Mourya Meda

Project Report

Ocean Ship Logbooks (1750-1850)

Motivation and Objective

The underlying motivation behind this project is to dissect historical logbooks and extract certain knowledge that is not necessarily written down in history. With the use of modern, data mining techniques, there is a great deal of knowledge and insight to be gathered form these historical logbooks. The main goal of this project is to explore the dataset of the Ocean Ship Logbooks from the years 1750-1850, and generate information such as the ships' routes, the clustering of routes, and whether a certain log and the route is associated with a certain nation.

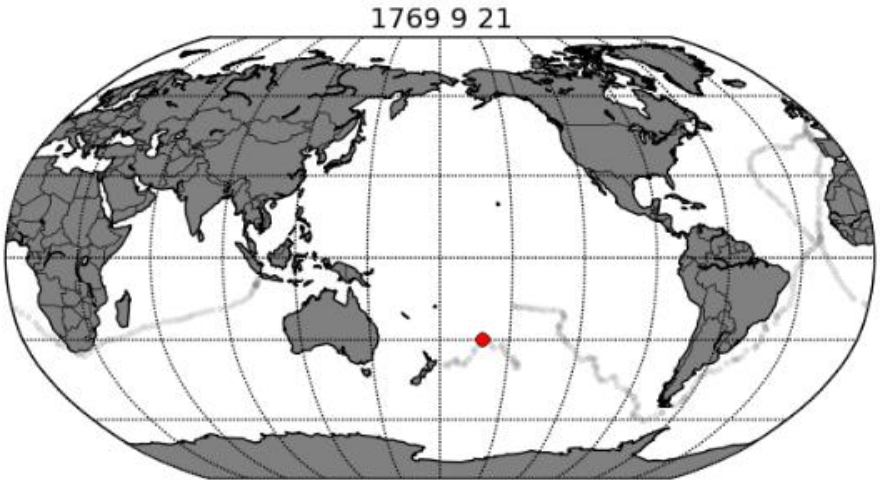
Related Work

There is one significant work submitted by the user kati in which a python script was written to animate Captain Cook's travel. The following submission is in the link below:

https://www.kaggle.com/katacs/d/kaggle/climate-data-from-ocean-ships/captain-cook-s-travels

This method can be adopted and changed. Resulting in an extremely useful depiction of all the routes of the ships that have been taken in each decade. With this we can begin to identify the prevalence of a certain captain or a nation who have dominated the seas at that time. Furthermore, we can easily visualize the dataset that is given in a map.

The result would be a similar animation except with all the ships in each decade:



Approach and Methodology

The basic outline of this project involved three main objectives and the following methodologies:

1. Generated the ship's routes and discover the prevalence of a specific nation. This will take a big influence from the animated Captain Cook's travel program in the related works. However, by adding on all the ships we can aim to identify the dominance of a certain nation in the seas.

To begin with, NumPy is used to hold the dataset and manipulate the data as needed.

The following attributes were focused on for mapping the logbook entries: Latitude, Longitude, and Nation. However, UTC, Year, Month, and the ShipName are all stored as metadata so that the program can have many uses including plotting the entries of only a certain time period or even by name. The entries that are missing the coordinates have been removed obviously.

Basemap was imported from mpl_toolkits which provided with a map to plot the coordinated based on the latitude and longitude.

The following construction prepared the map to plot the coordinates:

```
data = np.column_stack((coord, ship, year, month, utc, nation))

# sets up the base map
m = Basemap(projection='robin', lon_0=0, resolution='c', llcrnrlon=120, urcrnrlon=-30)
m.drawcoastlines()
m.drawcountries()
m.fillcontinents(color='grey')
m.drawmeridians(np.arange(0, 360, 30))
m.drawparallels(np.arange(-90, 90, 30))

nationList = np.matrix([['Spanish', 'yellow'], ['French', 'cyan'], ['Swedish', 'magenta'], ['Dutch', 'red'],
                        ['British', 'blue'], ['Danish', 'green'], ['Hamburg', 'grey']])
```

Each nation took on a certain color so they can be easily identified in the map. Furthermore, **data** is the current dataset after all the filtering.

