Flag Dataset

Statistics for Business Analytics II

Project 2

Eva Giannatou
BAFT 1616

Contents

Figures

1. Introduction

The data set contains details of various nations and their flags. In the first part of this project, we will try predicting the religion of a country from its flag's characteristics. In order to efficiently train the models we will need to transform the data set into proper form and we will also need to reduce the number of attributes which will be used as input in the models. It is crucial that we accurately specify, which characteristics of a flag are correlated to a country's religion and therefore we might be able to predict it. In the second part of the project clustering methods will be used to find clusters of flags with respect to their characteristics. We will then discover the common characteristics of the countries within each cluster. We hope that there will be a connection between the clusters and the religions of the within countries.

2. Dataset

The dataset consists of 30 variables, 10 of them are numeric and the rest of them are either Boolean or nominal. Figure 1 describes the names and the values of each variable.

1. name: Name of the country concerned
2. landmass: 1=N.America, 2=S.America, 3=Europe, 4=Africa, 4=Asia, 6=Oceania
3. zone: Geographic quadrant, based on Greenwich and the Equator; 1=NE, 2=SE, 3=SW, 4=NW
4. area: in thousands of square km
5. population: in round millions
6. language: 1=English, 2=Spanish, 3=French, 4=German, 5=Slavic, 6=Other Indo-European, 7=Chinese, 8=Arabic, 9=Japanese/Turkish/Finnish/Magyar, 10=Others
7. religion: 0=Catholic, 1=Other Christian, 2=Muslim, 3=Buddhist, 4=Hindu, 5=Ethnic, 6=Marxist, 7=Others
8. bars: Number of vertical bars in the flag
9. stripes: Number of horizontal stripes in the flag
10. colours: Number of different colours in the flag
11. red: 0 if red absent, 1 if red present in the flag
12. green: same for green
13. blue: same for blue
14. gold: same for gold (also yellow)
15. white: same for white
16. black: same for black
17. orange: same for orange (also brown)
18. mainhue: predominant colour in the flag (tie-breaks decided by taking the topmost hue, if that fails then the most central hue, and if that fails the leftmost hue)
19. circles: Number of circles in the flag
20. crosses: Number of (upright) crosses
21. saltires: Number of diagonal crosses
22. quarters: Number of quartered sections
23. sunstars: Number of sun or star symbols
24. crescent: 1 if a crescent moon symbol present, else 0
25. triangle: 1 if any triangles present, 0 otherwise
26. icon: 1 if an inanimate image present (e.g., a boat), otherwise 0
27. animate: 1 if an animate image (e.g., an eagle, a tree, a human hand) present, 0 otherwise
28. text: 1 if any letters or writing on the flag (e.g., a motto or slogan), 0 otherwise
29. topleft: colour in the top-left corner (moving right to decide tie-breaks)
30. botright: Colour in the bottom-left corner (moving left to decide tie-breaks)

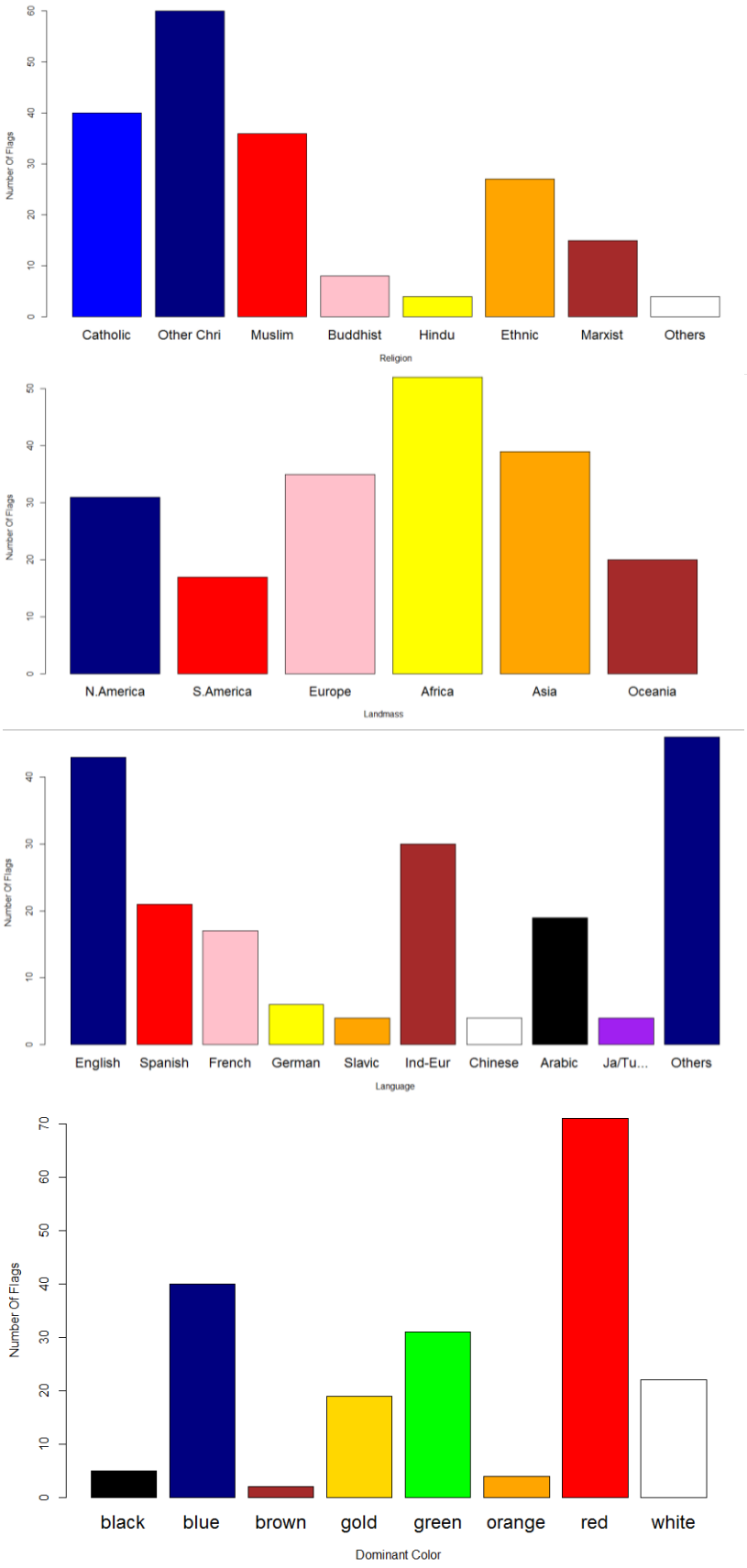
Figure 1 Flag dataset description

2.1. Transformation

```
$ religion       : int  2 6 2 1 0 5 1 1 0 0 ...
$ bars           : int  0 0 2 0 3 0 0 0 0 0 ...
$ stripes        : int  3 0 0 0 0 2 1 1 3 3 ...
$ colours        : int  5 3 3 5 3 3 3 5 2 3 ...
$ red            : int  1 1 1 1 1 1 0 1 0 0 ...
$ green          : int  1 0 1 0 0 0 0 0 0 0 ...
$ blue           : int  0 0 0 1 1 0 1 1 1 1 ...
$ gold           : int  1 1 0 1 1 1 0 1 0 1 ...
$ white          : int  1 0 1 1 0 0 1 1 1 1 ...
$ black          : int  1 1 0 0 0 1 0 1 0 0 ...
$ orange         : int  0 0 0 1 0 0 1 0 0 0 ...
$ circles        : int  0 0 0 0 0 0 0 0 0 0 ...
$ crosses        : int  0 0 0 0 0 0 0 0 0 0 ...
$ saltires       : int  0 0 0 0 0 0 0 0 0 0 ...
$ quarters       : int  0 0 0 0 0 0 0 0 0 0 ...
$ sunstars       : int  1 1 1 0 0 1 0 1 0 1 ...
$ crescent       : int  0 0 1 0 0 0 0 0 0 0 ...
$ triangle       : int  0 0 0 1 0 0 0 1 0 0 ...
$ icon           : int  1 0 0 1 0 1 0 0 0 0 ...
$ animate        : int  0 1 0 1 0 0 1 0 0 0 ...
$ text           : int  0 0 0 0 0 0 0 0 0 0 ...
$ dominant_green : num  1 0 1 0 0 0 0 0 0 0 ...
$ dominant_red   : num  0 1 0 0 0 1 0 1 0 0 ...
$ dominant_blue  : num  0 0 0 1 0 0 0 0 1 1 ...
$ dominant_gold  : num  0 0 0 0 1 0 0 0 0 0 ...
$ dominant_white : num  0 0 0 0 0 0 1 0 0 0 ...
$ dominant_orange: num  0 0 0 0 0 0 0 0 0 0 ...
$ dominant_black : num  0 0 0 0 0 0 0 0 0 0 ...
$ dominant_brown : num  0 0 0 0 0 0 0 0 0 0 ...
$ topleft_black  : num  1 0 0 0 0 0 0 1 0 0 ...
$ topleft_red    : num  0 1 0 0 0 1 0 0 0 0 ...
$ topleft_green  : num  0 0 1 0 0 0 0 0 0 0 ...
$ topleft_blue   : num  0 0 0 1 1 0 0 0 1 1 ...
$ topleft_white  : num  0 0 0 0 0 0 1 0 0 0 ...
$ topleft_orange : num  0 0 0 0 0 0 0 0 0 0 ...
$ topleft_gold   : num  0 0 0 0 0 0 0 0 0 0 ...
$ botright_green : num  1 0 0 0 0 0 0 0 0 0 ...
$ botright_red   : num  0 1 0 1 1 0 0 1 0 0 ...
$ botright_white : num  0 0 1 0 0 0 0 0 0 0 ...
$ botright_black : num  0 0 0 0 0 1 0 0 0 0 ...
$ botright_blue  : num  0 0 0 0 0 0 1 0 1 1 ...
$ botright_gold  : num  0 0 0 0 0 0 0 0 0 0 ...
$ botright_orange: num  0 0 0 0 0 0 0 0 0 0 ...
$ botright_brown : num  0 0 0 0 0 0 0 0 0 0 ...
```

Figure 2 Transformed dataset: Nominal variables were transformed to binary

The dataset was transformed in order to be used as input when training the models. All the factors consisting the nominal variables were converted to binary columns. This resulted the dataset which is represented below (Figure 2).



2.2. Data visualization

For better understanding the data set, the factors of three important variables were visualized (Figure 3). The following bar plots represent the different religions, landmasses and languages and the number of countries which are related to them. Moreover, the last bar plot represents the dominant colors of the flags. Other Christians, Catholics and Muslims are the most common religions among countries in this dataset. Most of the data set's countries belong to Africa's landmass and the most commonly spoken language is English. Finally, the most common dominant color in flags is red.

Figure 3 Attribute visualization

3. Attribute selection

Three attribute selection methods were used and their outputs were cross validated. Our final dataset consists of the attributes which were selected from at least two out of these three methods.

