C4.5

Dataset preparation

Dropped Country column. Changed Continent column to factor type. Using sampling with replacement. Doing an 80-20 split. Making 5 groups of such splits and modelling and classifying per group. Using sample() function in the base package to do sampling with replacement with probabilities as 0.8 and 0.2 for a 80-20 split

Function used and choice of parameters

Using J4.8 of the RWeka package for C4.5 implementation.

J48(formula, data, subset, na.action, control = Weka_control(), options = NULL)

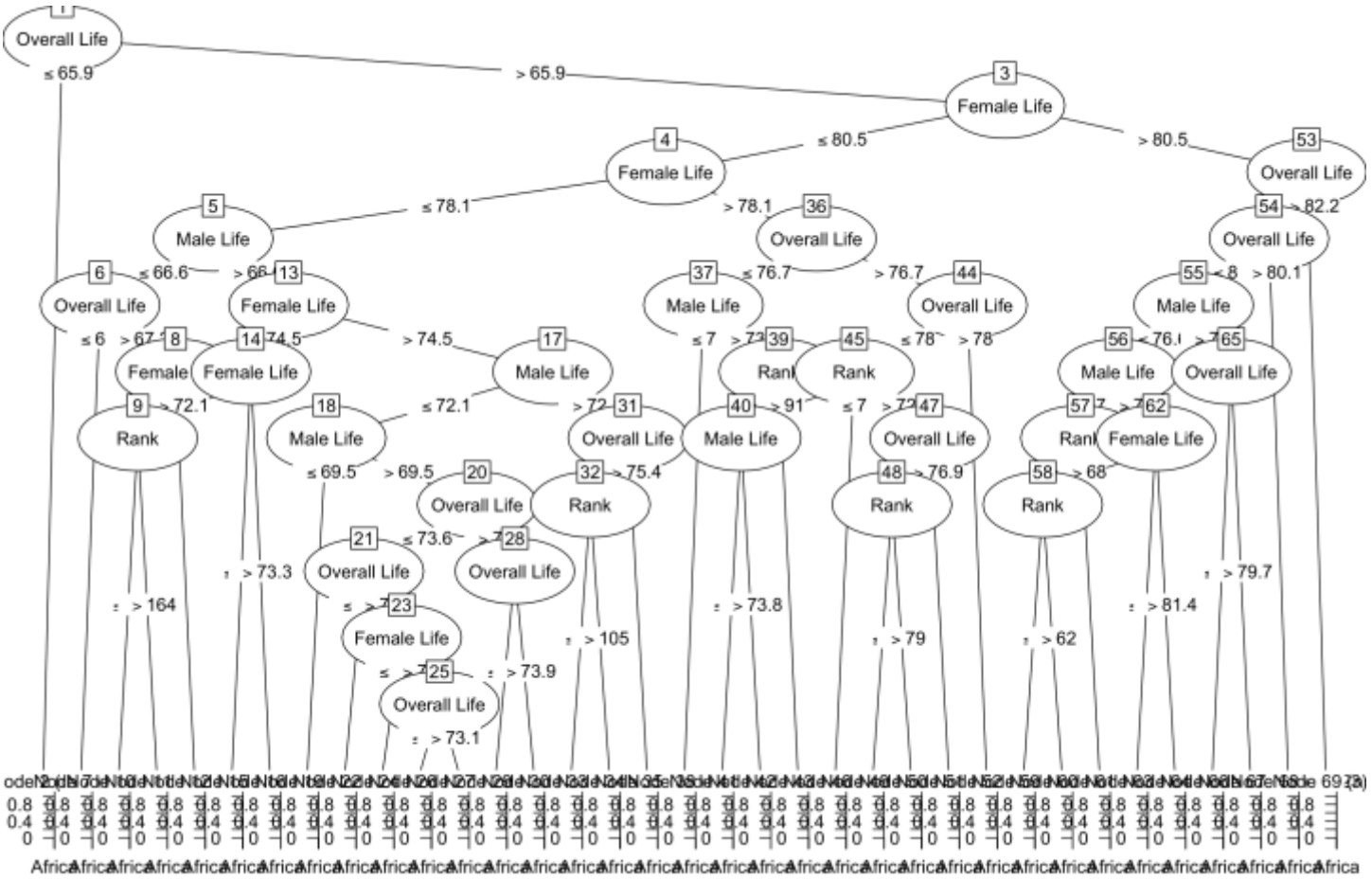
- *formula*: Continent as a function of Overall Life, Male life, Female life and Rank
- *data*: 80% of dataset (training)
- *subset:* null. Accuracy drastically goes down if any column name is added
- *na.action*: not required as we are inputting non-empty data
- *control*: Experimented with using Unpruned Tree (U), setting minimum number of instances (M), reduced error pruning (R) and using Binary splits (B). Could not find a combination of values that might increase accuracy

