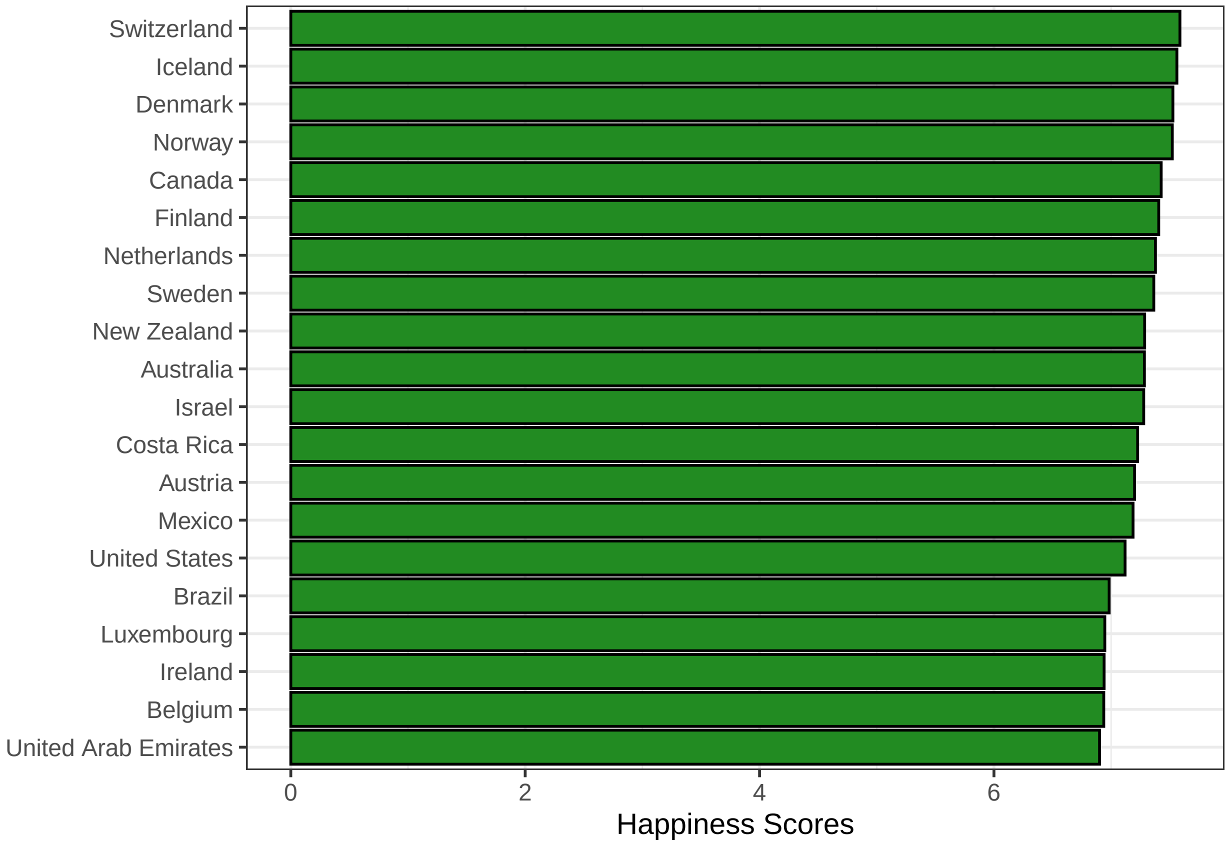
```
##    country              score          GDP_capita     healthy_life_expectancy
##  Length:158         Min.   :2.839   Min.   :0.0000   Min.   :0.0000
##  Class :character   1st Qu.:4.526   1st Qu.:0.5458   1st Qu.:0.4392
##  Mode  :character   Median :5.232   Median :0.9102   Median :0.6967
##                     Mean   :5.376   Mean   :0.8461   Mean   :0.6303
##                     3rd Qu.:6.244   3rd Qu.:1.1584   3rd Qu.:0.8110
##                     Max.   :7.587   Max.   :1.6904   Max.   :1.0252
##     freedom         generosity       corruption
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.3283   1st Qu.:0.1506   1st Qu.:0.06168
##  Median :0.4355   Median :0.2161   Median :0.10722
##  Mean   :0.4286   Mean   :0.2373   Mean   :0.14342
##  3rd Qu.:0.5491   3rd Qu.:0.3099   3rd Qu.:0.18025
##  Max.   :0.6697   Max.   :0.7959   Max.   :0.55191
```

```r
summary(dat19)
```

```
##    country              score          GDP_capita     healthy_life_expectancy
##  Length:156         Min.   :2.853   Min.   :0.0000   Min.   :0.0000
##  Class :character   1st Qu.:4.545   1st Qu.:0.6028   1st Qu.:0.5477
##  Mode  :character   Median :5.380   Median :0.9600   Median :0.7890
##                     Mean   :5.407   Mean   :0.9051   Mean   :0.7252
##                     3rd Qu.:6.184   3rd Qu.:1.2325   3rd Qu.:0.8818
##                     Max.   :7.769   Max.   :1.6840   Max.   :1.1410
##     freedom         generosity       corruption
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.3080   1st Qu.:0.1087   1st Qu.:0.0470
##  Median :0.4170   Median :0.1775   Median :0.0855
##  Mean   :0.3926   Mean   :0.1848   Mean   :0.1106
##  3rd Qu.:0.5072   3rd Qu.:0.2482   3rd Qu.:0.1412
##  Max.   :0.6310   Max.   :0.5660   Max.   :0.4530
```