3.1. Caret package

The first package that was used for attribute selection is “caret”. Caret package provides tools for automatically reporting the relevance and importance of attributes in the data and for selecting the most important features. It also provides the findCorrelation which analyzes a correlation matrix of the attributes and determines which ones can be removed. It is suggested that attributes with an absolute correlation of 0.75 or higher should be removed. The importance of features was estimated by building a model and the variable importance was estimated using a Learning Vector Quantization (LVQ) model, which is a supervised classification algorithm. In Figure 4 the importance of each attribute is represented in ascending order. According to caret package “crosses”, “dominant_blue” and “white” are the most important attributes as far as the prediction of religion is concerned.

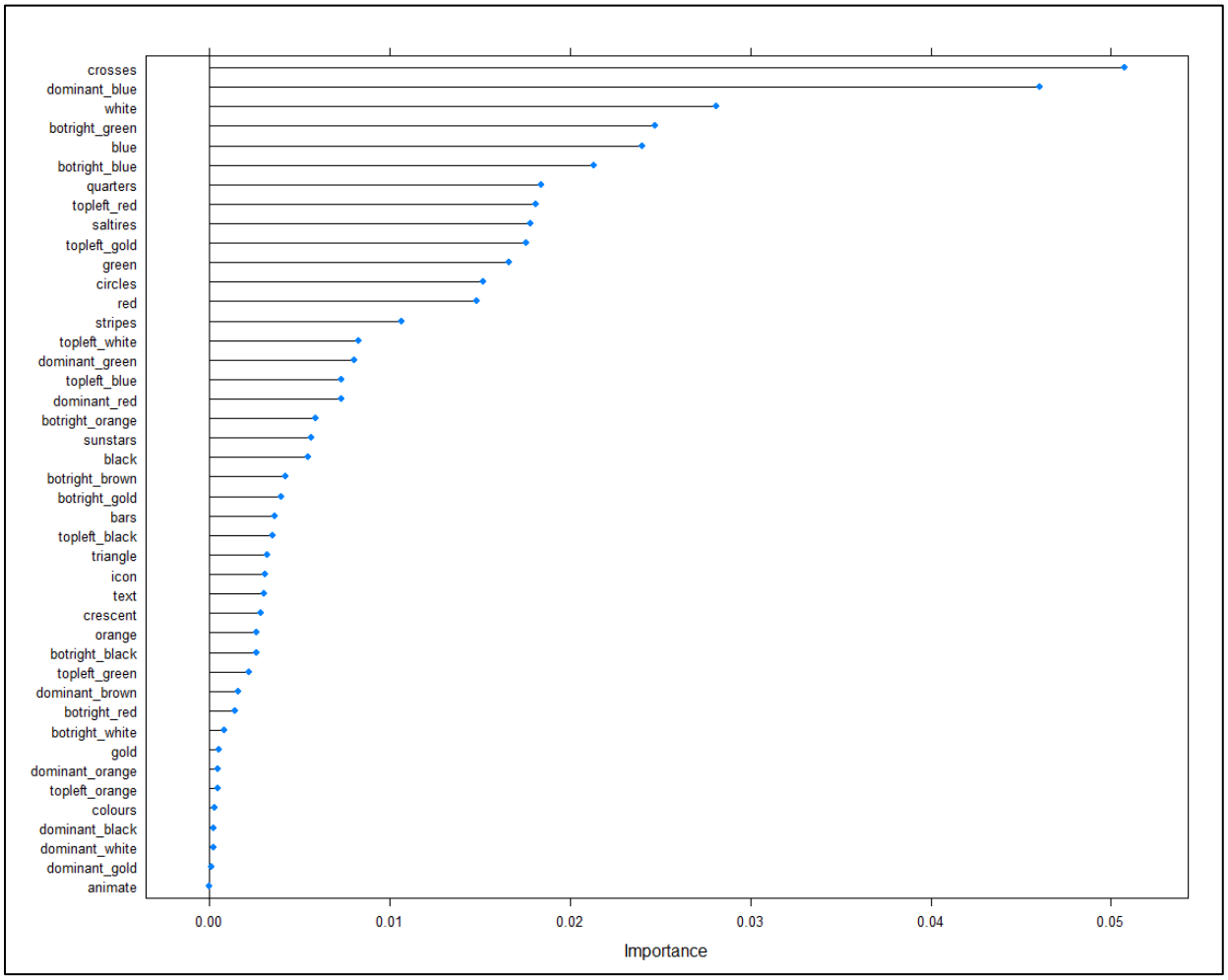


Figure 4 Attribute importance: Caret package

3.2. Boruta package

The second package that was used for attribute selection is “Boruta”, found in the “CRAN” repository. Boruta is a feature selection algorithm and it works as a wrapper algorithm around Random Forest. This package is also used for removing highly correlated attributes. It follows an all-relevant feature selection method where it captures all features which are in some circumstances relevant to the outcome variable. Firstly, it adds randomness to the given data set by creating shuffled copies of all features (which are called shadow features). Then, it trains a random forest classifier on the extended data set and applies a feature importance measure (the default is Mean Decrease Accuracy) to evaluate the importance of each feature where higher means more important. At every iteration, it checks whether a real feature has a higher importance than the best of its shadow features (i.e. whether the feature has a higher Z score than the maximum Z score of its shadow features) and constantly removes features which are deemed highly unimportant. Finally, the algorithm stops either when all features gets confirmed or rejected or it reaches a specified limit of random forest runs.

Boruta performed 99 iterations in 25.54982 secs. Boruta’s results confirmed 20 attributes as important (animate, bars, black, blue, botright_black and 15 more) and 23 attributes as unimportant (botright_brown, botright_gold, botright_orange, botright_red, botright_white and 18 more). According to Figure 5 blue boxplots correspond to minimal, average and maximum Z score of a shadow attribute. Red, yellow and green boxplots represent Z scores of rejected, tentative and confirmed attributes respectively.

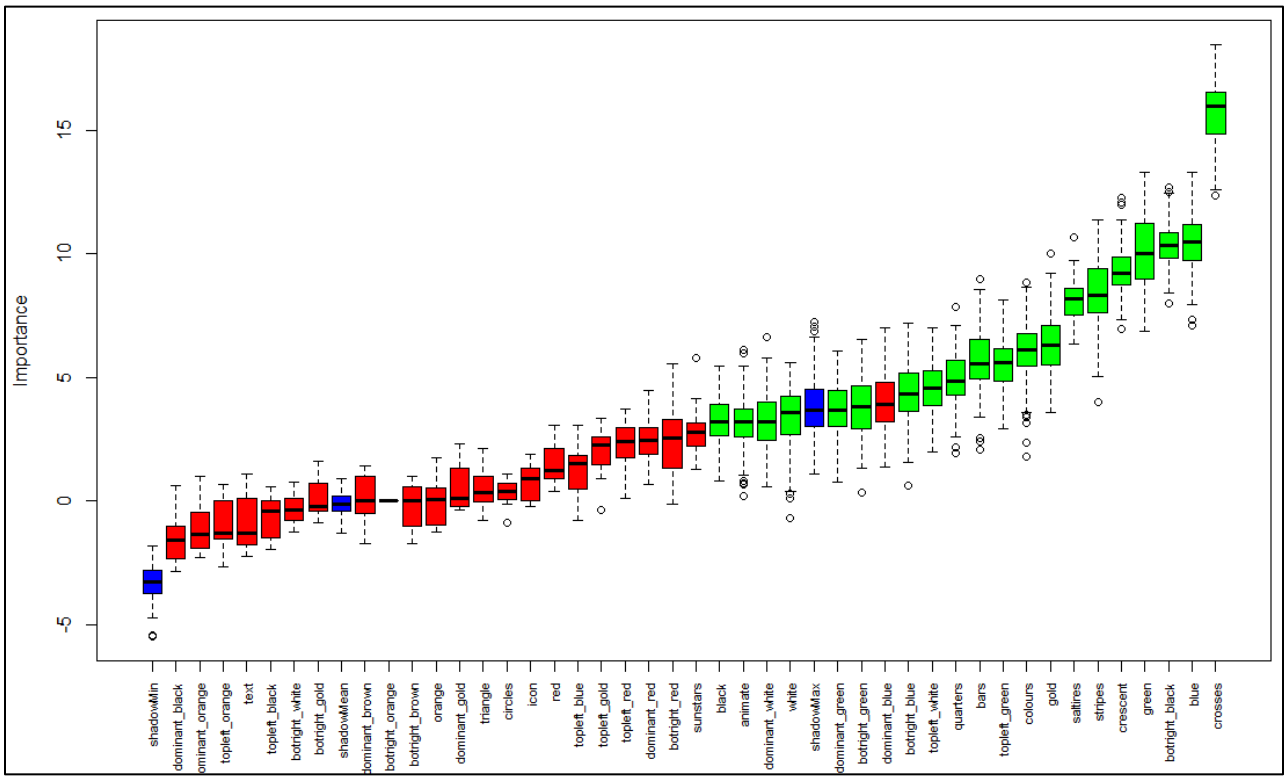


Figure 5 Attribute importance: Boruta package

3.3. Random Forest package

Finally a Random Forest model using all the flag related attributes was trained and a variable importance plot was obtained by evaluating the resulted trees. Figure 6 represents the *mean decrease in node impurity* when further attributes are being added in the model (and not the *mean decrease in accuracy*).

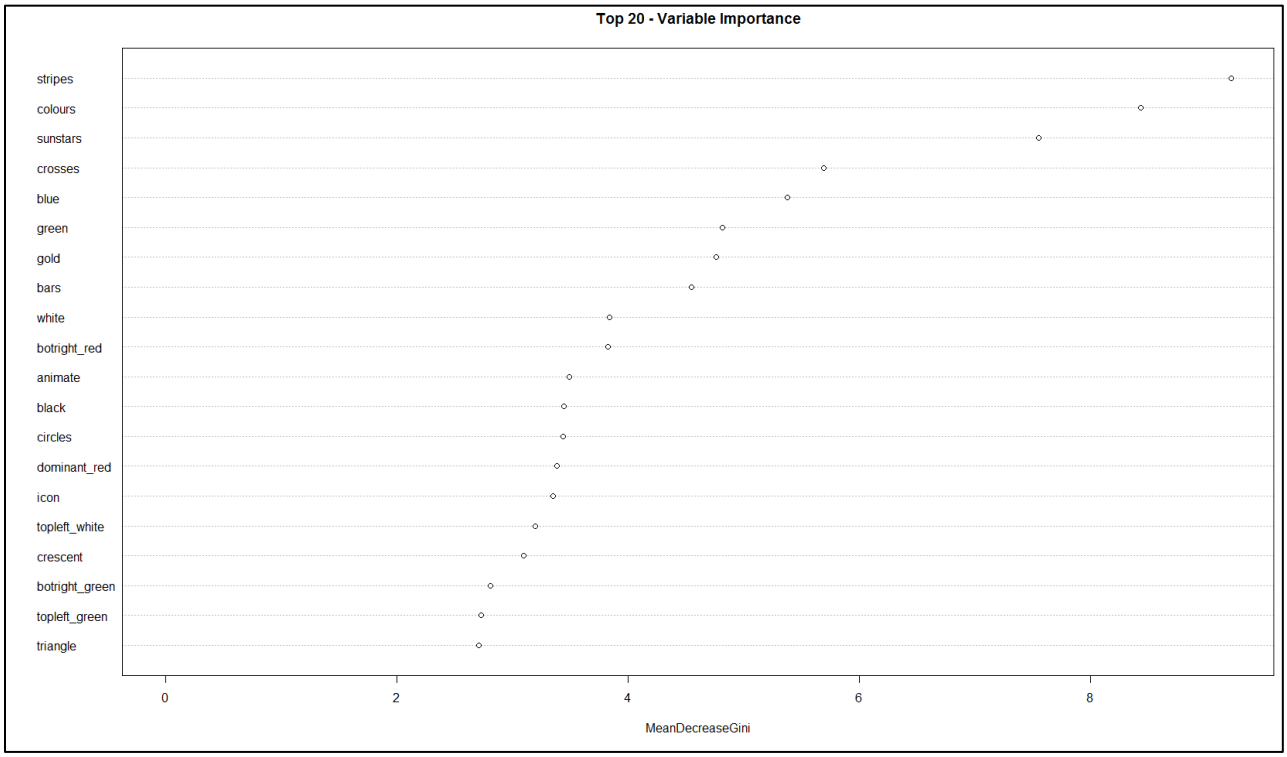


Figure 6 Attribute importance: randomForest package

3.4. Combined attribute selection

Figure 7 reports the selected attributes from each one of the three attribute reduction methods. Attributes that were selected from at least two out of three methods can be found in the last column of figure. These attributes will be used for training both the classification and clustering models.