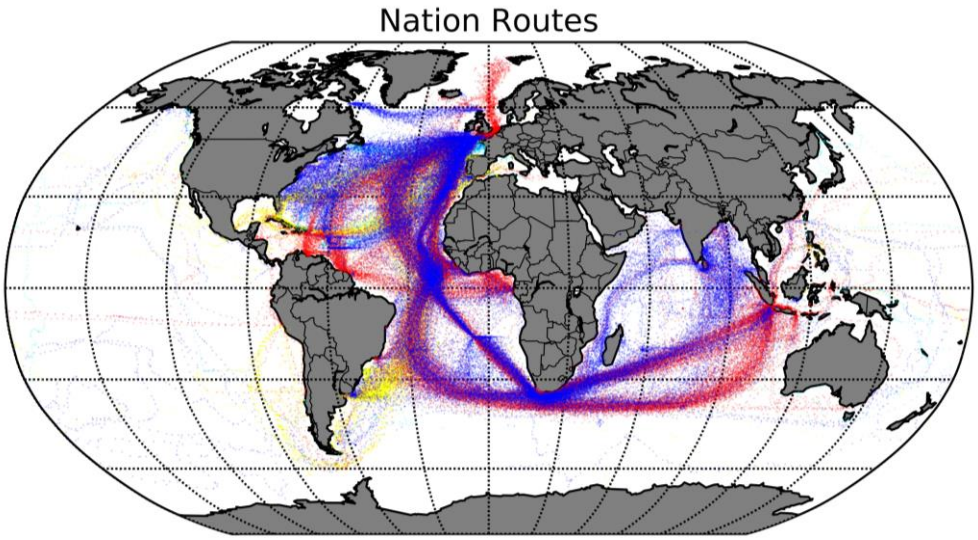
The following strategy is used to plot all the data points:

```
# plot data
for nation in nationList:
    temp = data[data[:, 6] == nation.item(0)]
    #temp = data[data[:, 3] < 1800]

    # sort time
    temp = temp[temp[:, 5].argsort()]

    # draw  the paths on the background
    x, y = m(temp[:, 0], temp[:, 1])
    m.plot(x, y, '.', color=nation.item(1), alpha=0.2, markersize=.5, label=nation.item(0))
```

Note that **#temp = data[data[:, 3] < 1800** comment is where the metadata of the dataset could come in to customize the map in any format needed. The **3** being the index of the year in **data**, and **1800** being the year selected. Further refinements could be made to take in an input when running the program to create a customized map. However, to display all the logbook entries' coordinates with their nationality the following map is created with this program:



You can already see some of the clusters that have formed in certain regions of the world.

2. Clustering the routes to identify relationship. The main goal here is to gather any insights by performing clustering on the routes. The clustering will most likely take k-means algorithm. The data's distance formulas such as Euclidian will be applied. This can give us a good estimation of the clusters of certain nations and their dominance. With the use of Confusion Matrix we can identify why certain ships are identified as a certain nationality.

Once again NumPy is used to hold and manipulate the dataset. Mainly, sklearn library is imported as it holds any preprocessing and classifiers functions.

In clustering the entries, the following feature vector and target variable was set:

Feature vector: latitude, longitude, UTC, year, month
Target variable: Nationality

Deciding the attributes to select to conduct the clustering was tricky. For the present case, only the numerical attributes were selected. Further preprocessing involved removing any entries that had missing attributes. However, it is important to note that latitude and longitude will play a dominant role because it would closely resemble the nations' set routes. So any future ships that take the route will follow very closely to the coordinates.

Before the KNeighbors model is run, the data is first normalized and standardized. Normalization involved replacing nominal features, so that each of them would be in the range

from 0 to 1. As for standardization, it involved data pre-processing, after which each feature has an average 0 and 1 dispersion.

The following model with **k=7** is run:

```
# run k-means
model = KNeighborsClassifier(metric='euclidean', weights='distance', n_neighbors=7)
model.fit(X, y)

expected = y
predicted = model.predict(X)
```

And the output which holds the classification report and confusion matrix is sent to a textfile named **OutputKMeans.txt**.

3. Naive Bayes model to identify a certain log or a route to a specific nation. This is very useful in a future work when if discovered a new entry or a log, then you can run the classification model to identify its nation. The practice from the Classification assignment will be a great help in this section. Furthermore, we can compare the classification with that of the K-means clustered classifications to decide which provided a better outcome. However the main objective is to develop an accurate model for any future findings of logbook entries.

   Naive Bayes model followed similar approach to the KMeans clustering approach with the use of sklearn library and the GaussianNB in which the likelihood of the features is assumed to be Gaussian:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

   I chose this because the latitude and longitude is a continuous value.

Likewise, in classifying the entries, the following feature vector and target variable was set:

Feature vector: latitude, longitude, UTC, year, month
Target variable: Nationality

The data was normalized and standardized as same as for the k-means implementation.

