Part 2: Model fitting, Model Evaluation, Model Deployment

Juweria Ali

2022-03-25

Clearing the workspace and setting the working directory.

```
rm(list=ls())
```

Set working directory

```
setwd("C:/Users/juwer/Desktop/MSc/Projects")
```

Loading required libraries

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(ggplot2)
library(mlbench)
library(MASS)
library(leaps)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

Loading the dataset

```
escapesClean = read.csv("escapesClean.csv", stringsAsFactors = T, header = T)
```

Explore the data

```
summary(escapesClean)
```

```
##    Season           Species         Age        Average.Weight     Number
##  Autumn:57   Other        : 55   Min.   : 2.00   Min.   :  15   Min.   :     1
##  Spring:58   Salmon       :151   1st Qu.:10.00   1st Qu.: 600   1st Qu.:   216
##  Summer:51   Salmon.Brood:  2   Median :15.00   Median :2000   Median :  3000
##  Winter:55   Salmon.Fresh: 13   Mean   :15.31   Mean   :2191   Mean   : 13536
##                                 3rd Qu.:19.00   3rd Qu.:3400   3rd Qu.: 10775
##                                 Max.   :48.00   Max.   :9250   Max.   :336470
##      Cause     Producing      SLR                Cu                 Zn
##  Human  :138   No : 21   Min.   :-0.7633   Min.   :-3.9100   Min.   :-0.4252
##  Natural: 83   Yes:200   1st Qu.: 0.6599   1st Qu.: 0.5429   1st Qu.: 6.8072
##                          Median : 2.2654   Median : 1.7553   Median : 9.5058
##                          Mean   : 3.1627   Mean   : 2.1218   Mean   :10.1518
##                          3rd Qu.: 3.1242   3rd Qu.: 3.6470   3rd Qu.:13.8066
##                          Max.   :35.2477   Max.   : 8.4502   Max.   :24.7346
##        N                P               Org
##  Min.   :-105.5   Min.   :-13.97   Min.   :  65.13
##  1st Qu.: 240.9   1st Qu.: 80.49   1st Qu.: 356.62
##  Median : 358.3   Median :121.98   Median : 564.55
##  Mean   : 340.0   Mean   :122.59   Mean   : 553.28
##  3rd Qu.: 437.0   3rd Qu.:162.52   3rd Qu.: 726.65
##  Max.   : 696.5   Max.   :244.29   Max.   :1092.56
```

Next analyse the correlation between of variables

In this case our target is a category, so correlations won’t work without some pre-processing

Creating a copy of the dataset to calculate correlations on it

```
CorEscapes = escapesClean
```

Create a numerical equivalent as {0,1}

```
CorEscapes$Cause01 = as.numeric(escapesClean$Cause)-1
```