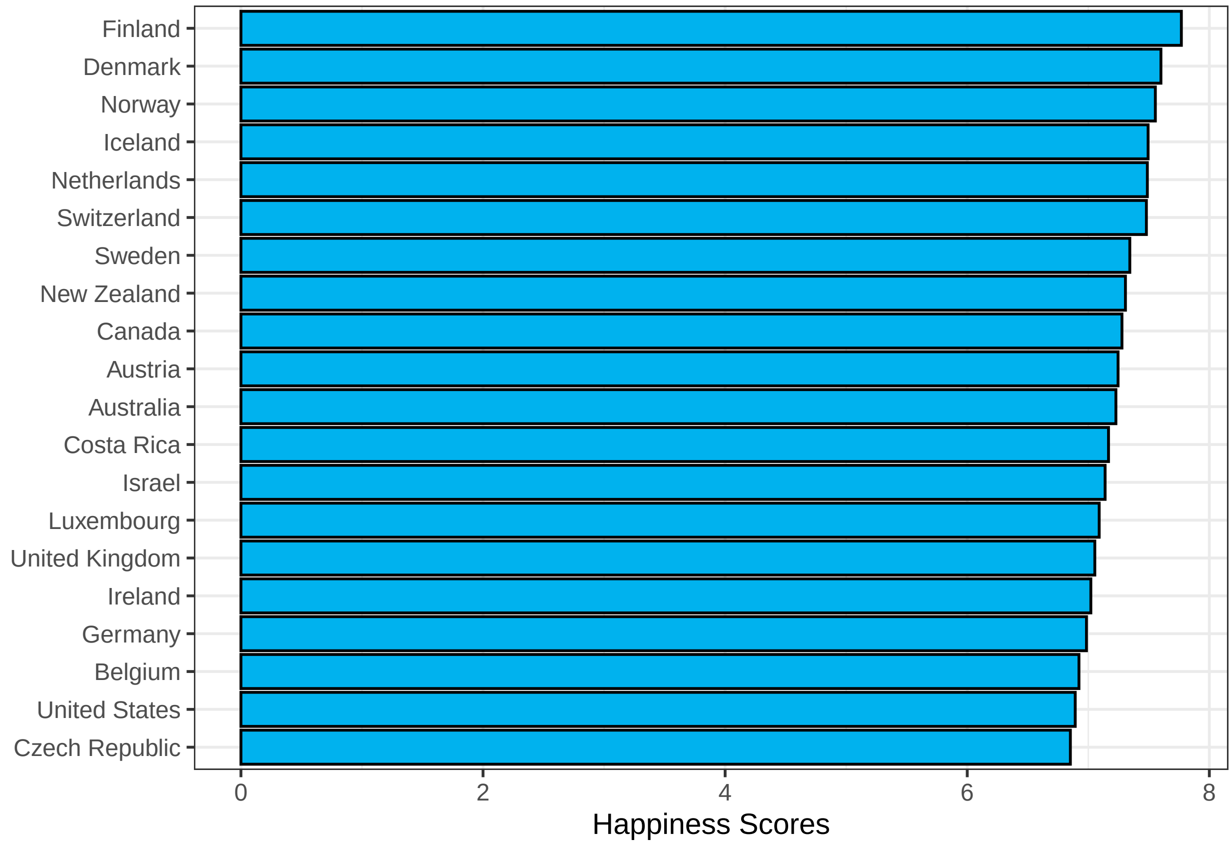
A look at the top 20 countries with the highest happiness scores in 2015 shows many coming from Europe.

```r
dat15 %>%
  arrange(-score) %>%
  top_n(20, score) %>%
  ggplot(aes(score, reorder(country, score))) +
  geom_bar(color = "black", fill = "forestgreen", stat = "identity") +
  xlab("Happiness Scores") +
  ylab(NULL) +
  theme_bw()
```