Caret	Boruta	Random Forest	Final Selection
crosses	crosses	stripes	**crosses**
dominant_blue	blue	colours	**white**
white	botright_black	sunstars	**botright_green**
botright_green	green	crosses	**blue**
blue	crescent	blue	**botright_blue**
botright_blue	stripes	green	**dominant_blue**
topleft_red	saltires	gold	**quarters**
quarters	gold	bars	**saltires**
saltires	colours	white	**green**
topleft_gold	topleft_green	botright_red	**circles**
green	bars	animate	**bars**
circles	quarters	black	**stripes**
red	topleft_white	circles	**colours**
stripes	botright_blue	dominant_red	**gold**
topleft_white	botright_green	icon	**black**
dominant_green	dominant_green	topleft_white	**crescent**
topleft_blue	white	crescent	**animate**
dominant_red	dominant_white	botright_green	**topleft_white**
botright_orange	animate	topleft_green	
sunstarts	black	triangle	

Figure 7 Attribute selection results

4. Dimension reduction

4.1. Principal component analysis

Our new data set consists of the 20 most correlated to religion attributes. This dataset's variables were further reduced by using the principal component analysis method (PCA). PCA is a dimensionality reduction technique that is widely used in data analysis. PCA projects p-dimensional data into a q-dimensional sub-space $(q \leq p)$ in a way that minimizes the residual sum of squares (RSS) of the projection. That is, it minimizes the sum of squared distances from the points to their projections. It turns out that this is equivalent to maximizing the covariance matrix (both in trace and determinant) of the projected data. Plotting $(1 - R^2)$ versus the number of components was used to visualize the number of principal components that retain most of the variability contained in the original data (Figure 8). A principal component is a normalized linear combination of the original predictors in a data set and the first principal component captures the maximum variance and maximum information in the data set. Every further principal component contains less information than the previous ones. In figure 8 we can see that the number of resulted components is 18 which equals to the dimension number of the previous dataset. However, now, the first 7 principal components encapsulate 91.5% of the total information in the dataset.

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5    PC6     PC7     PC8     PC9
Standard deviation     2.3574 1.4062 1.0268 0.76461 0.54080 0.4954 0.47565 0.43536 0.40101
Proportion of Variance 0.5118 0.1821 0.0971 0.05384 0.02694 0.0226 0.02084 0.01746 0.01481
Cumulative Proportion  0.5118 0.6939 0.7910 0.84485 0.87179 0.8944 0.91523 0.93268 0.94749
                          PC10    PC11    PC12    PC13    PC14    PC15    PC16    PC17    PC18
Standard deviation     0.34440 0.30702 0.29050 0.26879 0.25338 0.22245 0.19747 0.15909 0.15044
Proportion of Variance 0.01092 0.00868 0.00777 0.00665 0.00591 0.00456 0.00359 0.00233 0.00208
Cumulative Proportion  0.95842 0.96710 0.97487 0.98152 0.98744 0.99199 0.99558 0.99792 1.00000
```

Figure 8 Principal component analysis

Concluding, the first 10 principal components were chosen to replace the previous variables (flag characteristics) in the dataset while the "religion" column which we will try to predict, remained the same. The dataset was split into training and testing set with ratio 80% and 20% accordingly. The training set was used for model training and the test set for evaluating the performance of the prediction. Figure 9 represents the training set which contains 156 rows and 11 columns.

```
'data.frame':   156 obs. of  11 variables:
 $ religion: int  6 1 0 1 0 0 1 2 2 1 ...
 $ PC1     : num  -1.264 1.273 -1.136 -0.104 1.172 ...
 $ PC2     : num  -1.523 -0.725 0.222 -0.565 2.635 ...
 $ PC3     : num  -0.756 -1.533 -0.355 -1.587 -1.65 ...
 $ PC4     : num  -1.019 -0.573 -2.758 0.438 -0.413 ...
 $ PC5     : num  -1.909 -0.895 -0.796 -1.205 0.662 ...
 $ PC6     : num  -1.311 1.038 -0.226 -0.483 0.725 ...
 $ PC7     : num  -0.2599 -0.0319 -0.5769 0.2665 0.2759 ...
 $ PC8     : num  0.544 -0.83 -1.43 -0.243 0.667 ...
 $ PC9     : num  -0.622 0.15 0.895 -0.703 0.157 ...
 $ PC10    : num  -0.902 -1.741 0.859 0.212 -0.281 ...
```

Figure 9 Final dataset: Contains principal components and religion

5. Classification models

Several classification models were tested and evaluated but in this report only the top 5 of them as far as accuracy is concerned, will be discussed. Each model performed better with different number of principal components that the others.

5.1. Rpart classification

The rpart programs build classification or regression models of a very general structure using a two stage procedure; the resulting models can be represented as binary trees. The tree is built by the following process: first the single variable is found which best splits the data into two groups ('best' will be defined later). The data is separated, and then this process is applied separately to each sub-group, and so on recursively until the subgroups either reach a minimum size or until no improvement can be made. The resultant model is, with a certainty, too complex, and the question arises as it does with all stepwise procedures of when to stop. The second stage of the procedure consists of using cross-validation to trim back the full tree. In this case the full tree had 18 terminal regions and 5 principal components were for constructing the tree. In Figure 10 we can see that if PC3>=0.32 and PC1>=-1.1 and PC4<0.11 then the predicted class is class number 1. This tree model was able to predict 84% of the countries' religions in the test set. The confusion matrix in figure 11 represents the classification made by the rpart model. The model's accuracy on the test set equals to 84% which is calculated as the sum of the confusion matrix diagonal divided by the total number of classified objects.

Rpart Classification:

- **Before pruning accuracy 84%**
- **Variables actually used in tree construction: PC1 PC2 PC3 PC4 PC5**
- **After pruning accuracy 76%**
- **Variables actually used in pruned tree construction: PC1 PC2 PC3 PC4**

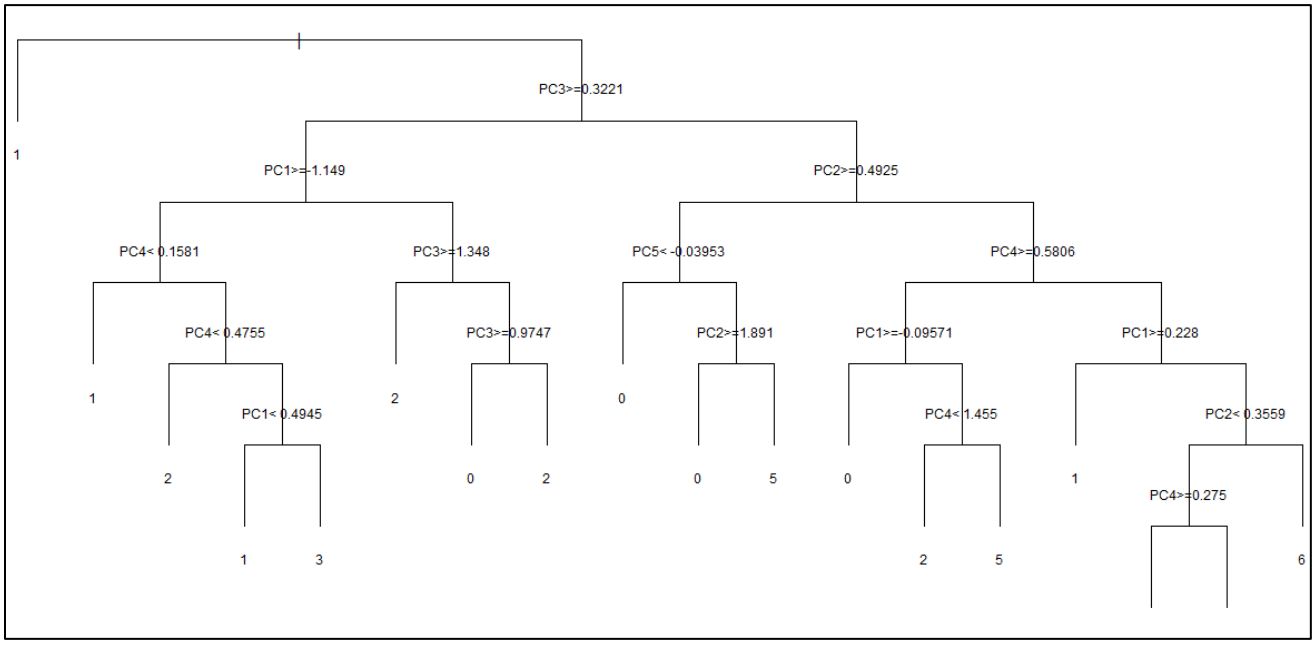


Figure 10 rpart classification: decision tree before pruning

rpartact \ rpartpred	0	1	2	3	4	5	6	7
0	7	1	0	0	0	1	0	0
1	0	11	0	0	0	0	0	0
2	0	0	7	0	0	0	0	0
3	1	0	0	1	0	0	0	0
5	0	1	1	0	0	6	0	0
6	1	0	0	0	0	0	0	0

Figure 11 rpart classification: confusion matrix before tree pruning

A cross validated estimate of risk was computed for a nested set of sub trees; this final model was that sub tree with the lowest estimate of risk (Figure 12). This procedure of reducing the complexity of a tree considering the lowest estimate of risk is called pruning. Figure represents the pruned tree. It has 10 terminal regions 8 less than the full and accuracy reduced by 8%, predicting 76% of the religions correctly. Only the first 4 principal components were for constructing this tree (Figure 12).

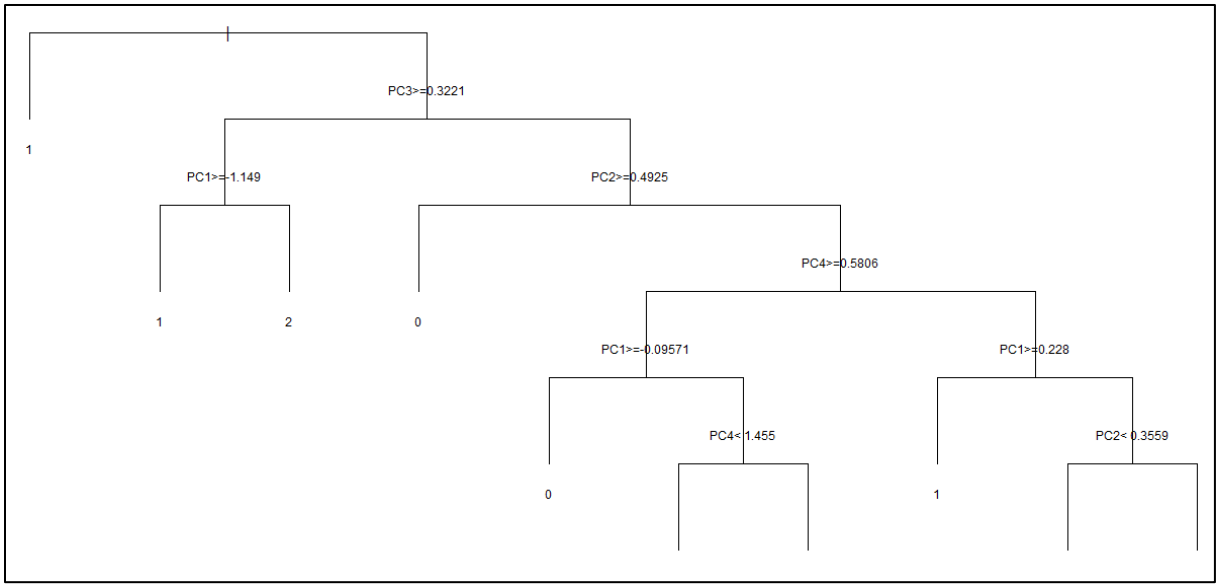


Figure 12 rpart classification: decision tree after pruning

rpartact	rpartpred 0	1	2	3	4	5	6	7
0	7	1	0	0	0	1	0	0
1	0	11	0	0	0	0	0	0
2	0	1	6	0	0	0	0	0
3	1	1	0	0	0	0	0	0
5	2	0	1	0	0	5	0	0
6	0	0	1	0	0	0	0	0

Figure 13 rpart classification: confusion matrix after tree pruning

5.2. Bagging cart classification

Bagging Trees (BT) are bootstrapping-based techniques that improve the prediction accuracy of tree models by constructing a set of trees by re-sampling from the data and 'averaging' the predictions of each. As with any other model averaging technique, it comes at a cost of model interpretability. BT is available in the *ipred* library and is built off of the *rpart* implementation.

The tree creating function for BT is bagging. The function then does what rpart does, except it first generates 30 (*nbagg*) new datasets randomly selected (with replacement) from the original data, and creates a separate tree for each. The model grows the maximum number of leaves. The *coob=T* argument tells R to use the 'out-of-bag' (left aside) values to calculate the misclassification rate. This tree model was too complex to be plotted but part of it was described as text in figure. Figure represents the tree branches as well as the conditions which accompany them. Lines which end with "*" are terminal regions. For example, if PC9 is less than -1.19 and PC2 is less than -1.64 then the predicted class is class number 1. The accuracy of this classification method equals to 55%.

Bagging Tree Classification:

- **Accuracy 55%**
- **Variables actually used in tree construction: PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8 PC9**

