Worldwide Happiness 2017: Ordinal Technical Analysis Using R

Author: Sean E. Curl

(U) INTRODUCTION:

(U) The world happiness rating is a measurement of a survey of the state of global happiness. The first report was published in 2012, and the second, third, and fourth were published in 2013, 2015, and 2016, respectively. The World Happiness 2017 ranks 155 countries by their happiness levels. The goal of this analysis is to develop a model which best predicts the 'Happiness Rank' based on several variables, which are the summation of a country's happiness score. The analysis will be determined by how well Happiness Rank is affected by variables such as "GDP", "Family", "Healthy Life Expectancy" etc. The value derived from determining a model that is best at predicting Happiness Rank is that the model could be used to predict next year's 2018 Happiness Rankings.

(U) AUTHOR NOTE: the outputted analysis is an abbreviated submission of the overall analysis completed. The majority of R code is redacted. The analysis was conducted to attempt to predict the accuracy of ranking (ordinal regression) using KNN/LDA. The analysis did not attempt transform variables to account for individual factors such as the population-weighted distribution for explaining each countries happiness ranking. The purpose was to identify the most significant variables for determining the response variable happiness rank and exploring which of these (KNN/LDA) models would have the highest classification rate for predicting rank.

(U) Fitting the Model:

```
# FULL MODEL USING RFIT: NON-PARAMETRIC ASSUMPTIONS
#### Reference 2A.

## Call:
## rfit.default(formula = Happiness.Rank ~ Economy.GDP.per.Capita +
##     Family + Health.Life.Expectancy + Freedom + Generosity +
##     Trust.Government.Corruption)
##
## Coefficients:
##                          Estimate Std. Error t.value   p.value
## (Intercept)              222.7373     7.4878 29.7466 < 2.2e-16 ***
## Economy.GDP.per.Capita   -37.8036     8.0828 -4.6771 6.504e-06 ***
## Family                   -42.7545     7.9858 -5.3538 3.219e-07 ***
## Health.Life.Expectancy   -46.0353    12.7073 -3.6227 0.0003999 ***
## Freedom                  -61.8755    13.5366 -4.5710 1.017e-05 ***
## Generosity               -14.5004    13.0156 -1.1141 0.2670515
## Trust.Government.Corruption -18.6675    19.1417 -0.9752 0.3310388
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Overall Wald Test: 670.3061 p-value: 0
```

(U) Y = B0 + B1X1 + B2X2 + B3X3 + B4X4 + B5X5 + B6X6

(U) Estimated model (reference 2A): Y = 222.7373 + -37.8036* Economy.GDP.per.Capita + -42.7545 * Family + -46.0353 * Health.Life.Expectancy + -61.8755 * Freedom + -14.5004 * Generosity + -18.6675 * Trust.Government.Corruption

(U) Coefficients for this model are interpreted as meaning: for everything increase in ~one (~1) for [ex.] Economy.GDP.per.Capita, holding all else constant, rank can be expected to decrease by -37.8036.

(U) Two predictors from the full model are non-significant, that is, they have p-values > 0.05. These predictors are Generosity with a p-value of 0.2670515 and Trust.Government.Corruption with a p-value of 0.3310388. These two predictors will be removed from the full model and then the model will be compared against a reduced model. However, before the predictors are removed, for information purposes only given that the model is non-parametric, diagnostics were conducted on the full model.

(U) Reduced Model and Variable Selection (F* Statistic Test):

```
#### Reference 3A.
# rfit ANOVA TEST, good these predictors were not significant to the model and could be dropped

## Call:
## rfit.default(formula = Happiness.Rank ~ Economy.GDP.per.Capita +
##     Family + Health.Life.Expectancy + Freedom)

## Coefficients:
##                    Estimate Std. Error t.value   p.value
## (Intercept)          221.9317     7.1376 31.0935 < 2.2e-16 ***
## Economy.GDP.per.Capita -37.9802     7.5629 -5.0219 1.434e-06 ***
## Family               -42.5743     7.7716 -5.4782 1.772e-07 ***
## Health.Life.Expectancy -46.2988    12.3256 -3.7563 0.0002462 ***
## Freedom              -72.1820    11.6161 -6.2139 4.854e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Overall Wald Test: 693.4842 p-value: 0
```

(U) Y = B0 + B1X1 + B2X2 + B3X3 + B4X4

(U) Estimated reduced model (reference 3A): Y = 221.9317 + -37.9802 * Economy.GDP.per.Capita + -42.5743 * Family + -46.2988 * Health.Life.Expectancy + -72.1820 * Freedom

(U) Coefficients for this model are interpreted as meaning: for everything increase in ~one (~1) for Economy.GDP.per.Capita, holding all else constant, rank can be expected to decrease by -37.8036. . . … increase in ~one (~1) for Family, holding all else constant, rank can be expected to decrease by -42.5743 … increase in ~one (~1) for Health.Life.Expectancy, holding all else constant, rank can be expected to decrease by -46.2988 … increase in ~one (~1) for Freedom, holding all else constant, rank can be expected to decrease by -72.1820

(U) All predictors are for the reduced model are statistically significant.

```
#### Reference 3B.
# Drop in Dispersion Test: ANOVA
## F-Statistic    p-value
##      1.3594      0.2600
```

(U) The drop-in dispersion test (see reference 3B) is a variation of the ANOVA test for rfit models. This test shows us only the f-statistic and p-value of the two compared models, full and reduced. A p-value of 0.2600 indicated that I fail to reject the null hypothesis and conclude Ha (the removal of the two predictors from the model is not statically significant to the prediction of y).

