k-Nearest Neighbours Lab

`monthly-data-labelled.csv` : A data set with labelled cities and features
`monthly-data-unlabelled.csv` : A data set with missing cities

Tasks

Weather Data

When recording the 2016 data, somebody forgot to record which city the observations came from! Whatever will we do? It is your job to figure out what cities should be in the unlabelled dataset. To accomplish this you are going to build a k-Nearest Neighbours Classifier and hope this works.

I will try to lay out step by step tasks. Do the following in a workbook

- Read in the CSV, do your usual checks with `.head()` etc.
- Decide what is the response variable and what are the features you will be using.
- Partition into training sets and test sets. You will need to import train_test_split
- Create a kNearestNeighborsClassifier with $k = 5$ to start.
- How did it perform on the test set?

Ok, we should not just use $k = 5$. We should use Cross Validation. Use KFold with 5 folds to decide on the value of k for our nearest neighbour classifier. Also, look at the features, they are not within the same scale. Should we actually build a model with some form of scaling? Think back to the notes

- Make a model with the k you selected using cross validation (and scaling). Remember to fit with the whole training set after you choose k.
- Print the accuracy score for the test set and your model.
- Do a confusion matrix for the model you have chosen. Are any cities having a lot of difficulties?
- Also print the classification report. Try to interpret the information here.

Now we have our chosen model.

- See what predictions the model makes with the unlabelled information.
- Add the prediction to a dataframe and save to a new csv file.

Colour Data

If you get that done, revisit the colour data set from last week. We are going to try a kNearestNeighborsClassifier with the colour data - both with RGB and LAB form. There is no need to normalise the data for kNN here - they are all within the same range anyway.

- Use Cross Validation to find the best k.

- Print the accuracy score for the best model. How does it compare to GaussianNB and LogisticRegression
- Do a confusion matrix for it and your previous GaussianNB model, see if there are any commonalities or where the difference lie.
- Also print a classification report.

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split (X, y)

from sklearn.model_selection import KFold
kf = KFold(n_splits =5, shuffle=true)
for train_index , val_index in kf.split ( X_train ) :
X_c, X_val = X_train [ train_index ] , X_train [ val_index ]
y_c, y_val = y_train [ train_index ] , y_train [ val_index ]
```

Commands may help.