Classification Results and Analysis

Highest accuracy (in variable *max_acc*): 0.6053
Average accuracy for 5 iterations (in variable *avg_accuracy*): 0.511
Standard deviation among accuracies calculated (in variable *std_accuracies*) : 0.068
Confusion matrix for highest accuracy(best_conf$table):

	Africa	Asia	Europe	North America	Oceania	South America
Africa	14	2	0	1	0	0
Asia	0	4	4	0	1	0
Europe	0	1	1	2	1	1
North America	0	1	0	0	0	0
Oceania	0	0	0	0	3	0
South America	0	0	0	0	1	1

Plot for model that gives best accuracy (plot(best_conf)):



J48 Pruned Tree

Overall Life <= 65.9: Africa (34.0/3.0)
Overall Life > 65.9
| Female Life <= 80.5
| | Female Life <= 78.1
| | | Male Life <= 66.6
| | | | Overall Life <= 67.2: Oceania (4.0/2.0)
| | | | Overall Life > 67.2
| | | | | Female Life <= 72.1
| | | | | | Rank <= 164: South America (2.0)
| | | | | | Rank > 164: Asia (2.0)
| | | | | Female Life > 72.1: Asia (6.0)
| | | Male Life > 66.6
| | | | Female Life <= 74.5
| | | | | Female Life <= 73.3: Asia (8.0/3.0)
| | | | | Female Life > 73.3: Africa (3.0/1.0)
| | | | Female Life > 74.5
| | | | | Male Life <= 72.1
| | | | | | Male Life <= 69.5: Europe (3.0)
| | | | | | Male Life > 69.5
| | | | | | | Overall Life <= 73.6
| | | | | | | | Overall Life <= 72.5: North America (3.0/1.0)

```
| | | | | | | | Overall Life > 72.5
| | | | | | | | | Female Life <= 75.2: Oceania (2.0)
| | | | | | | | | Female Life > 75.2
| | | | | | | | | | Overall Life <= 73.1: Oceania (3.0/1.0)
| | | | | | | | | | Overall Life > 73.1: North America (3.0/1.0)
| | | | | | | Overall Life > 73.6
| | | | | | | | Overall Life <= 73.9: South America (3.0/1.0)
| | | | | | | | Overall Life > 73.9: Europe (2.0/1.0)
| | | | | Male Life > 72.1
| | | | | | Overall Life <= 75.4
| | | | | | | Rank <= 105: North America (2.0)
| | | | | | | Rank > 105: Asia (8.0/1.0)
| | | | | | Overall Life > 75.4: Asia (3.0)
| | Female Life > 78.1
| | | Overall Life <= 76.7
| | | | Male Life <= 73: Europe (12.0/5.0)
| | | | Male Life > 73
| | | | | Rank <= 91
| | | | | | Male Life <= 73.8: Europe (2.0)
| | | | | | Male Life > 73.8: Africa (2.0)
| | | | | Rank > 91: North America (2.0)
| | | Overall Life > 76.7
| | | | Overall Life <= 78
| | | | | Rank <= 72: Asia (6.0/2.0)
| | | | | Rank > 72
| | | | | | Overall Life <= 76.9
| | | | | | | Rank <= 79: Africa (2.0)
| | | | | | | Rank > 79: South America (3.0/1.0)
| | | | | | Overall Life > 76.9: South America (3.0/1.0)
| | | | Overall Life > 78: North America (2.0)
| Female Life > 80.5
| | Overall Life <= 82.2
| | | Overall Life <= 80.1
| | | | Male Life <= 76.6
| | | | | Male Life <= 75.7
| | | | | | Rank <= 68
| | | | | | | Rank <= 62: Europe (4.0/1.0)
| | | | | | | Rank > 62: Oceania (3.0/1.0)
| | | | | | Rank > 68: Europe (5.0)
| | | | | Male Life > 75.7
| | | | | | Female Life <= 81.4: North America (3.0)
| | | | | | Female Life > 81.4: Europe (5.0/2.0)
| | | | Male Life > 76.6
| | | | | Overall Life <= 79.7: Asia (2.0/1.0)
| | | | | Overall Life > 79.7: North America (4.0/1.0)
| | | Overall Life > 80.1: Europe (24.0/6.0)
| | Overall Life > 82.2: Asia (10.0/4.0)
```