(U) CROSS VALIDATION OF RFIT MODELS

```
#### Reference 3C.
##      Train.MSE2 Test.MSE2
## [1,]    368.961  520.8718
```

(U) The cross-validation of the reduced model allows us to compute the prediction accuracy of the model, specifically, how large the mean variation is between the actual values and predicted. I split the data into a training and testing dataset, and then compare the reduced model training MSE & testing MSE against the training MSE & testing MSE of the full model.

(U) FULL Train.MSE Test.MSE [1,] 349.9091 549.5

(U) REDUCED Train.MSE Test.MSE [1,] 368.961 520.8718

(U) The reduced model has a better testing MSE when compared to the full model testing MSE.

(U) AUTHOR NOTE: the MSE is essentially measuring the 'quality of fit.' The training MSE is expected to be lower in almost all cases because the model is fitted to the training data, but not to the testing data. If the training MSE > testing MSE, the testing data is usually under-fitting. A higher variance also usually indicates lower bias.

(U) DISCRIMINANT ANALYSIS

```
#### Reference 4A.
Happy1 = rep("Happy Rank", length(Happiness.Rank))
Happy1[Happiness.Rank < 39 ] = "Super Happiness"
Happy1[Happiness.Rank >= 39 & Happiness.Rank < 76] = "Very Happiness"
Happy1[Happiness.Rank >= 76 & Happiness.Rank < 115] = "Somewhat Happiness"
Happy1[Happiness.Rank >= 115] = "Low Happiness"

# Linear Discriminant Analysis (LDA)
#### Reference 4B.
## Prior probabilities of groups:
##     Low Happiness Somewhat Happiness    Super Happiness
##         0.2645161         0.2516129         0.2451613
##    Very Happiness
##         0.2387097
##
## Group means:
##                    Economy.GDP.per.Capita    Family Health.Life.Expectancy
## Low Happiness               0.5418805 0.9080391           0.2931549
## Somewhat Happiness          0.8604513 1.1234034           0.5139554
## Super Happiness             1.4050478 1.4300662           0.7668678
## Very Happiness              1.1747244 1.3214676           0.6554928
##                    Freedom Generosity Trust.Government.Corruption
## Low Happiness      0.3095349  0.2359103              0.09881556
## Somewhat Happiness 0.3811003  0.2467355              0.08153675
## Super Happiness    0.5390112  0.3127489              0.21076866
## Very Happiness     0.4142042  0.1915533              0.10386632
##
## Coefficients of linear discriminants:
##                              LD1       LD2       LD3
## Economy.GDP.per.Capita     -1.7431054  0.7312724  1.295530
## Family                    -1.6679485 -0.9314977  1.099637
## Health.Life.Expectancy     -2.4583739 -2.3671218 -3.218011
## Freedom                   -3.1443366 -0.2319218 -2.702284
## Generosity                -0.7984351  3.6691952 -5.207168
## Trust.Government.Corruption -0.5027098  8.1882755  6.402691
##
## Proportion of trace:
##    LD1    LD2    LD3
## 0.9199 0.0672 0.0129
```

(U) The second model uses Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA). The lda.fit output provides the Prior probabilities, group means, and coefficients of linear discriminants. The prior probabilities depict the prior probability for each of the four groups: "Super Happiness" (0.2451613), "Very Happiness" (0.2387097), "Somewhat Happiness" (0.2516129), and "Low Happiness" (0.2645161). The group means provide the probability for each group for each predicator. The coefficients of linear discriminants computes score for each group by the coefficients and assigns them to the group with highest score.

(U) Each plot and histogram in 4B gives a visualization of the four probability groups for each coefficient of linear discriminants. In addition, the coefficients are helpful in deciding which variable is affecting classification. Comparing the values between the four groups, the highest coefficient value is the variable with the 'largest weight' for determining the classification. So for example, if we take a look at reference 4B I can see that Trust.Government.Corruption has the highest coefficient values in LD1, LD2 and LD3. This means Trust.Government.Corruption contributed more to that particular group's classification.