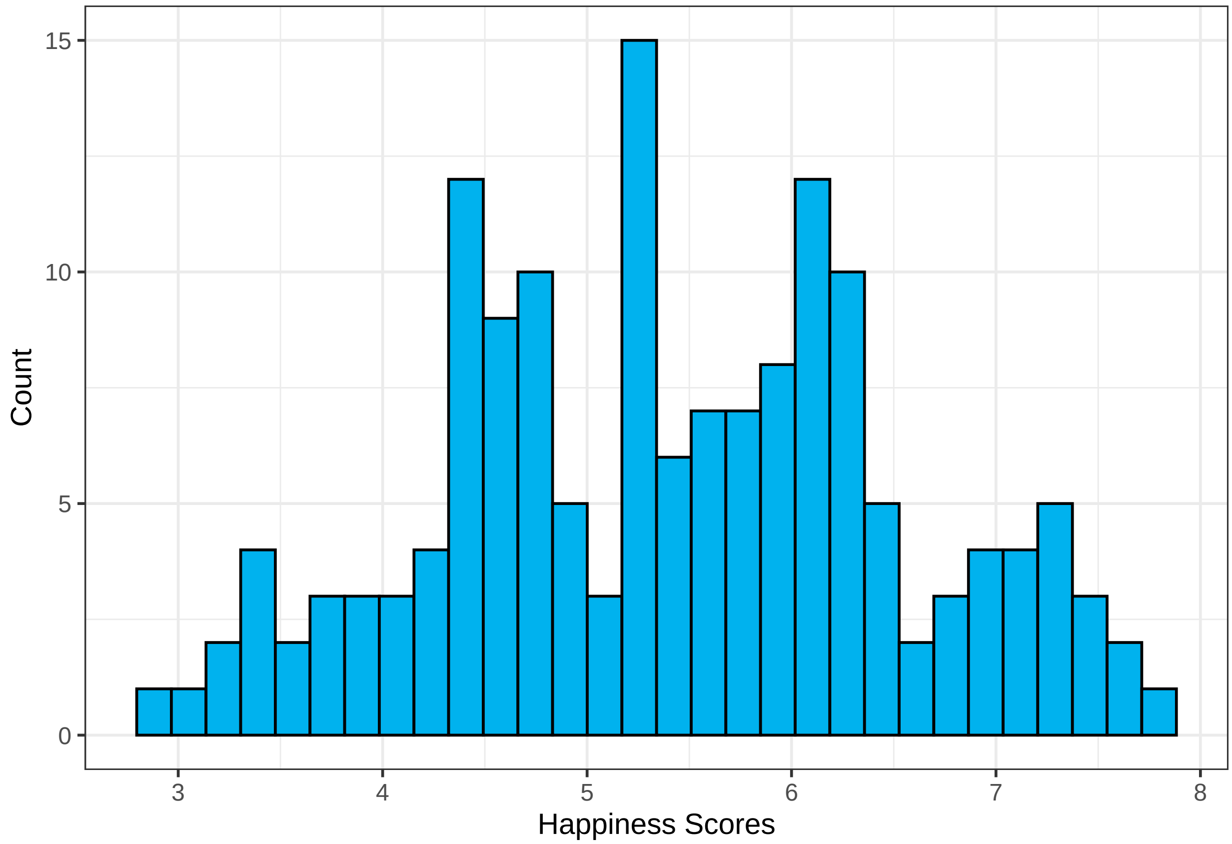
2019 saw Finland taking the lead over Switzerland. Scandinavian countries are consistently rated high. The United Kingdom, Germany and the Czech Republic made the top 20 that year.

```
dat19 %>%
  arrange(-score) %>%
  top_n(20, score) %>%
  ggplot(aes(score, reorder(country, score))) +
  geom_bar(color = "black", fill = "deepskyblue2", stat = "identity") +
  xlab("Happiness Scores") +
  ylab(NULL) +
  theme_bw()
```



The distribution of the happiness scores shows three peaks or modes. Making it a multimodal distribution.

```
dat19 %>%
  ggplot(aes(score)) +
  geom_histogram(color = "black", fill = "deepskyblue2", bins = 30) +
  labs(x = "Happiness Scores", y = "Count") +
  scale_x_continuous() +
  theme_bw()
```