Precision, Recall and F1-scores:

Precision (variable: *precision*)

```
Class: Africa  Class: Asia  Class: Europe Class: North America
    0.8235294    0.4444444     0.1666667           0.0000000
Class: Oceania Class: South America
    1.0000000          0.5000000
```

Recall (variable: *recall*)

```
Class: Africa  Class: Asia  Class: Europe Class: North America
          1.0          0.5           0.2                   0.0
Class: Oceania Class: South America
          0.5                0.5
```

F-1 scores (variable: *f1*)

```
Class: Africa  Class: Asia  Class: Europe Class: North America
    0.9032258    0.4705882     0.1818182                   NaN
Class: Oceania Class: South America
    0.6666667          0.5000000
```

RIPPER

Dataset preparation

Same as C4.5. Dropped Country column. Changed Continent column to factor type. Using sampling with replacement. Doing an 80-20 split. Making 5 groups of such splits and modelling and classifying per group. Using sample() function in the base package to do sampling with replacement with probabilities as 0.8 and 0.2 for a 80-20 split

Function used and choice of parameters

Similar to C4.5. Using JRip of the RWeka package for RIPPER implementation.

```
JRip(formula, data, subset, na.action, control = Weka_control(), options = NULL)
```

- *formula*: Continent as a function of Overall Life, Male life, Female life and Rank
- *data*: 80% of dataset (training)
- *subset:* null. Accuracy drastically goes down if any column name is added
- *na.action*: not required as we are inputting non-empty data
- *control*: Experimented with using Unpruned Tree (U), setting minimum number of instances (M), reduced error pruning (R) and using Binary splits (B). Could not find a combination of values that might increase accuracy

Classification Results and Analysis

JRip Rules

Number of Rules : 4

(Rank <= 136) and (Rank >= 134) => Continent=Oceania (3.0/0.0)
(Rank >= 171) => Continent=Africa (36.0/4.0)
(Female Life <= 78) => Continent=Asia (50.0/24.0)
 => Continent=Europe (96.0/54.0)

Highest accuracy: 0.6316
Average accuracy for 5 iterations: 0.505
Standard deviation among accuracies calculated: 0.095
Confusion matrix for highest accuracy(best_conf$table):

	Africa	Asia	Europe	North America	Oceania	South America
Africa	14	2	0	0	1	0
Asia	0	5	0	1	4	0
Europe	0	1	5	2	1	2
North America	0	0	0	0	0	0
Oceania	0	0	0	0	0	0
South America	0	0	0	0	0	0

Precision, Recall and F1-scores:

Precision

Class: Africa Class: Asia Class: Europe Class: North America
0.8235294 0.5000000 0.4545455 NA
Class: Oceania Class: South America
NA NA

Recall

Class: Africa Class: Asia Class: Europe Class: North America
1.000 0.625 1.000 0.000
Class: Oceania Class: South America
0.0 0.000

F1 measure

Class: Africa Class: Asia Class: Europe Class: North America

```
 0.9032258 0.5555556 0.6250000        NA
Class: Oceania Class: South America
        NA              NA
```

knn- K nearest neighbours

Dataset preparation

Training and Test set generation

Same as C4.5. Dropped Country column. Changed Continent column to factor type. Using sampling with replacement. Doing an 80-20 split. Making 5 groups of such splits and modelling and classifying per group. Using sample() function in the base package to do sampling with replacement with probabilities as 0.8 and 0.2 for a 80-20 split

Preprocessing

Centering and Scaling the data before creating model

Function used and choice of parameters

First create the train control parameter using cross validation as the choice of method. Use it as a control parameter to the train function along with the class function (Continent as a function of others), training data, knn as the choice of classifer, information that we centered and scaled the data during preprocessing and the tuneLength. The *train* function automatically tunes all the hyper-parameters required for best accuracy. So, that does not need to be done manually

```
ctrl <- trainControl(method="cv")
model <- train(Continent~., data = training, method = "knn", trControl = ctrl, preProcess = c("center","scale"), tuneLength = 20)
```

trainControl

- method = cv: Used cross-validation as the method of choice (no need to sue repeatedcv as we are already looping around 5 times)

train

- *formula*: Continent as a function of Overall Life, Male life, Female life and Rank
- *data*: 80% of dataset (training)
- *method*: knn: K nearest neighbours
- *trControl*: result of trainControl
- *preprocess*: How we pre-processed our data
- *tuneLength*: granularity in tuning grid; general good values I found were 10-20