Calculate the correlation between variables

Removing the categorical variables for correlation to work

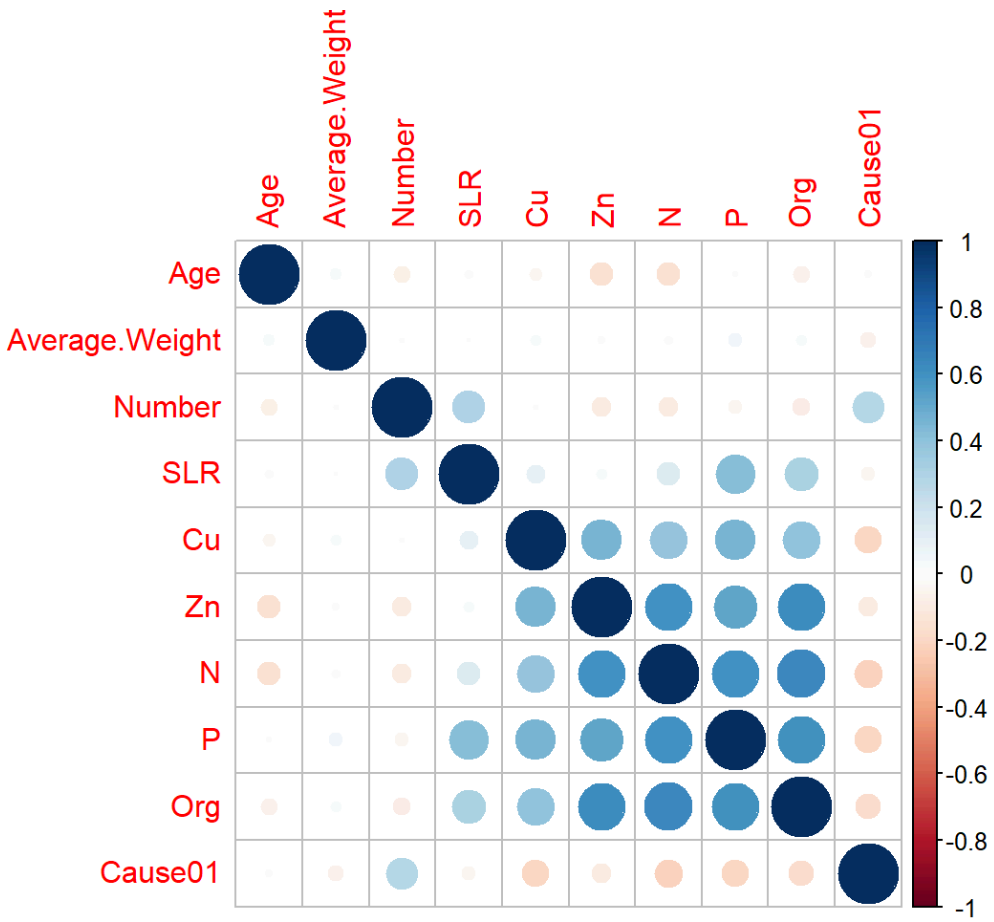
Feature selection

```
round(cor(CorEscapes[-c(1,2,6,7)], method = "spearman"), 2)
```

```
##                 Age Average.Weight Number   SLR    Cu    Zn     N     P   Org
## Age            1.00           0.03  -0.07  0.02 -0.04 -0.15 -0.14 -0.01 -0.07
## Average.Weight  0.03           1.00  -0.01  0.00  0.03  0.02 -0.02  0.05  0.03
## Number         -0.07          -0.01   1.00  0.29  0.01 -0.10 -0.10 -0.05 -0.08
## SLR             0.02           0.00   0.29  1.00  0.09  0.03  0.15  0.42  0.32
## Cu             -0.04           0.03   0.01  0.09  1.00  0.46  0.38  0.45  0.40
## Zn             -0.15           0.02  -0.10  0.03  0.46  1.00  0.60  0.52  0.62
## N              -0.14          -0.02  -0.10  0.15  0.38  0.60  1.00  0.60  0.64
## P              -0.01           0.05  -0.05  0.42  0.45  0.52  0.60  1.00  0.61
## Org            -0.07           0.03  -0.08  0.32  0.40  0.62  0.64  0.61  1.00
## Cause01         0.01          -0.06   0.27 -0.05 -0.20 -0.10 -0.22 -0.20 -0.17
##               Cause01
## Age              0.01
## Average.Weight  -0.06
## Number           0.27
## SLR             -0.05
## Cu              -0.20
## Zn              -0.10
## N               -0.22
## P               -0.20
## Org             -0.17
## Cause01          1.00
```

Visualise the correlation between variables

```
corrplot(cor(CorEscapes[-c(1,2,6,7)], method = "spearman"))
```



Validation method

```
control = trainControl(method = "cv", number = 10)
```

Pre-processing

```
prep2 = c('range')
```

MODEL 1: LOGISTIC REGRESSION

Design and implement a Logistic Regression model to predict Cause

Fitting a model

```
set.seed(123)
model1 = train(Cause ~ .,
               data = escapesClean,method = "glm", family = "binomial",
               trControl = control,

               )
model1
```

```
## Generalized Linear Model
##
## 221 samples
##  12 predictor
##   2 classes: 'Human', 'Natural'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 199, 199, 198, 199, 199, 199, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.6464427  0.1994412
```

Analyse and visualise the results

```
summary(model1)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2398  -0.8948  -0.5905   1.0690   2.0321
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           7.651e-01  9.436e-01   0.811   0.4175
## SeasonSpring          5.888e-01  4.692e-01   1.255   0.2095
## SeasonSummer          1.968e-01  4.884e-01   0.403   0.6870
## SeasonWinter          9.528e-01  4.686e-01   2.033   0.0420 *
## SpeciesSalmon        -6.188e-01  3.937e-01  -1.572   0.1160
## SpeciesSalmon.Brood -8.371e-01  1.598e+00  -0.524   0.6004
## SpeciesSalmon.Fresh -8.496e-01  7.246e-01  -1.173   0.2410
## Age                  -1.228e-03  2.318e-02  -0.053   0.9578
## Average.Weight       -4.361e-05  8.596e-05  -0.507   0.6119
## Number                1.875e-05  1.075e-05   1.744   0.0812 .
## ProducingYes          2.919e-01  5.381e-01   0.542   0.5875
## SLR                   2.511e-03  4.499e-02   0.056   0.9555
## Cu                   -1.159e-01  7.588e-02  -1.528   0.1266
## Zn                    3.560e-02  4.603e-02   0.773   0.4393
## N                    -2.656e-03  1.739e-03  -1.527   0.1268
## P                    -3.946e-03  3.966e-03  -0.995   0.3198
## Org                  -7.487e-04  9.914e-04  -0.755   0.4501
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 292.54  on 220  degrees of freedom
## Residual deviance: 253.64  on 204  degrees of freedom
## AIC: 287.64
##
## Number of Fisher Scoring iterations: 5
```

Predict using the model

```
probs1 = predict(model1, escapesClean, type = "prob")
predicted1 = predict(model1, escapesClean)
escapesClean$predicted = predicted1
escapesClean$correct = (escapesClean$Cause==escapesClean$predicted)
```