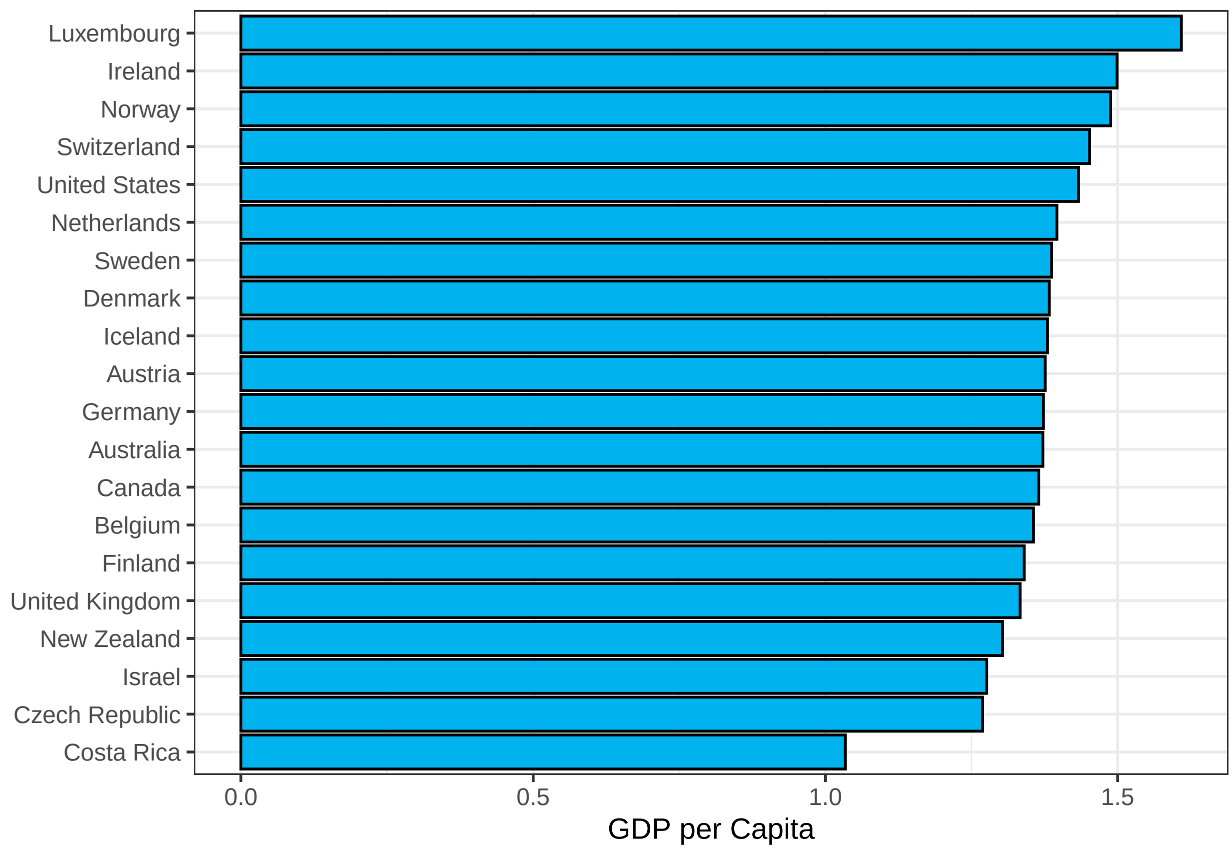
A correlation matrix reveals how the data points are correlated to each other. It is not surprising that healthy life expectancy is the most correlated with the score. A more useful measure is GDP per capita. It makes sense to expect economic growth to have a high impact on happiness levels. Interestingly generosity and corruption have the lowest numbers. Correlation is not causation however and more analysis is needed to make conclusions.

```
dat19 %>%
  select(-country) %>%
  cor()
```

```
##                              score  GDP_capita healthy_life_expectancy
## score                   1.00000000  0.79388287              0.77988315
## GDP_capita              0.79388287  1.00000000              0.83546212
## healthy_life_expectancy 0.77988315  0.83546212              1.00000000
## freedom                 0.56674183  0.37907907              0.39039478
## generosity              0.07582369 -0.07966231             -0.02951086
## corruption              0.38561307  0.29891985              0.29528281
##                           freedom  generosity corruption
## score                   0.5667418  0.07582369  0.3856131
## GDP_capita              0.3790791 -0.07966231  0.2989198
## healthy_life_expectancy 0.3903948 -0.02951086  0.2952828
## freedom                 1.0000000  0.26974181  0.4388433
## generosity              0.2697418  1.00000000  0.3265375
## corruption              0.4388433  0.32653754  1.0000000
```