```
1) root 156 112 1 (0.22 0.28 0.17 0.064 0.019 0.13 0.09 0.019)
  2) PC9< -1.188266 23   6 2 (0.043 0.13 0.74 0.043 0 0.043 0 0)
    4) PC2< -1.647057 3   0 1 (0 1 0 0 0 0 0 0) *
    5) PC2>=-1.647057 20   3 2 (0.05 0 0.85 0.05 0 0.05 0 0)
     10) PC1>=0.2171393 1   0 0 (1 0 0 0 0 0 0 0) *
     11) PC1< 0.2171393 19   2 2 (0 0 0.89 0.053 0 0.053 0 0)
       22) PC2< 2.239798 18   1 2 (0 0 0.94 0 0 0.056 0 0)
         44) PC9< -1.236183 16   0 2 (0 0 1 0 0 0 0 0) *
         45) PC9>=-1.236183 2   1 2 (0 0 0.5 0 0 0.5 0 0)
           90) PC1>=-0.4469593 1   0 2 (0 0 1 0 0 0 0 0) *
           91) PC1< -0.4469593 1   0 5 (0 0 0 0 0 1 0 0) *
       23) PC2>=2.239798 1   0 3 (0 0 0 1 0 0 0 0) *
  3) PC9>=-1.188266 133  92 1 (0.26 0.31 0.075 0.068 0.023 0.14 0.11 0.023)
    6) PC3< 0.3506378 79  48 0 (0.39 0.2 0.013 0.038 0.038 0.18 0.14 0)
     12) PC3< -2.289131 6   0 1 (0 1 0 0 0 0 0 0) *
     13) PC3>=-2.289131 73  42 0 (0.42 0.14 0.014 0.041 0.041 0.19 0.15 0)
       26) PC2>=-1.086724 51  23 0 (0.55 0.16 0.02 0.02 0 0.078 0.18 0)
         52) PC2< -0.6030984 13   1 0 (0.92 0.077 0 0 0 0 0 0)
          104) PC8>=-0.3585423 12   0 0 (1 0 0 0 0 0 0 0) *
          105) PC8< -0.3585423 1   0 1 (0 1 0 0 0 0 0 0) *
         53) PC2>=-0.6030984 38  22 0 (0.42 0.18 0.026 0.026 0 0.11 0.24 0)
          106) PC2< -0.001612419 6   1 1 (0 0.83 0 0 0 0 0.17 0)
            212) PC6>=-1.739485 5   0 1 (0 1 0 0 0 0 0 0) *
            213) PC6< -1.739485 1   0 6 (0 0 0 0 0 0 0 1 0) *
          107) PC2>=-0.001612419 32  16 0 (0.5 0.062 0.031 0.031 0 0.12 0.25 0)
            214) PC7>=0.2887437 11   7 0 (0.36 0.18 0.091 0 0 0.36 0 0)
              428) PC5< 0.4880915 7   3 0 (0.57 0.29 0.14 0 0 0 0 0)
                856) PC4< -1.138325 4   0 0 (1 0 0 0 0 0 0 0) *
                857) PC4>=-1.138325 3   1 1 (0 0.67 0.33 0 0 0 0 0)
                 1714) PC4>=-0.3920356 2   0 1 (0 1 0 0 0 0 0 0) *
                 1715) PC4< -0.3920356 1   0 2 (0 0 1 0 0 0 0 0) *
              429) PC5>=0.4880915 4   0 5 (0 0 0 0 0 1 0 0) *
                                    ⋮
```

Figure 14 bagging tree classification: decision tree

fitact \ fitpred	0	1	2	3	5	6
0	5	3	0	0	1	0
1	0	9	2	0	0	0
2	0	1	4	0	2	0
3	1	1	0	0	0	0
5	1	2	0	1	3	1
6	0	0	0	1	0	0

Figure 15 bagging tree classification: confusion matrix

5.3. C50 classification

The next model that was used is C50 decision trees

Boosting is the process of adding weak learners in such a way that newer learners pick up the slack of older learners. In this way we can (hopefully) incrementally increase the accuracy of the model. Using the C5.0 function, we can increase the number of boosting iterations by changing the trials parameter. Trials=10 was used. Classifiers constructed by C5.0 are evaluated on the training data from which they were generated, and also on a separate file of unseen test cases if available.

The model trained 10 different trees out of which the one with the best predictive power had size equal to 22 and 38 errors (24.4%). Size is the number of non-empty leaves on the tree and Errors shows the number and percentage of cases misclassified. The tree, with 22 leaves, misclassifies 38 of the 156 (train set) given cases, an error rate of 24.4%. While on the training set the accuracy of this model equals to 75%, when it comes to the test set, its accuracy drops to 61%.The discrepancy arises because parts of a case split as a result of unknown attribute values can be misclassified and yet, when the votes from all the parts are aggregated, the correct class can still be chosen.

When there are no more than twenty classes, the model output also contains the performance on the training cases, further analyzed in a confusion matrix that pinpoints the kinds of errors made. If the model was perfect, figure's 18 confusion matrix would have zeros or nulls to all cells except from the diagonal. Furthermore, the output of this model describes how the individual attributes contribute to the classifier (Figure 16). On the right part of figure 16 we can see that the first 3 principal components contribute 100% to the classifier while the other components contribute less. This tree model was too complex to be plotted and therefore it was described as text in figure 17. Figure 17 represents that the different outputs of the conditions made on the principal components lead to different tree branches. Each branch eventually ends with a leaf which contains the name of the choses class or in this case religion. For example if PC7 is less or equal to -1.31, then the predicted class is class number 2.

C50 Trees Classification:

- **Accuracy 61%**
- **Variables actually used in tree construction: PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8 PC9**

									Attribute usage:
(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	<-classified as	100.00% PC1
									100.00% PC5
25	5				1			(a): class 0	100.00% PC7
1	47	1						(b): class 1	90.38% PC2
1	1	27						(c): class 2	85.90% PC8
	1	2	2			1		(d): class 3	80.77% PC9
		1		3				(e): class 4	72.44% PC3
1		1			17			(f): class 5	61.54% PC4
2	4	1				7		(g): class 6	47.44% PC6
	1	1					2	(h): class 7	

Figure 16 C50 decision trees model summary

```
Decision tree:

PC7 <= -1.312195: 2 (10/3)
PC7 > -1.312195:
:...PC1 > 1.216675: 1 (29/6)
    PC1 <= 1.216675:
    :...PC7 > 0.9172115:
        :...PC4 > -0.1529604: 2 (10)
        :   PC4 <= -0.1529604:
        :   :...PC5 <= 0.6638305: 0 (9/2)
        :       PC5 > 0.6638305: 5 (3/1)
        PC7 <= 0.9172115:
        :...PC5 <= -0.08902029:
            :...PC5 <= -2.017972: 5 (3/1)
            :   PC5 > -2.017972:
            :   :...PC7 <= -0.1724993:
            :       :...PC3 <= -0.2732337:
            :       :   :...PC9 <= -0.2446177: 6 (3/1)
            :       :   :   PC9 > -0.2446177: 0 (8/1)
            :       :   PC3 > -0.2732337:
            :       :   :...PC3 <= 0.1261605: 6 (5/1)
            :       :       PC3 > 0.1261605: 3 (3/1)
            :       PC7 > -0.1724993:
            :       :...PC1 <= -0.3323838: 0 (9/4)
            :           PC1 > -0.3323838:
            :           :...PC8 <= -1.750824: 6 (2)
            :               PC8 > -1.750824: 1 (14/4)
            PC5 > -0.08902029:
            :...PC3 > 0.2608237:
                :...PC2 <= -1.68973: 1 (2)
                :   PC2 > -1.68973: 2 (12/4)
                PC3 <= 0.2608237:
                :...PC1 > 0.3328322:
                    :...PC8 <= 0.7893205: 0 (6/1)
                    :   PC8 > 0.7893205: 1 (3/1)
                    PC1 <= 0.3328322:
                    :...PC9 > 0.9324919: 0 (4/2)
                        PC9 <= 0.9324919:
                        :...PC1 <= -1.986462: 5 (4/1)
                            PC1 > -1.986462:
                            :...PC7 > 0.3669907: 5 (6/1)
                                PC7 <= 0.3669907:
                                :...PC1 <= -0.5955708: 1 (5/1)
                                    PC1 > -0.5955708: 5 (6/2)
```

Figure 17 C50 decision trees classification: decision tree

C5act \ C5pred	0	1	2	3	4	5	6	7
0	6	3	0	0	0	0	0	0
1	0	9	2	0	0	0	0	0
2	0	1	5	0	0	1	0	0
3	1	1	0	0	0	0	0	0
5	1	3	0	1	0	3	0	0
6	0	0	0	1	0	0	0	0

Figure 18 C50 decision trees classification: confusion matrix

5.4. J48 classification

The final classifier that will be presented is J48. Decision tree J48 is the implementation of algorithm ID3 (Iterative Dichotomiser 3) developed by the WEKA project team. R includes this into package RWeka. The resulted tree was too complex and only a part of it was represented in figure 19. Moreover, in figure 20 we can see how J48 has taken a subset of the data set as a training set and after that he has applied the resulting decision tree to the rest of dataset's objects. The confusion matrix of figure 21 represents the classification made by the model. The model's accuracy on the training set equals to 79.5% which is calculated as the sum of the confusion matrix diagonal divided by the total number of classified objects. When we evaluate the classifier using the test set, its accuracy drops to 58%.

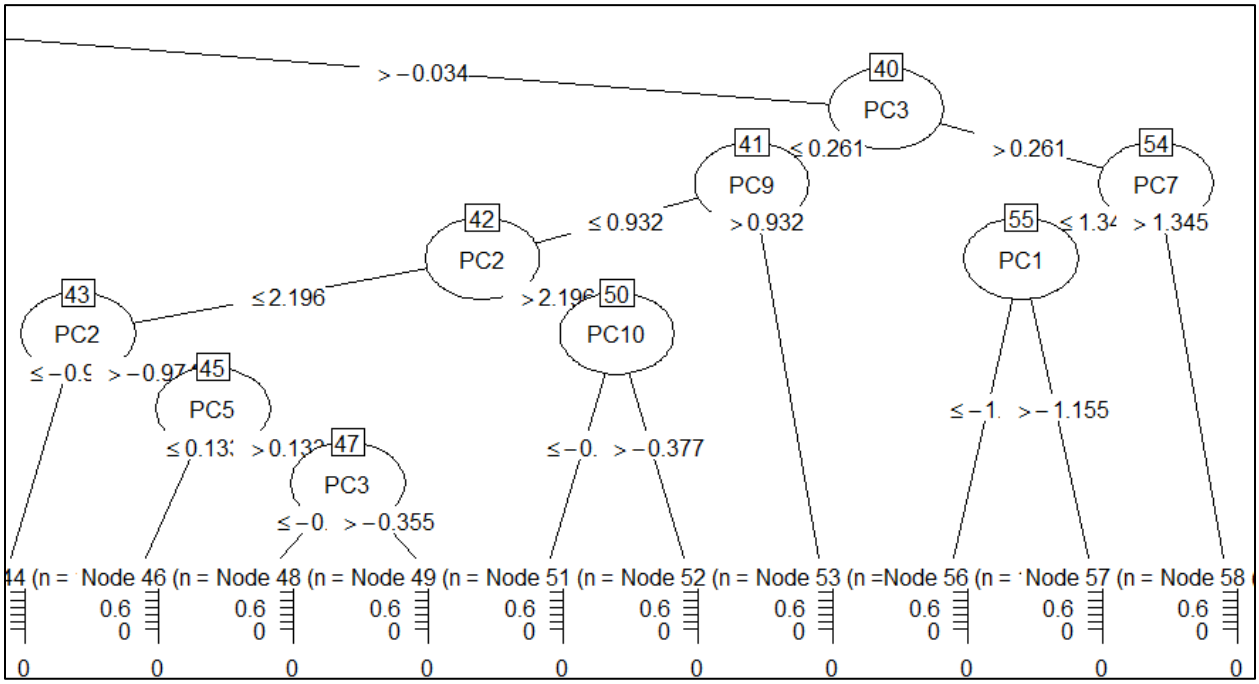


Figure 19 J48 tree classification: decision tree

```
=== Summary ===

Correctly Classified Instances         124               79.4872 %
Incorrectly Classified Instances        32               20.5128 %
Kappa statistic                          0.7385
Mean absolute error                      0.075
Root mean squared error                  0.1937
Relative absolute error                 37.2622 %
Root relative squared error             61.1723 %
Total Number of Instances              156

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h   <-- classified as
 24  5  0  0  0  2  0  0 |  a = 0
  0 45  2  0  0  1  0  1 |  b = 1
  1  3 25  0  0  0  0  0 |  c = 2
  0  1  1  3  0  0  1  0 |  d = 3
  2  0  0  1  0  1  0  0 |  e = 4
  1  2  1  0  0 15  0  0 |  f = 5
  0  3  1  0  0  0 10  0 |  g = 6
  0  1  0  0  0  1  0  2 |  h = 7
```

Figure 20 J48 decision tree model summary

