Raymond Huang, Gurpreeth Singh
Professor Mane
5/15/2020
CSE 353

Introduction

Countries are designed to represent a group of people and their beliefs. Most countries have a single major religion that most of their people follow. So, can we figure out a country's religion based on its geographical position and/or flag? For this project, we used the flags dataset taken from UC Irvine, to try and predict a country's religion based on the attributes of its geography and its flag. This flag dataset included attributes like continent the country is located on, population of the country, country's language, total land area, flag colors, religion, and more. To see the full set of attributes, you can open the csv file or go to the website in the references. To achieve our goal of predicting the religion, we used a KNN classifier to differentiate between Christian, Islam and Other religions. We then used random forest decision trees to compare results to our KNN classifier.

Raymond Huang, Gurpreeth Singh
Professor Mane

Description

For the KNN classifier, we had to determine which attributes we would choose for our model. We got the best results using landmass and language (this and other results are explained in depth in the results section). We first read the data from the csv file and then split the data into a test set and training set and normalized the data if it was too large like for population and land area. We chose 30% test set size and 70% train set size since that got us the best results for the prediction. We created ranges for the plots based on the data and then created the KNN classifier with k=5. We then calculated the accuracy and printed out the test predictions and actual values for religion to see which test set countries were predicted right and which ones were predicted wrong. We also decided to plot the error rate based on the value of k for our classifier from 1 to 25. Finally, we did a 5-fold cross validation to assess our prediction model.

As stated in the intro, we decided to implement random forest decision trees to compare our results from the KNN classifier. For the random forest decision trees algorithm, we decided to create 100 decision trees using the landmass and languages features from the KNN classifier along with the crescent, crosses, and sun stars features. These additional features were chosen by looking at common religious symbols for each religion across many different countries. Again, we chose a 30% test set size and 70% train set size since that gave us the best results. We printed the accuracy and saved the first tree of the forest as an image file. We did this because the actual plot was too big to show. However, if you do want to see the plot then just add plt.show() at the end of the decision tree python code. You can also view other trees in the forests by changing the index in the tree.plot_tree function.

Algorithm and Implementation

We used a lot of python libraries for both parts:

- Numpy for arrays
- Pandas for reading the csv
- Matplotlob for plotting
- Sklean for algorithms and models

Both the KNN classifier and random forest uses similar algorithms. A general description of what we did is explained in the description section, but here you will see images of specific parts. After picking the attributes and splitting the data into the test and training set, the code below would be used to normalize the data if necessary:

```
# Normalize data if you need to
normalize_scale = StandardScaler().fit(x_train)
x_train = normalize_scale.transform(x_train)
x_test = normalize_scale.transform(x_test)
```

We then used the sklearn libraries to make the KNN classifier and random forests classifier which we would use to make our prediction:

```
# Use KNN classifier to predict religion
knn_class = KNeighborsClassifier(n_neighbors=5, weights='distance')
knn_class.fit(x_train, y_train)
```

```
# Create decision tree classifier
classifier = RandomForestClassifier(n_estimators=100)
classifier.fit(x_train, y_train)
```

For both predictions, we used the following algorithm to get the total correctly predicted values and the accuracy of our model:

```
# Get accuracy of total prediction
y_total = len(y_test)
y_correct = 0
for i in range(0, y_total):
    if y_test[i] == y_prediction[i]:
        y_correct += 1
y_accuracy = y_correct/y_total
```

For measuring the KNN error for the size of K:

```
# Calculate error for KNN from 1 to 25
errors = []
for i in range(1, 25):
    knn_class_i = KNeighborsClassifier(n_neighbors=i, weights='distance')
    knn_class_i.fit(x_train, y_train)
    y_prediction_i = knn_class_i.predict(x_test)
    errors.append(np.mean(y_prediction_i != y_test))
```

Finally for KNN, we preformed the cross validation (we did a 5-fold cross validation since we felt splitting up our data into 5 parts worked best) and got the data from the cross validation using the following code:

```
# Do a 5-fold cross validation
knn_cross_valid = KNeighborsClassifier(n_neighbors=5, weights='distance')
cross_val_score = cross_val_score(knn_cross_valid, x, y, cv=5)
cross_val_mean = np.mean(cross_val_score)
```

For the random forests, we plot the first tree using the following code (to plot another tree change the index to a number in the range 0-99 inclusive since there are 100 trees):