Evaluate results

```
confusionMatrix(predicted1, escapesClean$Cause)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Human Natural
##    Human     116      44
##    Natural    22      39
##
##                Accuracy : 0.7014
##                  95% CI : (0.6363, 0.7609)
##     No Information Rate : 0.6244
##     P-Value [Acc > NIR] : 0.01013
##
##                   Kappa : 0.3278
##
##  Mcnemar's Test P-Value : 0.00974
##
##             Sensitivity : 0.8406
##             Specificity : 0.4699
##          Pos Pred Value : 0.7250
##          Neg Pred Value : 0.6393
##              Prevalence : 0.6244
##          Detection Rate : 0.5249
##    Detection Prevalence : 0.7240
##       Balanced Accuracy : 0.6552
##
##        'Positive' Class : Human
##
```

MODEL 2: CLASSIFICATION MODEL OF YOUR CHOICE - Random Forest

Design and implement another classification model that has been covered in the CMM535 module to predict Cause.

Fitting the model

```
set.seed(123)
model2 = train(Cause ~ .,method = "rf",
               data = escapesClean,
               trControl = control,
               tuneGrid = expand.grid(mtry=c(1,2,3,4)),

              )
model2
```

```
## Random Forest
##
## 221 samples
##  14 predictor
##   2 classes: 'Human', 'Natural'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 199, 199, 198, 199, 199, 199, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   1     0.7011858  0.2701582
##   2     0.9001976  0.7879508
##   3     0.9638340  0.9253697
##   4     0.9820158  0.9626765
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 4.
```

Predict using the model

```
probs2 = predict(model2, escapesClean, type = "prob")
predicted2 = predict(model2, escapesClean)
escapesClean$predicted = predicted2
escapesClean$correct = (escapesClean$Cause==escapesClean$predicted)
```

Evaluate the results of the predictions

```
confusionMatrix(predicted2, escapesClean$Cause)
```

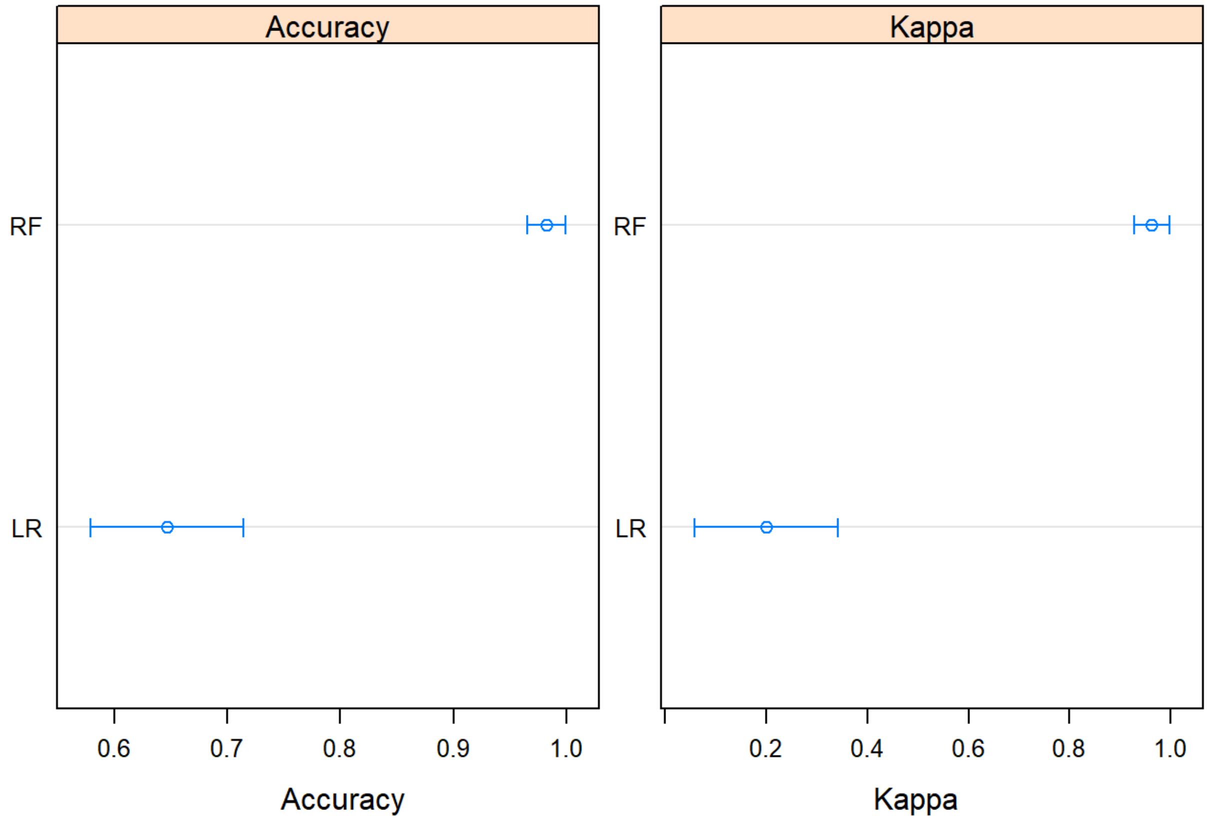
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Human Natural
##    Human     138       0
##    Natural     0      83
##
##                Accuracy : 1
##                  95% CI : (0.9834, 1)
##     No Information Rate : 0.6244
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.6244
##          Detection Rate : 0.6244
##    Detection Prevalence : 0.6244
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : Human
##
```

Comparing model1 & model2