Classification Results and Analysis

Highest accuracy: 0.658
Average accuracy for 5 iterations: 0.55
Standard deviation among accuracies calculated: 0.101
Confusion matrix for highest accuracy(best_conf$table):

	Africa	Asia	Europe	North America	Oceania	South America
Africa	14	2	0	0	1	0
Asia	0	4	0	1	3	0
Europe	0	0	5	0	1	1
North America	0	2	0	1	1	0
Oceania	0	0	0	0	0	0
South America	0	0	0	1	0	1

Precision, Recall and F1-scores:

Precision

```
     Class: Africa       Class: Asia     Class: Europe Class: North America
         0.8235294         0.5000000         0.7142857         0.2500000
    Class: Oceania Class: South America
                NA         0.5000000
```

Recall

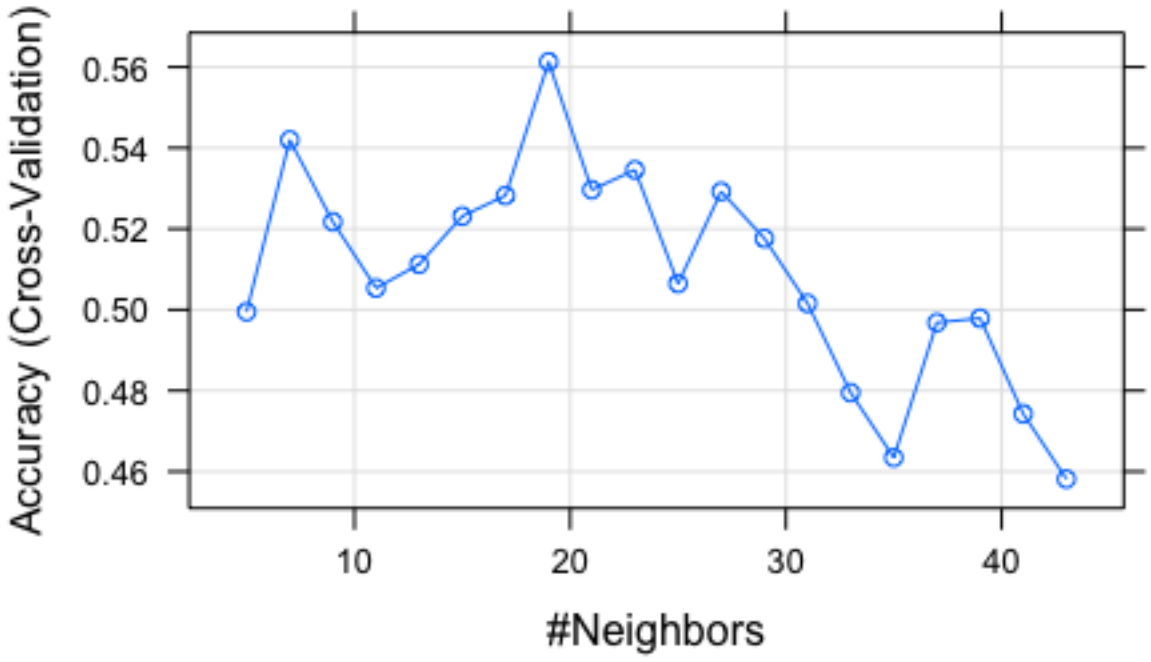
```
     Class: Africa       Class: Asia     Class: Europe Class: North America
         1.0000000         0.5000000         1.0000000         0.3333333
    Class: Oceania Class: South America
         0.0000000         0.5000000
```

F1 measure

```
     Class: Africa       Class: Asia     Class: Europe Class: North America
         0.9032258         0.5000000         0.8333333         0.2857143
    Class: Oceania Class: South America
                NA         0.5000000
```

Accuracy vs k plot

Best value of accuracy obtained when k is **19.** (best_model$bestTune)



SVM

Dataset preparation

Training and Test set generation

Same as C4.5. Dropped Country column. Changed Continent column to factor type. Using sampling with replacement. Doing an 80-20 split. Making 5 groups of such splits and modelling and classifying per group. Using sample() function in the base package to do sampling with replacement with probabilities as 0.8 and 0.2 for a 80-20 split