```
      J48pred
J48act 0 1 2 3 4 5 6 7
     0 5 2 0 0 0 1 1 0
     1 1 8 0 0 0 0 1 1
     2 0 1 5 0 0 1 0 0
     3 1 0 0 1 0 0 0 0
     5 1 4 0 0 0 3 0 0
     6 0 0 1 0 0 0 0 0
```

Figure 21 J48 decision tree classification: confusion matrix

5.5. Classification evaluation

Concluding, the top four classifiers were presented out of which "rpart" model performed significantly better than the others, achieving accuracy of prediction on the test set equal to 84%. Figure 22 summarizes the number of principal components that each one of the model used, as well as the accuracy of the model in predicting the religions on the test set.

Classifier	Number of PC's used	Accuracy
rpart	5	84%
bagging cart	9	55%
C5.0	9	61%
J48	10	58%

Figure 22 Summary of classification model results

6. Clustering models

This part of the project, will focus on clustering models. Clustering models focus on identifying groups of similar records, or flags in our case, and labeling the records according to the group to which they belong. This is done without the benefit of prior knowledge about the groups and their characteristics. This is what distinguishes clustering models from the other machine-learning techniques; there is no predefined output or target field for the model to predict. These models are often referred to as unsupervised learning models, since there is no external standard by which to judge the model's classification performance. There are no *right* or *wrong* answers for these models. Their value is determined by their ability to capture interesting groupings in the data and provide useful descriptions of those groupings. Clustering methods are based on measuring distances between records and between clusters. Records are assigned to clusters in a way that tends to minimize the distance between records belonging to the same cluster.

For the purposes of this project, 8 clusters, as many as the religion values, were arbitrarily defined. In this part of the project four different models will be trained to form 8 clusters, by evaluating just the flag characteristics of the countries. We will then evaluate whether the formed clusters are somehow related to the religions the respective countries.

6.1. Hierarchical clustering

The first clustering algorithm was hierarchical clustering and for this hclust method was used. Hclust requires us to provide the data in the form of a distance matrix and therefore method dist was also used. The first step was calculating a distance matrix. For a data set with n observations, the distance matrix will have n rows and n columns; the (i,j)th element of the distance matrix will be the difference between observation i and observation j. The default Euclidean distance function was used to calculate the distance matrice in R. The dist function, is included in every version of R. We used the default method of hclust, which is to update the distance matrix using what R calls "complete" linkage. Using this method, when a cluster is formed, its distance to other objects is computed as the maximum distance between any object in the cluster and the other object. The main graphical tool for looking at a hierarchical cluster solution is known as a dendrogram. This is a tree-like display that lists the objects which are clustered along the x-axis, and the distance at which the cluster was formed along the y-axis. Figure 23 represents the dendrogram of the hierarchical cluster solution. Red bordering identifies the 8 clusters which we are interested to further investigate. From the dendrogram we can see that the optimal number of flag clusters is 3 or 4, however for the purposes of this project we will identify 8 clusters. Figure 24 represents the output of the clPairs function. This function creates a scatter plot for each pair of variables in given data. Observations in different clusters are represented by different colors and symbols.

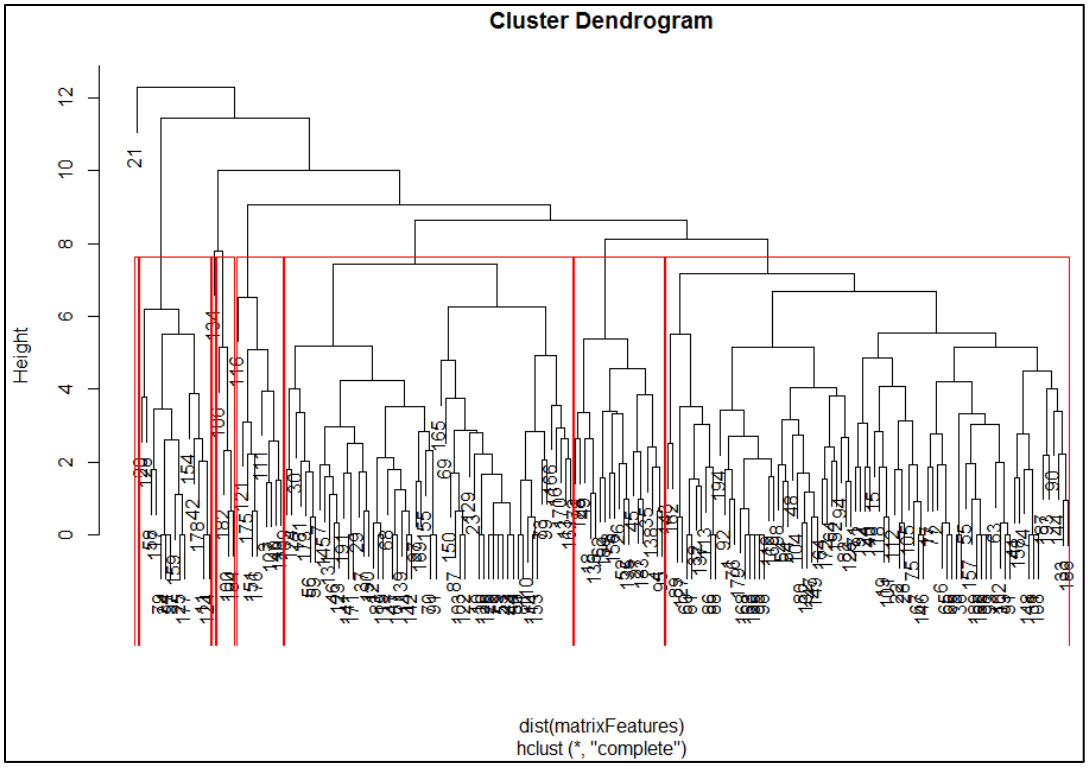


Figure 23 Hierarchical clustering: dendrogram

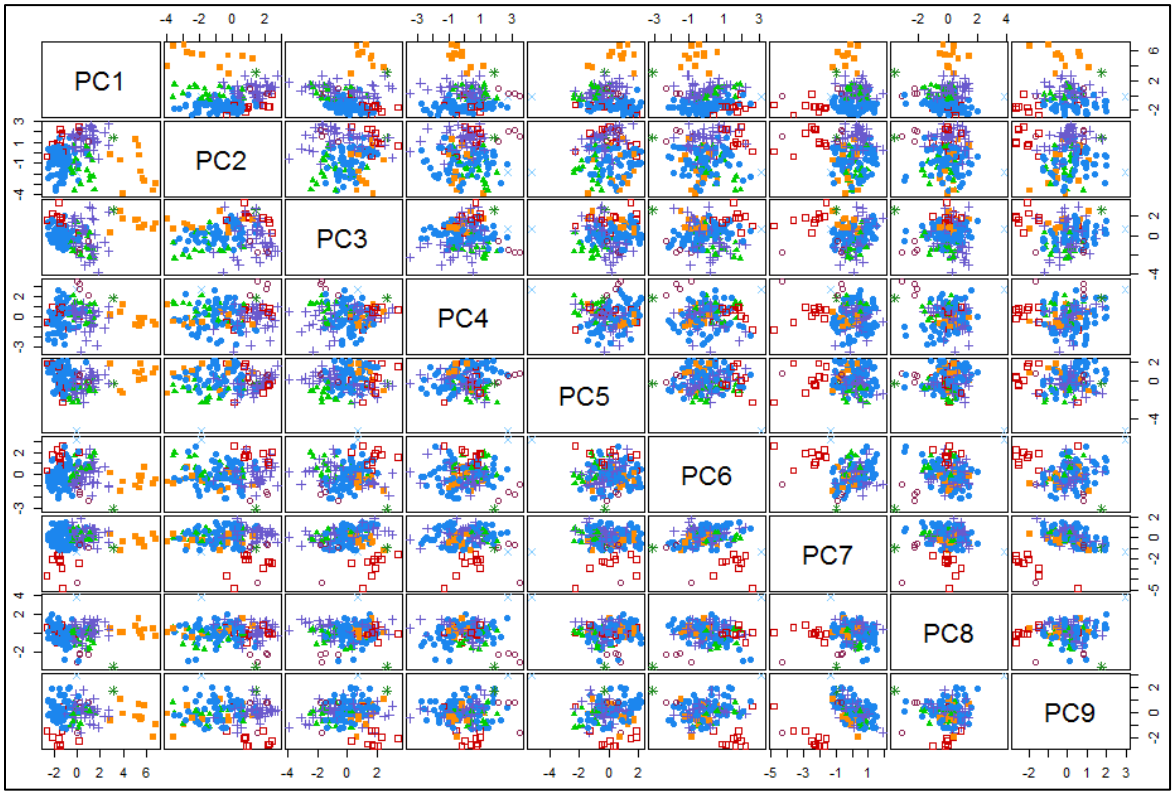


Figure 24 Hierarchical clustering: cluster visualization

6.2. PAM clustering

Unlike the hierarchical clustering methods, techniques partitioning around medoids, available through the pam function in the cluster library, require that we specify the number of clusters that will be formed in advance. Pam offers some additional diagnostic information about a clustering solution, and provides a nice example of an alternative technique to hierarchical clustering. To use pam, you must first load the cluster library. We can pass pam a data frame or a distance matrix; since we've already formed the distance matrix, we'll use that. Pam also needs the number of clusters we wish to form. The 8 cluster solution produced by pam are presented below in figure 25. Since the first 3 components encapsulate most of the dataset's information, we will focus on them. First of all, let's see if the pam solution agrees with the hclust solution. Since pam only looks at one cluster solution at a time, we don't need to use the cutree function as we did with hclust; the cluster memberships are stored in the clustering component of the pam object. Figure 26 represents the table which was created to compare the results of the hclust and pam solutions. Unfortunately, the results of pam and hierarchical clustering do not agree. The row numbers of figure 26 represent the hierarchical clustering solutions while the column numbers represent the pam solutions. Hierarchical clustering seems to perform poorly in this case, because it placed 155 out of 194 flags in the first cluster (second row of figure 26). On the contrary pam clustering placed 21 flags in the first cluster, 35 in the second, 11 in the third, 27 in the fourth, 39 in the fifth, 26 in the sixth, 15 in the seventh and 20 in the last cluster.