```
results = resamples(list(LR = model1, RF = model2))
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: LR, RF
## Number of resamples: 10
##
## Accuracy
##         Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LR 0.5000000 0.5909091 0.6363636 0.6464427 0.6704545 0.8181818    0
## RF 0.9545455 0.9550395 1.0000000 0.9820158 1.0000000 1.0000000    0
##
## Kappa
##           Min.    1st Qu.    Median      Mean  3rd Qu.      Max. NA's
## LR -0.01680672 0.05912621 0.1689393 0.1994412 0.197092 0.5849057    0
## RF  0.90434783 0.90829873 1.0000000 0.9626765 1.000000 1.0000000    0
```

```
dotplot(results, conf.level = 0.95, scales = "free")
```



Critically compare and contrast the effectiveness of model 1 and model 2 [Word limit of 150 words].

- Below is the confusion matrix for model1. We can see that 22 instances of class Human and 44 instances of class Natural were classified incorrectly.

  ```
      Reference
  ```

  Prediction Human Natural Human 116 44 Natural 22 39

The accuracy for this model is 0.6923 and Human was returned as the positive class with fewer misclassifications in comparison to class Natural.

- Below is the confusion matrix for model2. We can see that all the instances were correctly classified.

  ```
        Reference
  ```

  Prediction Human Natural Human 138 0 Natural 0 83

The accuracy for this model is 1.

Accuracy has been used as a measure to evaluate the performance of the models.From the confusion matrices we can clearly say that model2 has outperformed model1 by giving 100% accuracy. Using the dot plot for comparison, we can see the intervals and the margin of error do not overlap, hence, the difference in performance is said to be statistically significant.

MODEL 3: LINEAR REGRESSION

Select data features that will be suitable and relevant for predicting Number Design and implement a Linear Regression model to predict Number

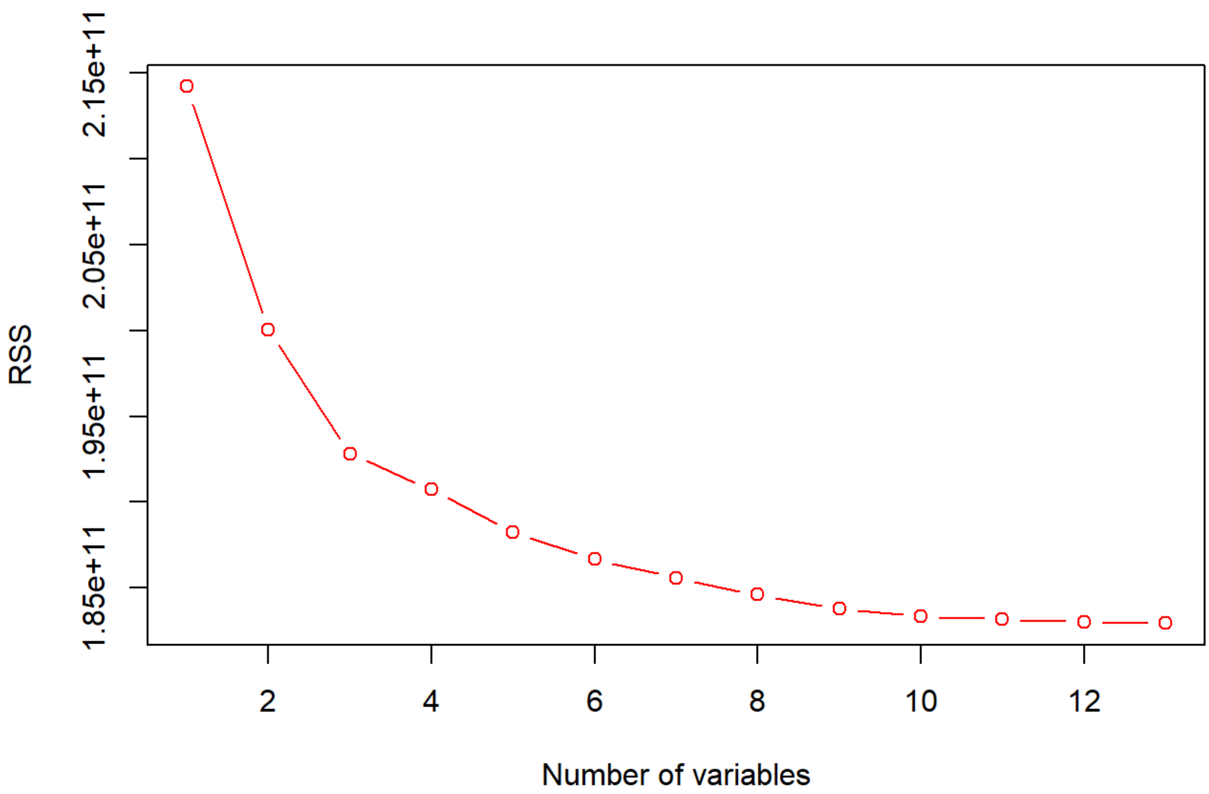
Removing unnecessary columns