Here is a list of the countries with the highest GDP per capita. Not surprisingly we also find the happiest ones which reflects the high correlation. The order is a bit different.

```
dat19 %>%
  arrange(-GDP_capita) %>%
  top_n(20, score) %>%
  ggplot(aes(GDP_capita, reorder(country, GDP_capita))) +
  geom_bar(color = "black", fill = "deepskyblue2", stat = "identity") +
  xlab("GDP per Capita") +
  ylab(NULL) +
  theme_bw()
```



Model 1

The first model that will be used to predict happiness scores based on all the data points is a simple linear regression model. It will provide a baseline to work from.

```
lm_train <- train_set %>% select(-country) %>% train(score ~ ., method = "lm", data = .)
lm_predict <- predict(lm_train, test_set)
lm_result <- RMSE(test_set$score, lm_predict)
results <- tibble(Method = "Model 1: Linear Regression", RMSE = lm_result)
results %>% knitr::kable()
```

Method	RMSE
Model 1: Linear Regression	0.5866397

Our first RMSE result comes in at 0.5866. Let's see if we can improve on it with different methods.

Model 2

The second model will use Random Forest. It uses randomness to build an uncorrelated forest of trees which are used to predict an outcome.

```
set.seed(1, sample.kind="Rounding")
fitcontrol <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
rf_train <- train_set %>% select(-country) %>% train(score ~ ., method = "rf", trControl = fitcontrol, 
rf_predict <- predict(rf_train, test_set)
rf_result <- RMSE(test_set$score, rf_predict)
results <- bind_rows(results, tibble(Method = "Model 2: Random Forest", RMSE = rf_result))
results %>% knitr::kable()
```

Method	RMSE
Model 1: Linear Regression	0.5866397
Model 2: Random Forest	0.5808796

Random Forest provides a slight gain on the linear model.

Model 3

The third model uses the Ranger implementation of Random Forests.

```
set.seed(1, sample.kind="Rounding")
ranger_train <- train_set %>% select(-country) %>% train(score ~ ., method = "ranger", trControl = trai
ranger_predict <- predict(ranger_train, test_set)
ranger_result <- RMSE(test_set$score, ranger_predict)
results <- bind_rows(results, tibble(Method = "Model 3: Ranger RF", RMSE = ranger_result))
results %>% knitr::kable()
```

Method	RMSE
Model 1: Linear Regression	0.5866397
Model 2: Random Forest	0.5808796
Model 3: Ranger RF	0.5730181

Ranger provides a decent gain on the previous model breaking below 0.58.

Model 4

We turn to a non-parametric algorithm, K-Nearest Neighbors. Let's see how it performs versus the others.

```
set.seed(1, sample.kind="Rounding")
knn_train <- train_set %>% select(-country) %>% train(score ~ ., method = "knn", trControl = trainContro
knn_predict <- predict(knn_train, test_set)
knn_result <- RMSE(test_set$score, knn_predict)
results <- bind_rows(results, tibble(Method = "Model 4: K-Nearest Neighbors", RMSE = knn_result))
results %>% knitr::kable()
```

Method	RMSE
Model 1: Linear Regression	0.5866397
Model 2: Random Forest	0.5808796
Model 3: Ranger RF	0.5730181
Model 4: K-Nearest Neighbors	0.5496701

KNN has lowered the RMSE significantly. We will use it as our final model.

Final Validation

Having found our model with the lowest RMSE using KNN the final step is to train it on dat15 and test its accuracy using dat19.

```
set.seed(1, sample.kind="Rounding")
knn_train15 <- dat15 %>% select(-country) %>% train(score ~ ., method = "knn", trControl = trainControl
knn_predict19 <- predict(knn_train15, dat19)
final_result <- RMSE(dat19$score, knn_predict19)
results <- bind_rows(results, tibble(Method = "Final validation: K-Nearest Neighbors", RMSE = final_resu
results %>% knitr::kable()
```

Method	RMSE
Model 1: Linear Regression	0.5866397
Model 2: Random Forest	0.5808796
Model 3: Ranger RF	0.5730181
Model 4: K-Nearest Neighbors	0.5496701
Final validation: K-Nearest Neighbors	0.5803704

Conclusion

The goal of this project was to collect, process and analyze data on the World Happiness Reports from 2015 and 2019. We then used several basic algorithms on small sample sizes to make predictions on the happiness scores. We started with a baseline model and progressively improved the RMSE results with minimal tuning. The final validation shows a RMSE of 0.5804. The accuracy is limited. The continued evolution of machine learning allows for limitless approaches in tackling such exercises with more complexity and accuracy. With our goals achieved this concludes the project.