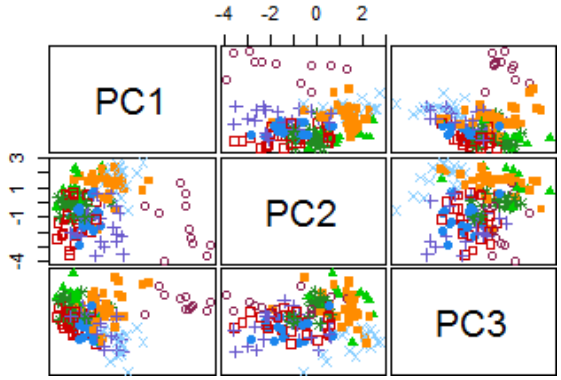


Figure 25 Pam clustering: cluster visualization (zoomed in first 3 PCAs)

6.3. Clustering evaluation

groups.8	1	2	3	4	5	6	7	8
1	21	35	0	20	34	25	0	20
2	0	0	9	0	0	0	0	0
3	0	0	0	0	0	0	15	0
4	0	0	0	7	0	1	0	0
5	0	0	0	0	1	0	0	0
6	0	0	1	0	3	0	0	0
7	0	0	1	0	0	0	0	0
8	0	0	0	0	1	0	0	0

Figure 26 Compare pam and hclust results

Another feature available with pam is a plot known as a silhouette plot. First, a measure is calculated for each observation to see how well it fits into the cluster that it's been assigned to. This is done by comparing how close the object is to other objects in its own cluster with how close it is to objects in other

clusters. Values near one mean that the observation is well placed in its cluster; values near 0 mean that it's likely that an observation might really belong in some other cluster. Within each cluster, the value for this measure is displayed from smallest to largest. If the silhouette plot shows values close to one for each observation, the fit was good; if there are many observations closer to zero, it's an indication that the fit was not good. The silhouette plot is very useful in locating groups in a cluster analysis that may not be doing a good job; in turn this information can be used to help select the proper number of clusters. For the current example, figure 27 represents the silhouette plot for the 8 cluster pam solution. The plot indicates that there is a poor structure to the clusters, with most observations seeming not to belong to the cluster that they're in. There is a summary measure at the bottom of figure 27labeled "Average Silhouette Width" which equals to 0.23. An "Average Silhouette Width" lower than 0.25 indicated that no substantial structure has been found. A successful clustering model would have "Average Silhouette Width" greater than 0.7 which denotes that a strong cluster structure.

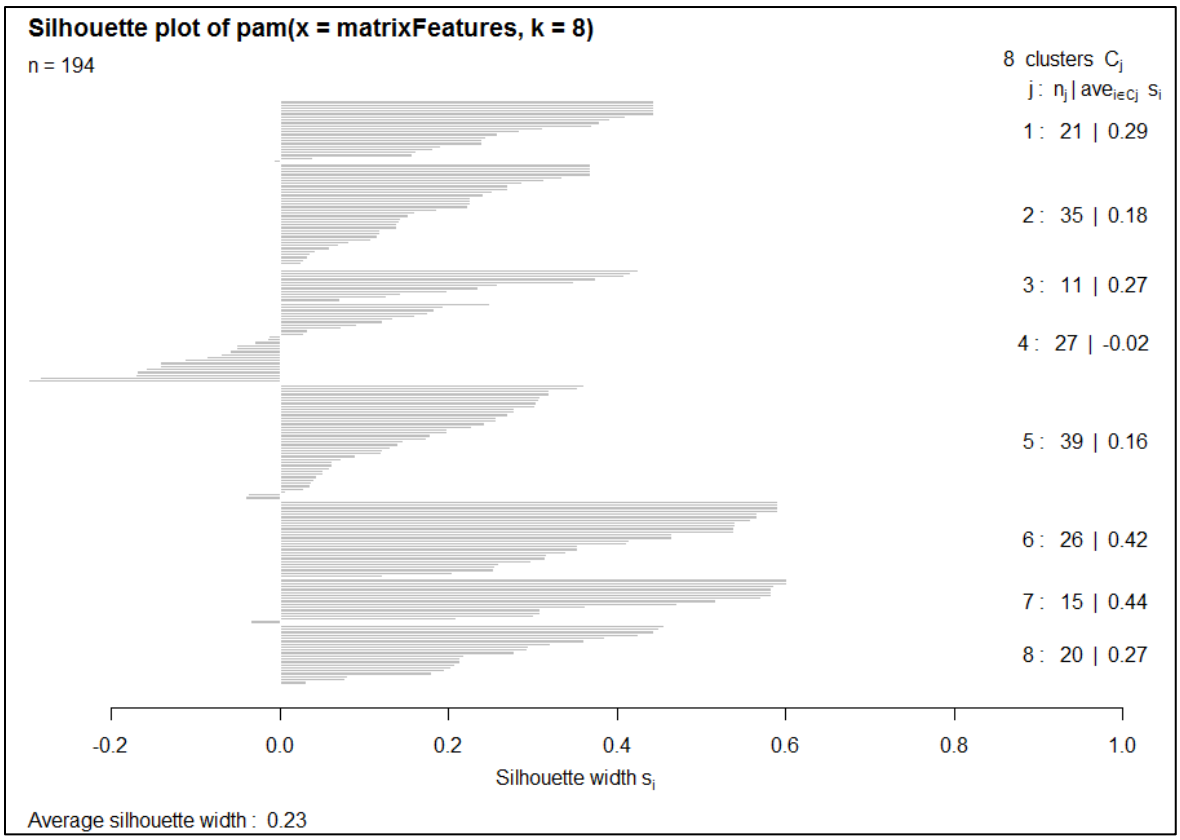


Figure 27 Silhouette plot: Pam clustering model

Figure 28 represents the silhouette plot for the hierarchical clustering model. It has an "Average Silhouette Width" if 0.21, which indicates poorer clustering than the one of pam's model because once again no substantial structure has been found. Moreover, from this plot it is obvious that during hierarchical clustering most of the flags where placed in the first cluster.

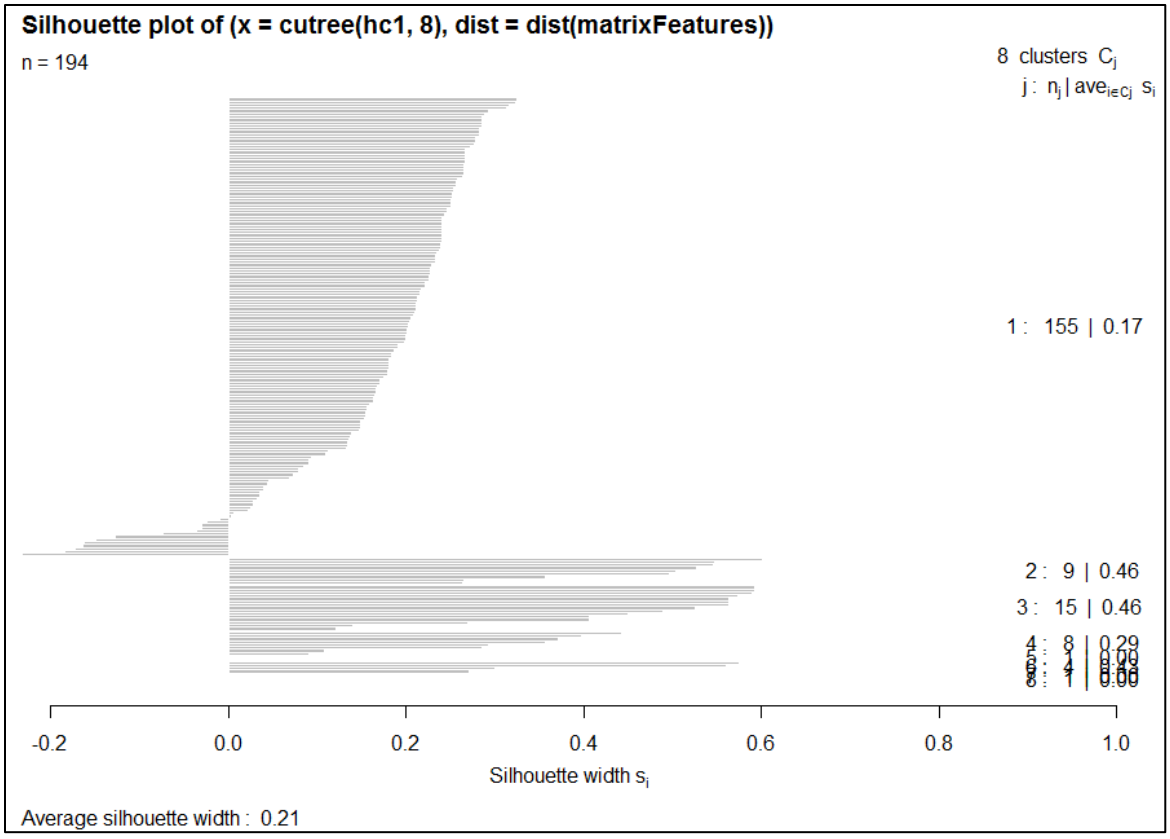


Figure 29 Silhouette plot: Hierarchical clustering model

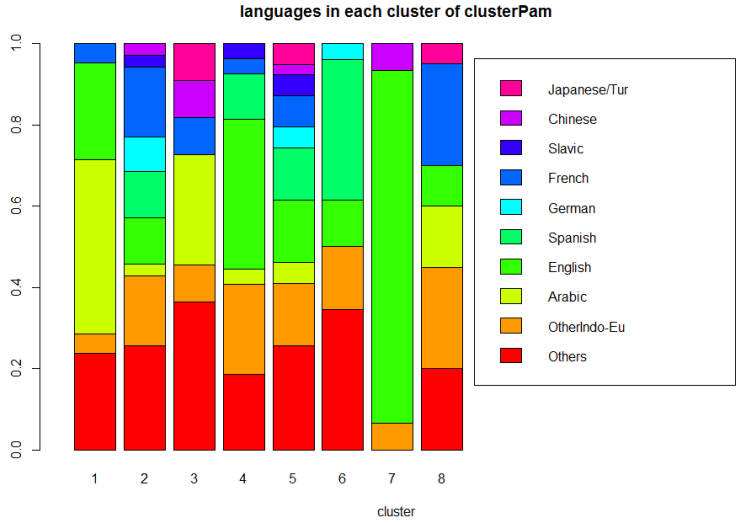


Figure 28 Visualization of pam clustering characteristics: Language

6.4. Cluster analysis

Since "pam" clustering model performed better than hierarchical clustering we will now visualize the characteristics of the clusters formed by this model.

Figure 29 represents the languages spoken in the countries of each cluster. More specifically, the y-axis describes the percentage of each language's frequency within the clusters. For example, in figure 29, the most common language of the first cluster is Arabic.

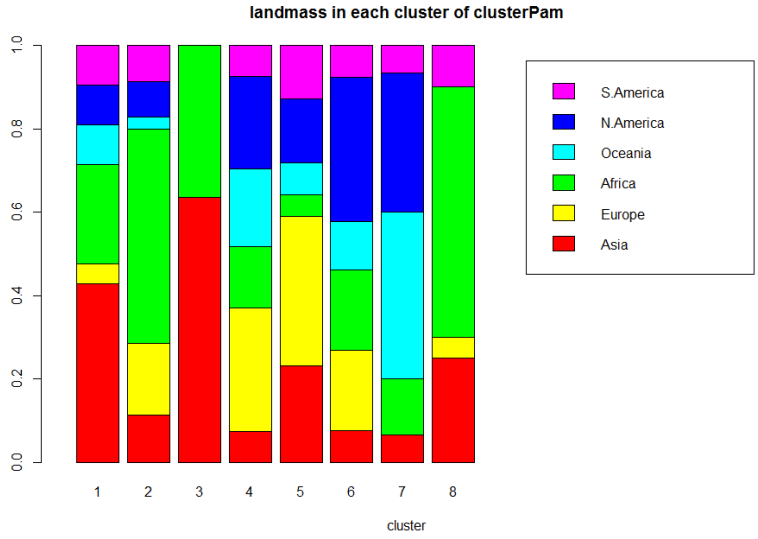


Figure 30 Visualization of pam clustering characteristics: Landmass

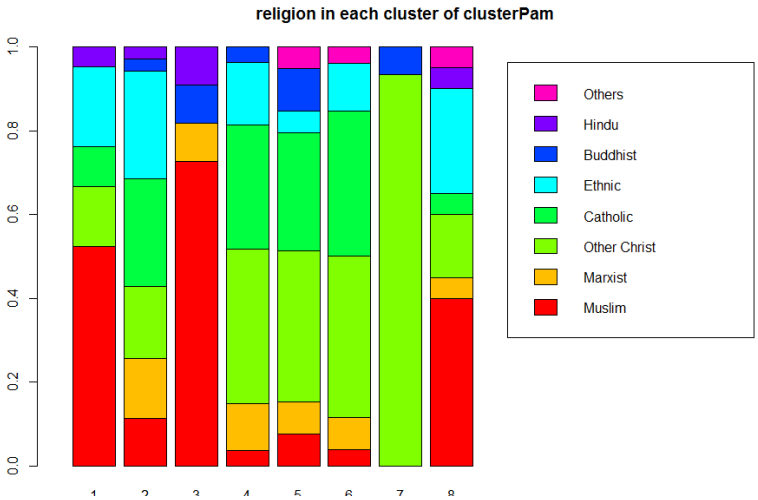


Figure 31 Visualization of pam clustering characteristics: Religion

Figure 30 and 31 visualizes the landmass and the religion of the countries of each cluster. For example, in figure 30, we can see that the most countries of the first cluster are located in Asia and in figure 31, that most of them Muslims.

Concluding, the clustering results up to this point are not satisfactory. The silhouette values are low and the cluster characteristics do not lead to concrete conclusions. At this point we can assume that either the nature of the attributes or the selected number of clusters led to poor clustering results.

6.5. Clustering improvement

We will now investigate the second hypothesis and check whether a different number of clusters will lead to better clustering quality.

For this purpose we will use one more clustering method, mclust. Mclust is a contributed R package for model-based clustering, classification, and density estimation based on finite normal mixture modelling. It provides functions for parameter estimation via the EM algorithm for normal mixture models with a variety of covariance structures, and functions for simulation from these models.

Figure 32 represents the optimal BIC which is used for choosing the number of clusters. In this case the optimal (max) BIC value equals to -4612.868 and occurs when the number of clusters equals to 3 and the shape of the clusters is VVV (ellipsoidal, varying volume, shape, and orientation). Figure 33 is focused on the first three principal components and it visualizes the three clusters. Now that we have a lower number of clusters, the distinction between them is more obvious.

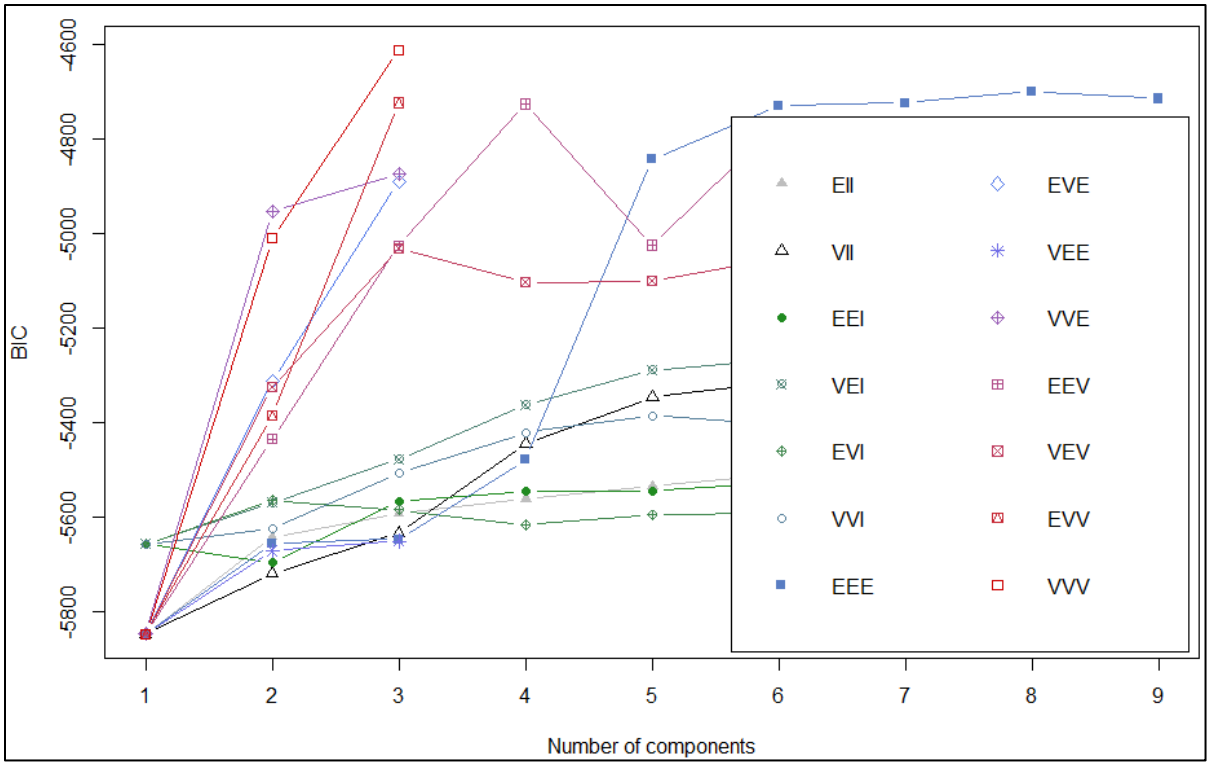