```
escapesClean<-subset(escapesClean,select=-c(predicted,correct))
```

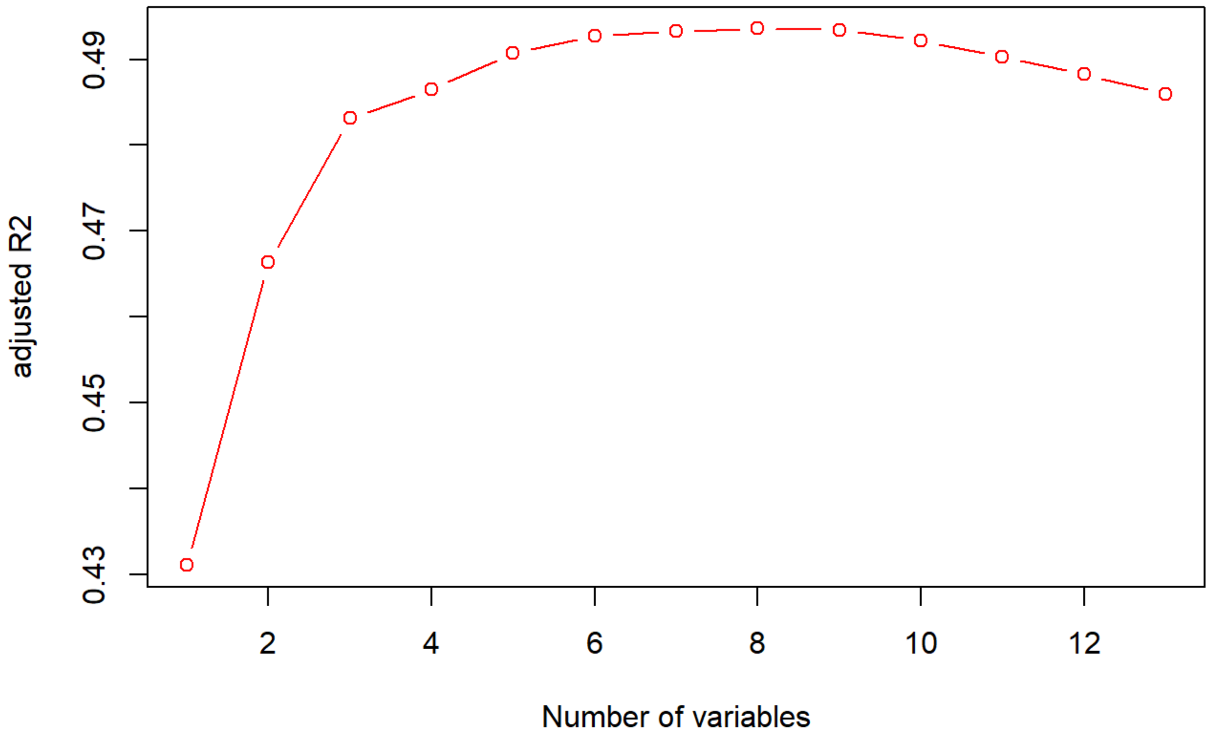
Feature selection

```
fullSearch = regsubsets(Number ~ .,data = escapesClean,
                        method = "exhaustive", nvmax = 13)
full = summary(fullSearch)

plot(full$rss, type = "b", col = "red",
     ylab = "RSS", xlab = "Number of variables")
```




```
plot(full$adjr2, type = "b", col = "red",
     ylab = "adjusted R2", xlab = "Number of variables")
```




