CIS 6930 Project I Report

Xi Yang 16216573

Data collection and brief description

The dataset was available on wiki page, and an extra column is required to add as label. In order to avoid labor, I prepared a python script that can extract the data from the wiki page and automatically assign the continent value based on each country.

The prepared dataset has 6 columns as rank, entities, overall life expectancy, male life expectancy, female life expectancy, continent and label. 1-6 are assigned as label for each continent but not used in the experiment. The overall life expectancy, male life expectancy and female life expectancy are the analyzed data and continent is the predictor. The Totally 223 rows are existed in the dataset with no missing value. The dataset can be grouped by continents give 6 groups as Europe (count: 48), Asia(count: 51), Africa (count: 56), North America (count: 34), South America (count: 14) and Oceania (count: 20).

The whole experiment designs

The dataset will be split into training and testing set. The training set will be used to train each of the algorithm to get the predict model. Testing set will be used to evaluate each model based on parameters as accuracy, precision, recall and f-score summarized in the conclusion section. To train each method, 5-fold cross validation will be applied so that the variance of the resulting estimate is reduced based on the fold times. All the train sets are preprocessed with centering and scaling. Also, caret package provides a function to automate the parameter maximization as tunelength. With time and result trade off, the tunelength in my experiments are set to 10 which means that the train method will try 10 different sets of parameters then return the one with the best results.

Training set and Testing set preparation

The requirement of the project description is that the dataset have to be splitted into 80% as training set and 20% as testing set. Since there are six categories and each category has different number of rows, Stratified Random Sampling method is used for creating training and testing data sets. The detail procedure includes group data by labels, count each group number and shuffle each group, divided each group as 80% and 20% data sets and recombine all the data sets with 80% as training and 20% as testing.

The k-nearest neighbors algorithm (knn)

In the knn, the most important parameter is the number of clusters (k). To optimize the k value, I include a train grid in the training method which will check the k value from1 to 25 to find which k will provide the best result. What I found was that in each run the k value is different and fluctuates around the square root of the total row number. Below is a result table from one of the 5 parallel experiments.

All the data on the diagonal line are the true match data with a number equals to 26.

Select result table:

Predict vs data	Africa	Asia	Euro	North America	Oceania	South America
Africa	11	2	0	1	2	0
Asia	1	5	1	1	1	0
Euro	0	2	9	4	1	3
North America	0	2	0	1	0	0
Oceania	0	0	0	0	0	0
South America	0	0	0	0	0	0

The C4.5 algorithm

For C4.5 algorithm, the key parameters are the minimum cases (M) and pruning confidence level (CF). In this example, the M should be 5 since we have 6 categories. To optimize the CF, we either need experience to give a "good" CF or we can utilize the caret package to test several suggested CF and choose the best one. I used the later one to let R to choose the CF for me. From the results, we can tell that the system can provide very reasonable CF value. One of the experiment result tables are presented as below which is close to the one from the knn with true match number also equals to 26.

Select result table:

Predict vs data	Africa	Asia	Euro	North America	Oceania	South America
Africa	11	1	0	1	2	0
Asia	1	4	1	0	1	1
Euro	0	2	8	3	1	0
North America	0	4	0	1	0	0
Oceania	0	0	0	2	0	0
South America	0	0	1	0	0	2

The RIPPER algorithm

Similar as the C4.5 algorithm, RIPPER also have the tree grow phase and prune phase, since it is hard to tuning the parameters by hand. Again, I let the R to handle the parameters optimization by set tunelength to 10 which will test 10 set of different parameters.

One of the result tables are presented below. We can extract from the table that the true match value is 23.

Select result table:

Predict vs data	Africa	Asia	Euro	North America	Oceania	South America
Africa	12	5	2	5	2	3
Asia	0	4	1	0	1	0
Euro	0	2	7	2	1	0
North America	0	0	0	0	0	0
Oceania	0	0	0	0	0	0
South America	0	0	0	0	0	0

The SVM algorithm

For the SVM, there are several sub-methods including linear, polynomial, radial and exponential. There methods are used for specific cases. The radial method utilizes Gaussian radial basis function kernel which is suitable for this data set since we need to group data into different groups (clustering by circles). In "svmradial" method, there are two import parameters as cost and gamma. The cost parameter trades off mis-classification of training examples against simplicity of the decision surface. Low value cost tends to make decision surface smooth, while a high cost tries all training examples correctly by giving the model freedom to select more samples as support vectors. In my experiment cost set to 1 is good enough. Gamma parameter defines how far the influence of a single training example reaches, with low values connote far and high values connote the neighborhood. In my experiment, the optimized gamma is 0.333333. All the parameters are decided also by auto-selection from the caret package which is consistent with 1 and 0.3333.

One of the result tables are presented below. We can extract from the table that the true match value is 22.

Select result table:

Predict vs data	Africa	Asia	Euro	North America	Oceania	South America
Africa	10	0	2	0	0	0
Asia	1	5	1	2	2	2
Euro	0	3	6	4	0	0
North America	1	3	1	1	2	1
Oceania	0	0	0	0	0	0
South America	0	0	0	0	0	0

Conclusion

Output:

==========

Classification Results

==========

method name: KNN; averaged accuracy: 0.53; accuracy standard deviation: 0.051

averaged recall: 0.402; averaged percision: 0.633; averaged f-score: 0.492

==========

method name: C4.5; averaged accuracy: 0.49; accuracy standard deviation: 0.037

averaged recall: 0.350; averaged percision: 0.611; averaged f-score: 0.445

==========

method name: RIPPER; averaged accuracy: 0.43; accuracy standard deviation: 0.041

averaged recall: 0.250; averaged percision: 0.644; averaged f-score: 0.360

==========

method name: SVM; averaged accuracy: 0.53; accuracy standard deviation: 0.035

averaged recall: 0.392; averaged percision: 0.657; averaged f-score: 0.491

==========

From the output we can conclude that the accuracy is about 0.5. At 95% level there is no difference among methods KNN, C4.5 and SVM (using **Tuckey HSD Test**). However, the result from RIPPER is significantly lower than other methods. One of the problem I found is that even using the Stratified Sampling Method in the creating training and testing data set stage, there is no significant improvement compared to directly shuffle the data set and separate it into training set and testing set without balance the distribution of each category. One potential explanations for this phenomenon, low accuracy and recall is that the data difference between Europe and North America and between Asia, south America and Oceania are very close, not enough data in group **South America** and the three data fields are not independent which degrade the accuracy of the training process.