Preprocessing

Centering and Scaling the data before creating model

Function used and choice of parameters

First create the train control parameter using cross validation as the choice of method. Use it as a control parameter to the train function along with the class function (Continent as a function of others), training data, *svmLinear* as the choice of classifier, information that we centered and scaled the data during preprocessing, the tuneLength and the tuneGrid. Similar to knn, the *train* function automatically tunes all the hyper-parameters required for best accuracy. So, that does not need to be done manually

```
ctrl <- trainControl(method="cv")
grid <- expand.grid(C = c(seq(0.25,10, by=0.25)))
model <- train(Continent~., data = training, method = "svmLinear", trControl = ctrl, preProcess = c("center","scale"), tuneLength = 20, tuneGrid = grid)
```

trainControl

- method = cv: Used cross-validation as the method of choice (no need to use repeatedcv as we are already looping around 5 times)

tuneGrid

- Used in the model so that we can create an accuracy vs cost plot

train

- *formula*: Continent as a function of Overall Life, Male life, Female life and Rank
- *data*: 80% of dataset (training)
- *method*: svmLinear. Using SVM with Linear Kernel
- *trControl*: result of trainControl
- *preprocess*: How we pre-processed our data
- *tuneLength*: granularity in tuning grid; general good values I found were 10-20
- *tuneGrid*: Using the data-frame generated in previous step; This is used to create the Accuracy vs cost graph

Classification Results and Analysis

Highest accuracy: 0.605
Average accuracy for 5 iterations: 0.521
Standard deviation among accuracies calculated: 0.065
Confusion matrix for highest accuracy(best_conf$table):

	Africa	Asia	Europe	North America	Oceania	South America
Africa	14	3	0	0	2	0
Asia	0	4	0	2	3	0
Europe	0	1	5	1	1	2
North America	0	0	0	0	0	0
Oceania	0	0	0	0	0	0
South America	0	0	0	0	0	0

Precision, Recall and F1-scores:

Precision

```
Class: Africa  Class: Asia  Class: Europe Class: North America
    0.7368421    0.4444444     0.5000000                  NA
Class: Oceania Class: South America
            NA                 NA
```

Recall

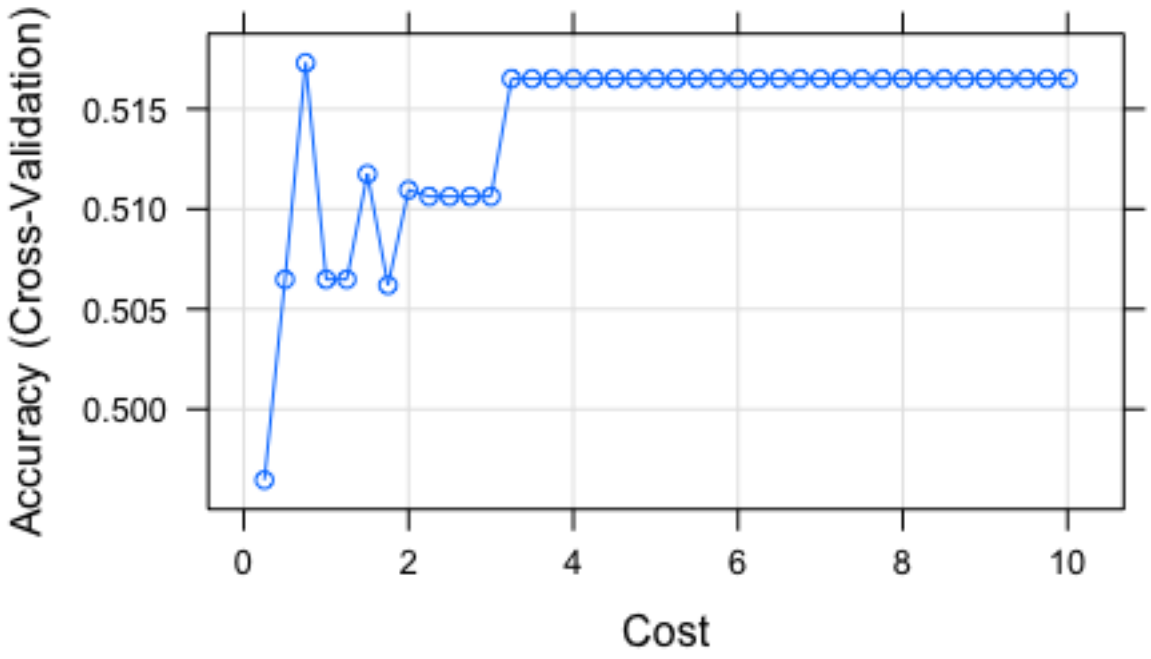
```
Class: Africa  Class: Asia  Class: Europe Class: North America
          1.0          0.5             1.0                  0.0
Class: Oceania Class: South America
           0.0                0.0
```