(U) Cross-validation (LOOCV):

```
#### Reference 4C.
# FULL MODEL
## Happy1            Low Happiness Somewhat Happiness Super Happiness
##   Low Happiness              31               7              1
##   Somewhat Happiness          7              20              0
##   Super Happiness             0               1             23
##   Very Happiness              0               8              8
##
## Happy1            Very Happiness
##   Low Happiness               2
##   Somewhat Happiness         12
##   Super Happiness            14
##   Very Happiness             21

## [1] 0.6129032

# REDUCED MODEL
## Happy1            Low Happiness Somewhat Happiness Super Happiness
##   Low Happiness              30               9              0
##   Somewhat Happiness          7              18              1
##   Super Happiness             0               1             29
##   Very Happiness              0               9              9
##
## Happy1            Very Happiness
##   Low Happiness               2
##   Somewhat Happiness         13
##   Super Happiness             8
##   Very Happiness             19

## [1] 0.6193548
```

(U) The Leave-one-out cross validation (LOOCV) uses a single observation from the original sample as the validation data, and the remaining observations as the training data. This is repeated such that each observation in the sample is used once as the validation data. Ultimately, the LOOCV provides use with a classification rate which can be calculated (as seen in reference 4C) using mean(Happy1 == lda.fit.full$class). In the output reference 4C, notice that the full model has a slightly worse classification rate when compared to the reduced: 0.6129032 < 0.6193548. Therefore, the reduced model is slightly better at classifying predicted values into one of the four actual groups.

(U) Both LDA and QDA assume that the measurements from each class are normally distributed. LDA assumes covariance is equal.

(U) KNN

```
#### Reference 6A.
set.seed(13)
happiness2 <- subset(happiness, select = -c(1, 3:5, 12))
Happy2 = rep("Happy Rank", length(happiness2$Happiness.Rank))
Happy2[happiness2$Happiness.Rank < 39 ] = "Super Happiness"
Happy2[happiness2$Happiness.Rank >= 39 & happiness2$Happiness.Rank < 76] = "Very Happiness"
Happy2[happiness2$Happiness.Rankk >= 76 & happiness2$Happiness.Rank < 115] = "Somewhat Happiness"
Happy2[happiness2$Happiness.Rank >= 115] = "Low Happiness"
```

(U) MODEL USING KNN = 1 AND KNN = 3

```
#### Reference 6B.

##                knn.result1
## Y.testing       Happy Rank Low Happiness Super Happiness Very Happiness
##   Happy Rank            22             0               0              0
##   Low Happiness          0            20               0              0
```

```
##   Super Happiness        0          0          19          1
##   Very Happiness         1          0          0          15

#### Reference 6C.

##                knn.result2
## Y.testing        Happy Rank Low Happiness Super Happiness Very Happiness
##   Happy Rank           22          0          0          0
##   Low Happiness         1         19          0          0
##   Super Happiness       0          0         19          1
##   Very Happiness        0          0          0         16

#### Reference 6D.
mean( Y.testing == knn.result1 )

## [1] 0.974359
mean( Y.testing == knn.result2 )

## [1] 0.974359
```

(U) DETERMINING BEST KNN AND CLASS SELECTION

```
#### Reference 6E.
which.max(class.rate)

## [1] 5

# The optimal neighborhood size for KNN is  k = 5 neighbors. This yields a correct classification rate of 100%.

## [1] 0.9871795
```

(U) The third model uses K-Nearest-Neighbors (KNN).

(U) AUTHOR NOTE: KNN is non-parametric and therefore a great fit for our non-parametric rfit ranking dataset.

(U) I removed columns: 1 (Country), 3:5 (Happiness.Score, Whisker.High, Whisker.low), and 12 (dystopian residual) from the dataset in order to be able calculate KNN accurately and so that the classification rate could compared accurately against LDA and QDA classification rates. The dataset was then split 50/50 into a training and testing dataset. The model was first run using KNN = 1 and then KNN = 3.

(U) Classification Rate: KNN = 1 = 0.974359 = 97.4% KNN = 3 = 0.974359 = 97.4% Instead of just randomly guessing which KNN is the best and will provide the highest or best classification rate, I used: which.max(class.rate) for KNN level and max(class.rate) for max classification rate. Thus, I determine the best KNN is 5 with a classification rate of 0.9871795 or 98.7%.

(U) Interestingly, the high classification rate for KNN makes sense intuitively. I expected countries which are both close to each other geographically, as well as culturally, to have a rank approximately similar to a near-by or adjacent country. This assumption is easily confirmed by the high classification rate of the KNN model where the model is able to predict rank based on a country's nearest neighbor (max class. K = 5). A KNN model would be expected to preform extremely well in predicting a country's rank for the 2018 Worldwide Happiness Report.

(U) CONCLUSION:

(U) The goal of this analysis was to select a model which was the best at predicting Happiness Rank. The 'best at predicting' can also be referred to as being the model with the highest classification rate. I used an rfit model as the basis of our analysis. This model is non-parametric and therefore, does not have any assumptions. The second model I used was the LDA & QDA model. Both of these models assume each class is normally distributed; the only difference is that the QDA model does not assume equal covariance.

(U) The last and final model I used to predict Happiness Rank was the KNN model, which has no assumptions of the underlying data distribution. The results of this analysis were that the QDA reduced model had a better classification rate when compared to the reduced LDA model. However, ultimately the KNN model had the best classification rate for predicting Happiness rank with a classification rate of 0.9871795 or 98.7%. Therefore, I can conclude that using these three models, KNN was the best model for predicting the Happiness Rank for each observation.