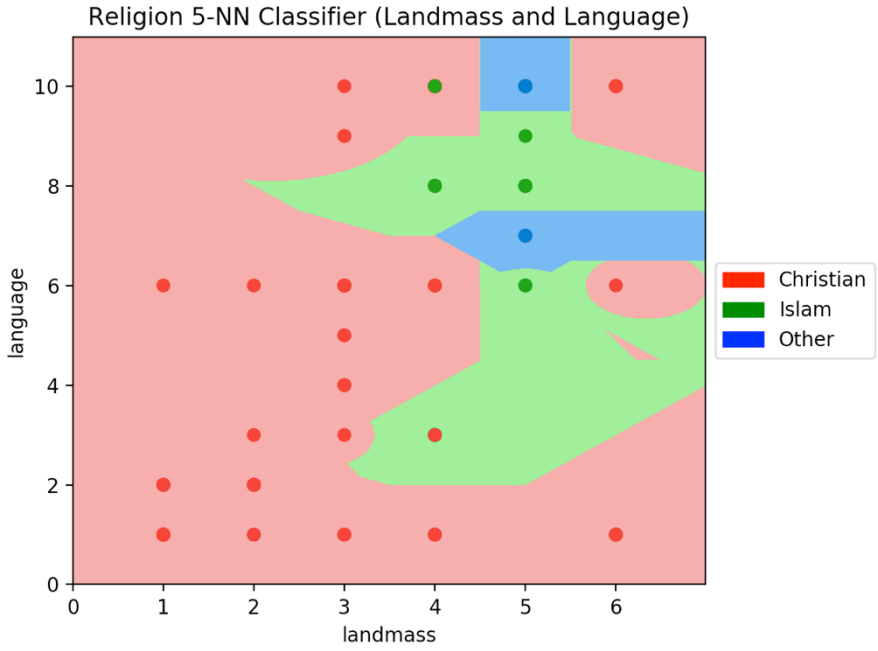
```
# Plot decision tree graph (change estimator index for which tree you want to show)
tree.plot_tree(classifier.estimators_[0], feature_names=features_arr, class_names=classes_arr, filled=True,
               fontsize=8, ax=ax)
```

Raymond Huang, Gurpreeth Singh
Professor Mane

5/15/2020
CSE 353

Results and Conclusions

For our KNN Classifier, we got the best results using ‘landmass’ and ‘language’ as the attributes. The classifier will be a bit different each time it is run, but here is one example:



language: 1=English, 2=Spanish, 3=French, 4=German, 5=Slavic, 6=Other Indo-European (Persian & Indian languages), 7=Chinese, 8=Arabic, 9=Japanese/Turkish/Finnish, 10=Others

landmass: 1=N.America, 2=S.America, 3=Europe, 4=Africa, 4=Asia, 6=Oceania

Below is the test set accuracy for each religion:

	precision
1	0.98
2	0.62
3	0.75

(1 is Christianity, 2 is Islam, 3 is Other)

Here is a part of the results that was printed in the output (10 of the 58):

```
['Vietnam' 3 3]
['Gambia' 2 1]
['Czechoslovakia' 1 1]
['Vatican-City' 1 1]
['Bahamas' 1 1]
['Sri-Lanka' 3 3]
['Jamaica' 1 1]
['South-Yemen' 2 2]
['Nicaragua' 1 1]
['Bahrain' 2 2]
```

It is formatted as ‘country’ ‘actual religion’ ‘predicted religion’. As you can see, only Gambia was predicted incorrectly as a Christian country when it is a Muslim one. Gambia is in Africa and we did predict Africa would be difficult. Asia was mostly well predicted, but it did predict Indonesia wrong too:

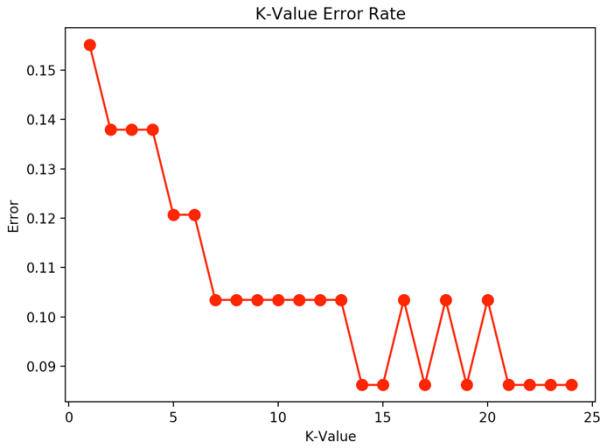
```
['Bolivia' 1 1]
['Indonesia' 2 1]
['Iran' 2 2]
```

Indonesia is a Muslim country but was predicted as a Christian one. Most Christian countries were predicted correctly, it had a 98% success rate.