```
full$outmat
```

```
##           SeasonSpring SeasonSummer SeasonWinter SpeciesSalmon
## 1  ( 1 )  " "          " "          " "          " "
## 2  ( 1 )  " "          " "          "*"          " "
## 3  ( 1 )  " "          " "          "*"          " "
## 4  ( 1 )  " "          " "          "*"          " "
## 5  ( 1 )  " "          " "          "*"          " "
## 6  ( 1 )  " "          " "          "*"          " "
## 7  ( 1 )  " "          " "          "*"          " "
## 8  ( 1 )  " "          " "          "*"          " "
## 9  ( 1 )  " "          " "          "*"          "*"
## 10  ( 1 ) " "          " "          "*"          "*"
## 11  ( 1 ) " "          " "          "*"          "*"
## 12  ( 1 ) " "          " "          "*"          "*"
## 13  ( 1 ) "*"          " "          "*"          "*"
##           SpeciesSalmon.Brood SpeciesSalmon.Fresh Age Average.Weight
## 1  ( 1 )  " "                 " "                 " " " "
## 2  ( 1 )  " "                 " "                 " " " "
## 3  ( 1 )  " "                 " "                 " " " "
## 4  ( 1 )  " "                 " "                 " " " "
## 5  ( 1 )  " "                 " "                 " " " "
## 6  ( 1 )  " "                 " "                 " " " "
## 7  ( 1 )  " "                 " "                 " " " "
## 8  ( 1 )  " "                 " "                 " " " "
## 9  ( 1 )  " "                 " "                 " " " "
## 10  ( 1 ) " "                 " "                 " " " "
## 11  ( 1 ) "*"                 " "                 " " " "
## 12  ( 1 ) "*"                 "*"                 " " " "
## 13  ( 1 ) "*"                 "*"                 " " " "
##           CauseNatural ProducingYes SLR Cu  Zn  N   P   Org
## 1  ( 1 )  " "          " "          "*" " " " " " " " " " "
## 2  ( 1 )  " "          " "          "*" " " " " " " " " " "
## 3  ( 1 )  "*"          " "          "*" " " " " " " " " " "
## 4  ( 1 )  "*"          " "          "*" " " " " " " "*" " "
## 5  ( 1 )  "*"          " "          "*" " " "*" " " "*" " "
## 6  ( 1 )  "*"          " "          "*" " " "*" " " "*" "*"
## 7  ( 1 )  "*"          " "          "*" "*" "*" " " "*" "*"
## 8  ( 1 )  "*"          "*"          "*" "*" "*" " " "*" "*"
## 9  ( 1 )  "*"          "*"          "*" "*" "*" " " "*" "*"
## 10  ( 1 ) "*"          "*"          "*" "*" "*" "*" "*" "*"
## 11  ( 1 ) "*"          "*"          "*" "*" "*" "*" "*" "*"
## 12  ( 1 ) "*"          "*"          "*" "*" "*" "*" "*" "*"
## 13  ( 1 ) "*"          "*"          "*" "*" "*" "*" "*" "*"
```

Three variables appear to be the best option.

```
q = full$which[3,-c(1)]
vars = paste(names(q[q == TRUE]), collapse = "+")
form = as.formula(paste("Number ~ ", vars))
form
```

```
## Number ~ SeasonWinter + CauseNatural + SLR
```

We can observe that the best three variable model is charges Season + Cause + SLR either from the matrix or the extracted formula.

Creating train and test data

To ensure that we are not overfitting, we have a train-test split.

```
set.seed(123)
selected = createDataPartition(escapesClean$Number, p = 0.7, list = F)
trainData = escapesClean[selected, ]
testData = escapesClean[-selected, ]
dim(trainData)
```

```
## [1] 157  13
```

Fitting the model

```
set.seed(123)
model3 = train(Number ~ Season + Cause + SLR, data = trainData, method = "lm",
trControl = control, preProcess = prep2)
```

```
varImp(model3)
```

```
## lm variable importance
##
##              Overall
## SLR           100.00
## SeasonWinter   28.38
## CauseNatural   17.47
## SeasonSpring   10.42
## SeasonSummer    0.00
```

Predict the model

```
pred3 = predict(model3, testData)
```

Summary of the key metrics of regression

```
postResample(pred3, testData$Number)
```

```
##         RMSE     Rsquared          MAE
## 2.828209e+04 5.819523e-01 1.672154e+04
```

MODEL 4: REGRESSION MODEL OF YOUR CHOICE - Random Forest

Design and implement a second Regression model, using the same set of data features as for Model 3, using techniques that have been covered in CMM535 module, to predict Number.

Fitting the model

```
set.seed(123)
model4 <- train(Number ~ Season + Cause + SLR,
    data = trainData,
    method = "rf",
    metric = "RMSE",
    ntree= 500,
    maxnodes = 5,
    trControl = control,
    preProcess = prep2,
    tuneGrid = expand.grid(mtry=c(1,2,3,4))
   )

print(model4)
```

```
## Random Forest
##
## 157 samples
##   3 predictor
##
## Pre-processing: re-scaling to [0, 1] (5)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 141, 141, 141, 141, 141, 141, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared   MAE
##   1     27795.10  0.2842107  14068.43
##   2     27664.46  0.2831501  13413.89
##   3     27857.89  0.2943915  13292.37
##   4     28434.05  0.2757579  13526.36
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 2.
```

```
varImp(model4)
```

```
## rf variable importance
##
##                Overall
## SLR           100.0000
## SeasonWinter   12.1559
## CauseNatural    5.3303
## SeasonSummer    0.9547
## SeasonSpring    0.0000
```

Predicting the model

```
pred4 = predict(model4, testData)
```

Summary of the key metrics of regression