F1 measure

```
Class: Africa  Class: Asia  Class: Europe Class: North America
    0.8484848    0.4705882     0.6666667                  NA
Class: Oceania Class: South America
            NA                 NA
```

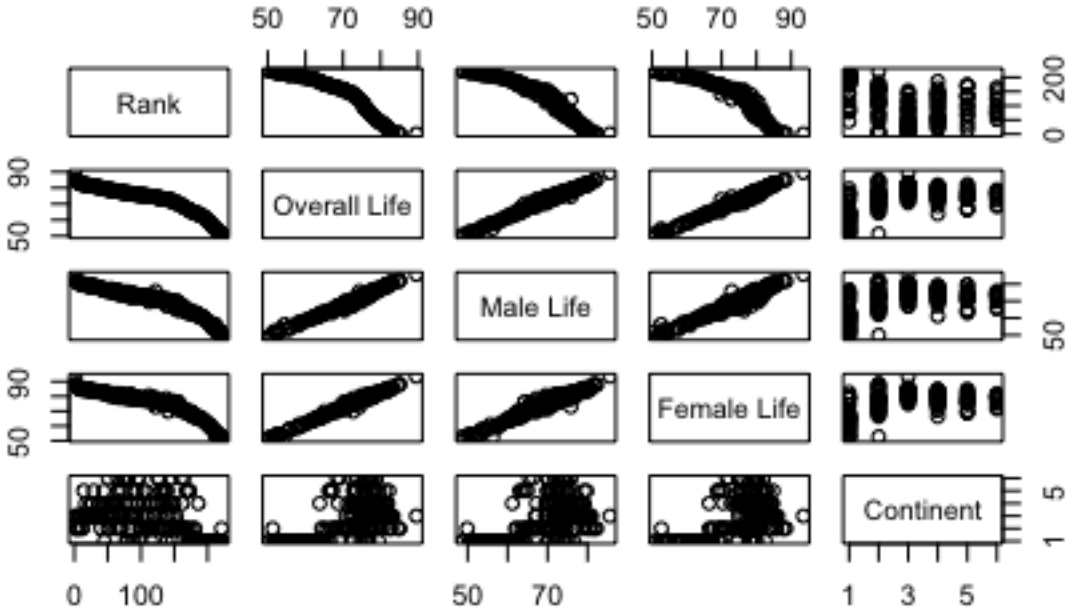
Accuracy vs cost plot

Best value of accuracy obtained when cost is around **0.75** (best_model$bestTune)



Conclusion and reference list

Relationship of data elements to each other



Conclusion and Results

1. Highest accuracies for each model:
    a. C4.5 - 0.6053
    b. RIPPER - 0.6316
    c. knn – 0.658
    d. svm- 0.605
2. Hence, on the basis of the tests done:
    accuracy(knn) > accuracy(RIPPER) > accuracy (svm) == accuracy(C4.5)
3. Africa is classified near perfectly everytime. This is because it's life ranges are generally non-overlapping with other continents'. The column generally has relatively better precision, recall and F1-measures too
4. As seen in the data relationship chart, Rank does not seem to have a good co-relation with Continent
5. For knn, using k = 19 gives the best accuracy as shown in the above plot
6. For svm, using cost factor of svmLinear model as 0.75 works well and gives the highest accuracy
7. Overall Life, Male Life and Female life are highly correlated

References

1. https://www.rdocumentation.org/packages/RWeka/versions/0.4-34/topics/Weka_classifier_trees
2. https://en.wikipedia.org/wiki/List_of_countries_by_life_expectancy
3. kknn: Hechenbichler K. and Schliep K.P. (2004) Weighted k-Nearest-Neighbor Techniques and Ordinal Classification, Discussion Paper 399, SFB 386, Ludwig-Maximilians University Munich
4. R documentation