The reason Christianity is very easy to predict makes sense when you start thinking about it. All of the Americas, Europe, and Oceana is Christian, so those countries are easy to predict. That leaves us with Africa and Asia, which is where most of the errors would be. However, for Africa and the Middle East, all the Arabic speaking countries are Islamic countries. Therefore, the hardest parts in the world are the non-Arabic speaking countries in Africa and the rest of Asia. Asia has countries in all 3 categories right next to each other and that is why it is very difficult to predict. Examples of this is South Asia and Southeast Asia. Below are the results of the KNN Classifier and the 5-fold cross validation

```
The KNN Classifier predicted 51 out of 58 correctly, meaning the accuracy is: 0.8793103448275862

5-fold Cross Validation Accuracy: [0.84615385 0.79487179 0.87179487 0.97368421 0.84210526]
Cross Validation Mean:{} 0.8657219973009447
```

The 5-fold cross validation is vey close to the KNN Classifier and is near 90% accuracy, meaning that our model is good. As mentioned before, we plotted the K-value error rate:



We chose 5 as our K for the models, but as you can see, the higher the k value is the more accurate the prediction model would be. However, when we chose a higher value for K, there is a greater chance that the cross-validation results and prediction model results are greater apart.

When we used 10 for K, our prediction model had an accuracy of 95% but the cross-validation results were still in the 80s:

```
The KNN Classifier predicted 55 out of 58 correctly, meaning the accuracy is: 0.9482758620689655

5-fold Cross Validation Accuracy: [0.84615385 0.79487179 0.87179487 0.97368421 0.84210526]
Cross Validation Mean:{} 0.8657219973009447
```

Therefore using 5 as K would be better since it is closer to the cross-validation results and thus more accurate.

Now what about predicting the religion based on a country's flag? The best predictions we could do were in the 60% accuracy. Below are the accuracies when we based our prediction on the attributes 'sun stars' and 'crosses':

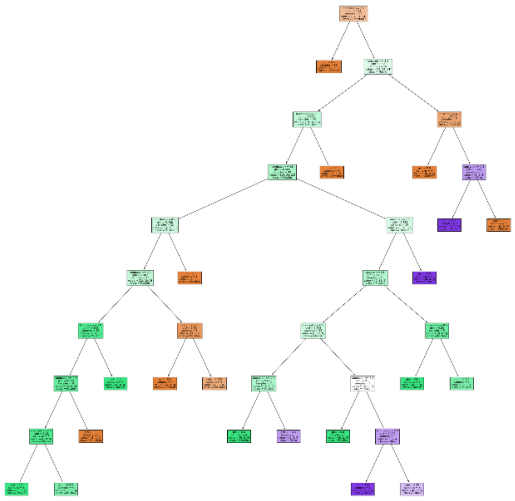
```
The KNN Classifier predicted 36 out of 58 correctly, meaning the accuracy is: 0.6206896551724138

5-fold Cross Validation Accuracy: [0.64102564 0.64102564 0.64102564 0.68421053 0.68421053]
Cross Validation Mean:{} 0.6582995951417004
```

Turns out flags of different religions have many similar features. Overall, predicting a country's religion based on geography is achievable, but trying to predict it based on a country's flag is much more difficult.

We could now compare our KNN classifier results with the random forest decision trees classifier. Looking at the results of the random forests algorithm, the accuracy averages around 85%:

```
Accuracy of decision tree:  0.8793103448275862
```

With this accuracy, we can see that most of the test cases were classified correctly and was around the same accuracy for our ‘landmass’ and ‘language’ KNN classifier, meaning that the features do a good job of predicting religion. Below is one of many decision trees in the forest:



To view the tree, open the image file ‘religion_decision_tree.png’.

Raymond Huang, Gurpreeth Singh
Professor Mane

5/15/2020
CSE 353

Team Contribution

Raymond wrote the code and project report sections for random forests. Gurpreeth wrote the code and project report sections for the KNN Classifier.

References and Citations

UCI Flags Dataset: https://archive.ics.uci.edu/ml/datasets/Flags

For sklearn code stuff: https://scikit-learn.org/stable/