```
postResample(pred4, testData$Number)
```

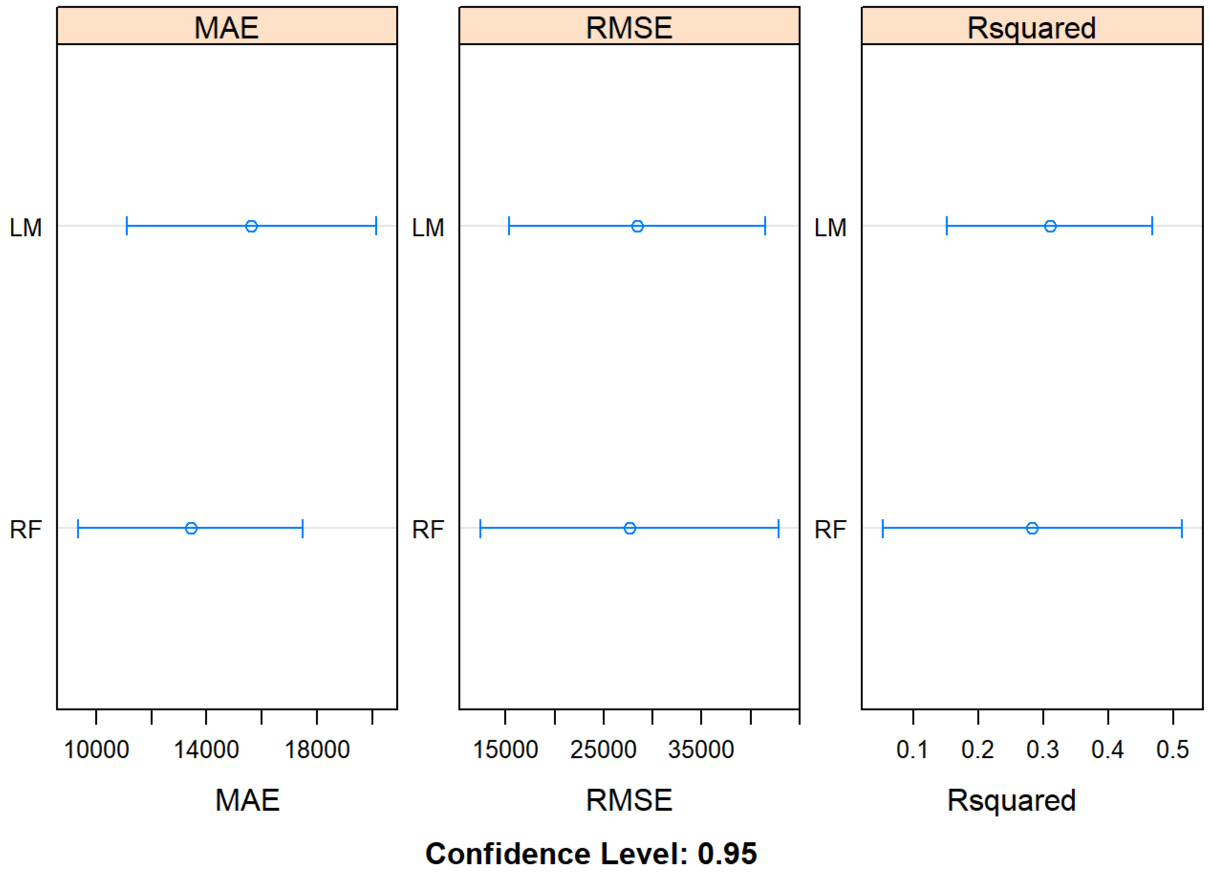
```
##         RMSE     Rsquared          MAE
## 2.759882e+04 7.580624e-01 1.272780e+04
```

Comparing model 3 & 4

```
results = resamples(list(LM = model3, RF = model4))
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: LM, RF
## Number of resamples: 10
##
## MAE
##         Min.   1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## LM 9112.822 10213.503 13227.27 15623.32 21299.87 25962.63    0
## RF 7729.020  8675.494 11365.67 13413.89 16406.15 22611.19    0
##
## RMSE
##          Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## LM 10736.132 13537.31 22056.21 28398.62 41439.50 58056.60    0
## RF  9376.964 11225.34 16511.32 27664.46 38535.54 67524.89    0
##
## Rsquared
##           Min.    1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM 0.020905476 0.22841889 0.2567143 0.3096691 0.3265861 0.7498006    0
## RF 0.000323988 0.06961685 0.1556052 0.2831501 0.3193349 0.9031341    0
```

```
dotplot(results, conf.level = 0.95, scales = "free")
```



Critically compare and contrast the effectiveness of model 3 and model 4 [Word limit of 150 words].

We can see for model3 the values are as below

```
    RMSE     Rsquared          MAE
```

2.828209e+04 5.819523e-01 1.672154e+04

And for model4 the values are as below