Then the following model is run:

```
# run naive bayes
model = GaussianNB()
model.fit(X, y)
expected = y
predicted = model.predict(X)
```

And the output of the classification report and its confusion matrix written in a textfile named, **OutputNaiveBayes.txt.**

Evaluation

In the mid-eighteenth to nineteenth centuries, navigating the open ocean was an imprecise and often dangerous feat. In order to calculate their daily progress and avoid running/sailing into the unknown, a ship's crew kept a detailed logbook with data on winds, waves, and any remarkable weather.

Handwritten in archived logbooks, these rich datasets were nearly impossible to study until the European Union funded their digitization in 2001. The actual data comes from Climatological Database for the World's Oceans 1750-1850 (CLIWOC), version 1.5 data release.

The primary data file is CLIWOC15.csv. The columns in this table are described on this page (scroll down to the table that starts with "Field abbreviation"). It includes 280,280 observational records of ship locations weather data, and other associated information.

After the following programs have run, mainly the K-Means clustering which extended to classification to test accuracy and Naive Bayes, the following resulted:

K-Mean neighbors:

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='euclidean',
        metric_params=None, n_jobs=1, n_neighbors=7, p=2,
        weights='distance')
```

Feature vector: latitude, longitude, UTC, year, month
Target variable: Nationality

	precision	recall	f1-score	support
American	1.00	1.00	1.00	200
British	0.99	1.00	0.99	86096
British	1.00	1.00	1.00	213
Dutch	1.00	0.99	1.00	115406
French	1.00	0.95	0.97	6601
Hamburg	1.00	0.99	0.99	68
Spanish	1.00	1.00	1.00	44333
avg / total	0.99	0.99	0.99	252917

Confusion Matrix

```
[[   200      0      0      0      0      0      0]
 [     0  86056      0     11      0      0     29]
 [     0      0    213      0      0      0      0]
 [     0    735      0 114671      0      0      0]
```

```
[    0   113     0   225  6263     0     0]
[    0     0     0     1     0    67     0]
[    0   154     0    37     6     0 44136]]
```

Analyzing this a great deal of information can be gathered. The British have been very dominant in the seas, followed by the Dutch. The precision very high showing how monotone the data is with very little grey area. However, looking at the confusion matrix, there have been few entries that are clustered with the wrong nationality. When looking at the French, it is clear that it has been clustered with the British and the Dutch 113 and 225 respectively. Hence these entries that have been clustered with them could be concluded as they were logged in the British or the Dutch territories.

The results of **Naive Bayes** have been very alarming:

GaussianNB()
Feature vector: latitude, longitude, UTC, year, month
Target variable: Nationality

	precision	recall	f1-score	support
American	0.00	0.00	0.00	200
British	0.40	0.59	0.47	86096
British	0.00	0.00	0.00	213
Dutch	0.59	0.64	0.61	115406
French	0.00	0.00	0.00	6601
Hamburg	0.00	0.00	0.00	68
Spanish	0.00	0.00	0.00	44333
avg / total	0.40	0.49	0.44	252917

Confusion Matrix

```
[[    0     0     0   200     0     0     0]
 [    0 50742     0 35354     0     0     0]
 [    0   119     0    94     0     0     0]
 [    0 41728     0 73678     0     0     0]
 [    0  3310     0  3291     0     0     0]
 [    0     3     0    65     0     0     0]
 [    0 32458     0 11875     0     0     0]]
```

Simply looking at the precision of the results, this model with its current feature vector is very poor. The confusion matrix shows just how dispersed the prediction is and its inaccuracy. The main problem of this model is most likely the result of a very small feature vector and lack of any other attributes to link a certain ship with a nation. Further preprocessing should take place in order to develop a better classification model.

Conclusion and future work

While the clustering of the logbook entries with the chosen attributes has proven to be very accurate and precise, the classification using Naive Bayes was terrible. This could simply be the result of these two programs needing certain attributes that are important to classify or cluster accurately.

Working with matplotlib to plot points, NumPy to store and manipulate data, and basemap has proved very useful in creating visual representation. I learned a lot in being able to create these visualizations that is customizable to the program.

Learning the K-Means clustering and the Naïve Bayes classification model helped tremendously in being able to implementing it with my program and using it with this specific dataset. I was able to analyze the results such as precision, confusion matrix, and recall using the material learned in class

Some of the extensions of this project could be running a correlation analysis regarding climate with the routes that the ships have taken. Furthermore, the classification model could be vastly improved by very fine tuned data preprocessing and using the attributes that are essential in determining the nationality. This could involve using categorical entries as well which could go into giving a distance function.