Figure 32 Mclust: Choosing optimal number and shape of clusters

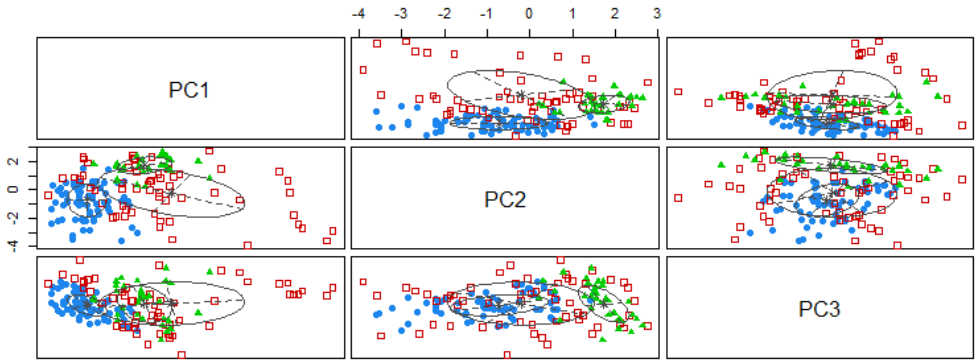


Figure 33 Mclust clustering: cluster visualization (zoomed in first 3 PCAs)

Our next step is to visualize the characteristics of each cluster in order to identify potential patterns.

6.6. Improved cluster analysis

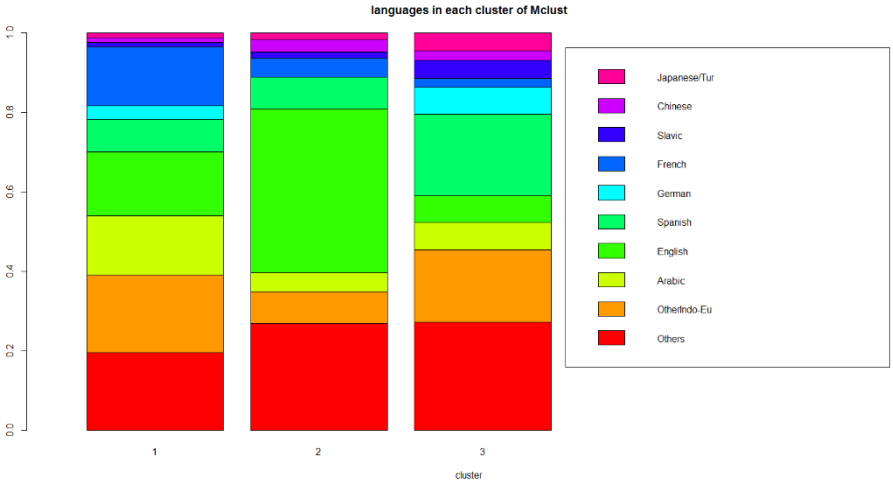


Figure 34 Visualization of mclust clustering characteristics: Language

Figure 34 represents the languages spoken in the countries of each cluster. Figures 35 and 36 represent the landmass and the religion accordingly.

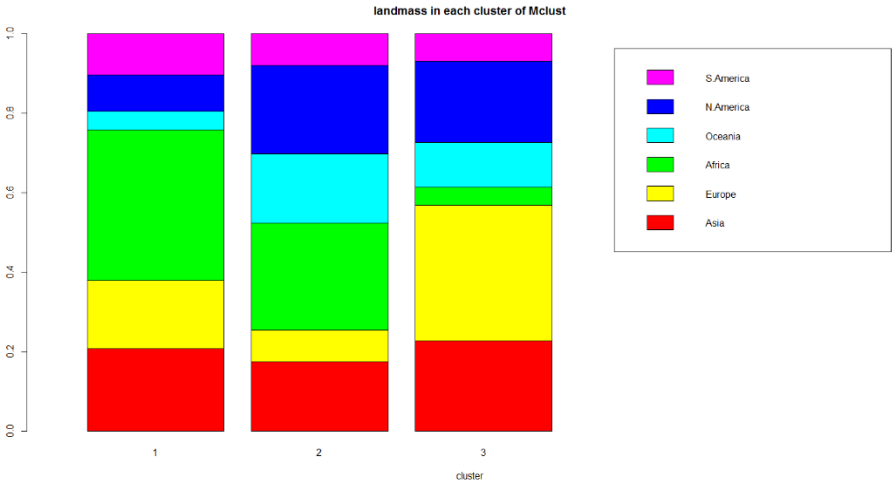


Figure 35 Visualization of mclust clustering characteristics: Landmass

The first cluster contains many African and Asiatic countries which speak English, French and Other languages. The most common religions in this cluster are Muslims, Catholics and Ethnics.

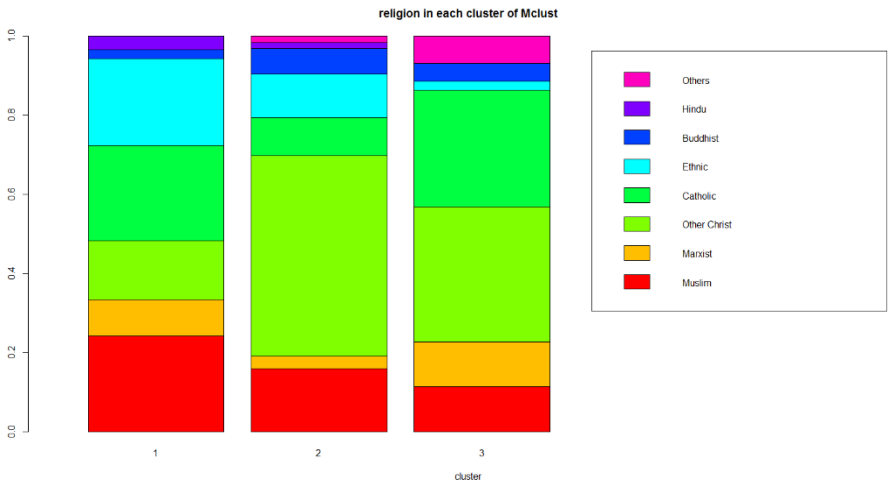


Figure 36 Visualization of mclust clustering characteristics: Religion

The second cluster contains many English speaking countries. The majority of them are Other Christians and many of them are located in North America. Finally the countries of the third cluster have increased percentage of Spanish and Other Indo-European speakers and increased percentage of countries located in Europe. The majority of the third cluster's countries are Catholics or Other Christians.

1st Cluster		2nd Cluster		3rd Cluster
Afghanistan	Kampuchea	Algeria	Parguay	Argentina
Albania	Kenya	American-Samoa	Philippines	Austria
Andorra	Kuwait	Anguilla	Sierra-Leone	Bahrain
Angola	Lebanon	Argentine	Singapore	Botswana
Antigua-Barbuda	Libya	Australia	South-Africa	Chile
Bangladesh	Liechtenstein	Bahamas	St-Helena	Costa-Rica
Belgium	Malawi	Barbados	St-Lucia	Cuba
Benin	Mali	Belize	Swaziland	Czechoslovakia
Bolivia	Mauritius	Bermuda	Sweden	Denmark
Brazil	Mexico	Bhutan	Tanzania	Dominican-Republic
Burkina	Morocco	British-Virgin-Isles	Togo	El-Salvador
Cameroon	Mozambique	Brunei	Tunisia	Faeroes
Canada	Niger	Bulgaria	Turkey	Finland
Cape-Verde-Islands	Nigeria	Burma	Turks-Cocos-Islan	Greenland
Central-African-Republic	North-Yemen	Burundi	Tuvalu	Honduras
Chad	Oman	Cayman-Islands	Uganda	Iceland
China	Papua-New-Guinea	Comorro-Islands	Uruguay	Indonesia
Colombia	Peru	Cook-Islands	USA	Israel
Congo	Portugal	Cyprus	US-Virgin-Isles	Japan
Dominica	Romania	Djibouti		Laos
Ecuador	Rwanda	Falklands-Malvinas		Luxembourg
Egypt	Sao-Tome	Fiji		Malta
Equatorial-Guinea	Saudi-Arabia	Gabon		Marianas
Ethiopia	Senegal	Gibraltar		Micronesia
France	Seychelles	Greece		Monaco
French-Guiana	Soloman-Islands	Guam		Netherlands
French-Polynesia	Spain	Guatemala		Netherlands-Antilles
Gambia	Sri-Lanka	Hong-Kong		Nicaragua
Germany-DDR	St-Kitts-Nevis	Jamaica		North-Korea
Germany-FRG	St-Vincent	Kiribati		Norway
Ghana	Sudan	Lesotho		Poland
Grenada	Surinam	Liberia		Puerto-Rico
Guinea	Syria	Malagasy		Qatar
Guinea-Bissau	UAE	Malaysia		San-Marino
Guyana	USSR	Maldive-Islands		Somalia
Haiti	Vanuatu	Mauritania		South-Korea
Hungary	Vatican-City	Mongolia		South-Yemen
India	Venezuela	Montserrat		Switzerland
Iran	Vietnam	Nauru		Taiwan
Iraq	Yugoslavia	Nepal		Thailand
Ireland	Zaire	New-Zealand		Tonga
Italy	Zambia	Niue		Trinidad-Tobago
Ivory-Coast	Zimbabwe	Pakistan		UK
Jordan		Panama		Western-Samoa

Figure 37 Mclust clustering: countries of each cluster

Figure 37 represents the countries included in each one of the three clusters. This dataset resulted poor clustering. Concluding, in feature improvements we will not use the principal components as input in the clustering models. It is possible that the original flag characteristics might lead to better clusters than what the principal components did.

7. Bibliography

Dataset source

http://archive.ics.uci.edu/ml/datasets/Flags

Attribute selection

- **Caret**

http://machinelearningmastery.com/feature-selection-with-the-caret-r-package/

- **Boruta**

https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/

- **Random Forest**

https://www.r-bloggers.com/variable-importance-plot-and-variable-selection/

Principal component analysis

https://tgmstat.wordpress.com/2013/11/21/introduction-to-principal-component-analysis-pca/

https://tgmstat.wordpress.com/2013/11/28/computing-and-visualizing-pca-in-r/

https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/

Classification

- **Rpart classification**

https://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf

https://www.r-bloggers.com/classification-trees-using-the-rpart-function/

- **Bagging cart classification**

http://plantecology.syr.edu/fridley/bio793/cart.html

- **C50 classification**

http://connor-johnson.com/2014/08/29/decision-trees-in-r-using-the-c50-package/

https://www.rulequest.com/see5-unix.html

- **J48 classification**

http://data-mining.business-intelligence.uoc.edu/home/j48-decision-tree

Clustering

https://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.modeler.help/nodes_clusteringmodels.htm

https://www.stat.berkeley.edu/~spector/s133/Clus.html

- **Hierarchical clustering**

https://www.r-bloggers.com/hierarchical-clustering-in-r-2/

- **Pam clustering**

https://www.stat.berkeley.edu/~spector/s133/Clus.html

- **Mclust clustering**

https://cran.r-project.org/web/packages/mclust/vignettes/mclust.html

- **Cluster characteristics' visualization**

https://www.safaribooksonline.com/library/view/r-machine-learning/9781783987740/ch05.html