```
    RMSE     Rsquared          MAE
```

2.759882e+04 7.580624e-01 1.272780e+04

RMSE is being used as a metric to compare the performance of the models. Lower the RMSE value the better the model. Hence, we can say that in this case model4 performs better as it has the lowest RMSE value among the two models.From the dot plots for comparison we see that the intervals and the margin of error are quite overlapping, hence, we may say that the difference in performance is not statistically significant.

MODEL DEPLOYMENT: A BASIC SHINY APP

model2 has given a better accuracy, hence considering it to refit for the Shiny R app

Variable importance to select the 6 best predictors

```
varImp(model2)
```

```
## rf variable importance
##
##                      Overall
## correctTRUE          100.000
## predictedNatural      73.991
## Number                34.365
## Cu                    30.266
## N                     28.868
## Org                   26.591
## P                     23.432
## Age                   22.403
## SLR                   20.621
## Zn                    19.161
## Average.Weight        17.450
## SeasonWinter           5.978
## SpeciesSalmon          2.821
## ProducingYes           2.045
## SeasonSummer           1.903
## SeasonSpring           1.506
## SpeciesSalmon.Fresh    0.303
## SpeciesSalmon.Brood    0.000
```

Refitting the model by selecting the six best variables i.e.Number, N,Cu,Org,P,Age according to their importance

```
set.seed(123)
model2refit = train(Cause ~ Number + Cu + Age + N + Org + P,
                    data = escapesClean,method = "rf",
                    trControl = control,
                    preProcess = prep2
                )
model2refit
```

```
## Random Forest
##
## 221 samples
##   6 predictor
##   2 classes: 'Human', 'Natural'
##
## Pre-processing: re-scaling to [0, 1] (6)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 199, 199, 198, 199, 199, 199, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.7055336  0.3539498
##   4     0.6964427  0.3302171
##   6     0.7055336  0.3501009
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

Save the refitted model

```
saveRDS(model2refit, "C:\\Users\\juwer\\Desktop\\MSc\\Semester 2\\Data Science Development - CMM53
5\\Courseworks\\Part2\\CW2\\model.rds")
```

Ethical and Social Issues

The use of machine learning algorithms for predictions raises several ethical and/or societal concerns, including, but not limited to, incomplete evidence leading to poor judgment, a lack of transparency, erroneous evidence leading to unintended bias, and unjust findings causing discrimination (Tilimbe 2019). As a result, the predictions generated using the above-mentioned model are not immune to criticism. Given specific input variables, the application, for example, will return 'Human' mistake as the reason of fish escape , resulting in unjust or discriminating treatment of farm workers, or vice-versa when the reason returned by the algorithm is 'Natural' and the fish farms trying to find a solution to the 'Natural' escapes when in reality the reason is human error. As a result, caution should be exercised when using the model's conclusions because algorithms are not always inherently ethical.

References

- Lonnie, D., 2022. Linear Regression Models. [lab]. Data Science Development. Robert Gordon University. School of Computing Science and Digital Media, 28th Feb. Available from: https://campusmoodle.rgu.ac.uk/pluginfile.php/5772404/mod_resource/content/8/CMM535%20Lab%2006%20-%20Linear%20Regression%20Models.pdf (https://campusmoodle.rgu.ac.uk/pluginfile.php/5772404/mod_resource/content/8/CMM535%20Lab%2006%20-%20Linear%20Regression%20Models.pdf) [Accessed 26/04/2022].
- Lonnie, D., 2022. Logistic Regression and LDA Models. [lab]. Data Science Development. Robert Gordon University. School of Computing Science and Digital Media, 7th March. Available from: https://campusmoodle.rgu.ac.uk/pluginfile.php/5772406/mod_resource/content/11/CMM535%20Lab%2007%20-

%20Logistic%20Regression%20and%20 (https://campusmoodle.rgu.ac.uk/pluginfile.php/5772406/mod_resource/content/11/CMM535%20Lab%2007%20-%20Logistic%20Regression%20and%20) [Accessed 26/04/2022].

- Lonnie, D., 2022. Revisions of Models and Model Comparison. [lab]. Data Science Development. Robert Gordon University. School of Computing Science and Digital Media, 28th March. Available from: https://campusmoodle.rgu.ac.uk/pluginfile.php/5833704/mod_resource/content/6/lab10solutions.html (https://campusmoodle.rgu.ac.uk/pluginfile.php/5833704/mod_resource/content/6/lab10solutions.html) [Accessed 26/04/2022].
- Lonnie, D., 2022. Deploying models using R Shiny. [lab]. Data Science Development. Robert Gordon University. School of Computing Science and Digital Media, 21st March. Available from: https://rgu.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=d6a154b4-a370-4033-81a5-ae6601164ea0 (https://rgu.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=d6a154b4-a370-4033-81a5-ae6601164ea0) [Accessed 29/04/2022].
- Tilimbe, Jiya. (2019). Ethical Implications of Predictive Risk Intelligence. ORBIT Journal, 2(2). https://doi.org/10.29297/orbit.v2i2.112 (https://doi.org/10.29297/orbit.v2i2.112)