Lab 7: Clustering in Scikit-Learn

Data Science for Biologists • University of Washington • BIOL 419/519 • Winter 2019

Course design and lecture material by Bingni Brunton (https://github.com/bwbrunton) and Kameron Harris (https://github.com/kharris/). Lab design and materials by Eleanor Lutz (https://github.com/eleanorlutz/), with helpful comments and suggestions from Bing and Kam.

Table of Contents

Helpful resources

- Python Data Science Handbook (http://shop.oreilly.com/product/0636920034919.do) by Jake VanderPlas
- An introduction to machine learning with Scikit-Learn (https://scikit-learn.org/stable/tutorial/basic/tutorial.html)
- Scikit-Learn user guide (https://scikit-learn.org/stable/user_guide.html)
- Scikit-Learn Cheat Sheet (https://datacamp-community-prod.s3.amazonaws.com/5433fa18-9f43-44cc-b228-74672efcd116) by Python for Data Science

Data

- The data in this lab is from the Palmer Penguin Project (https://github.com/allisonhorst/palmerpenguins) by Dr. Kristen Gorman. The data was edited for teaching purposes.

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
plt.style.use("seaborn-colorblind") # Use a colorblind friendly color scheme
%matplotlib inline
```

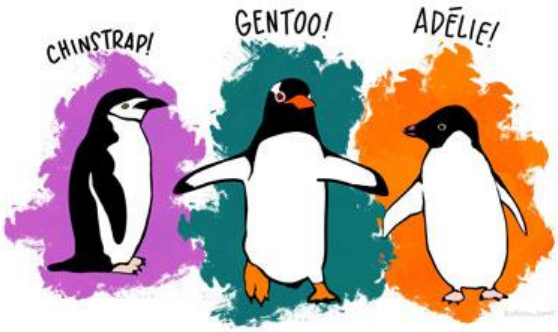
Lab 7 Part 1:

In previous labs, we created models that would either allow us to:

- predict one metric from another
- classify new data samples based on known categories in the training data

Both of these approaches were examples of supervised learning. We fit models using existing training data with the goal of predicting values for new test data.

Today, we want to see what patterns emerge from data when we apply the unsupervised learning technique of k-means clustering. This week's data consists of measurements taken from three species of penguins (we looked at this data previously in Lab 2).



Credit: Artwork by @allison_horst

Exercise 1: Read in the `penguin.csv` dataset and display the first five rows.

In [ ]:
```

```

Exercise 2: Create a scatterplot of culmen length vs culmen depth, with each species plotted as a different color.

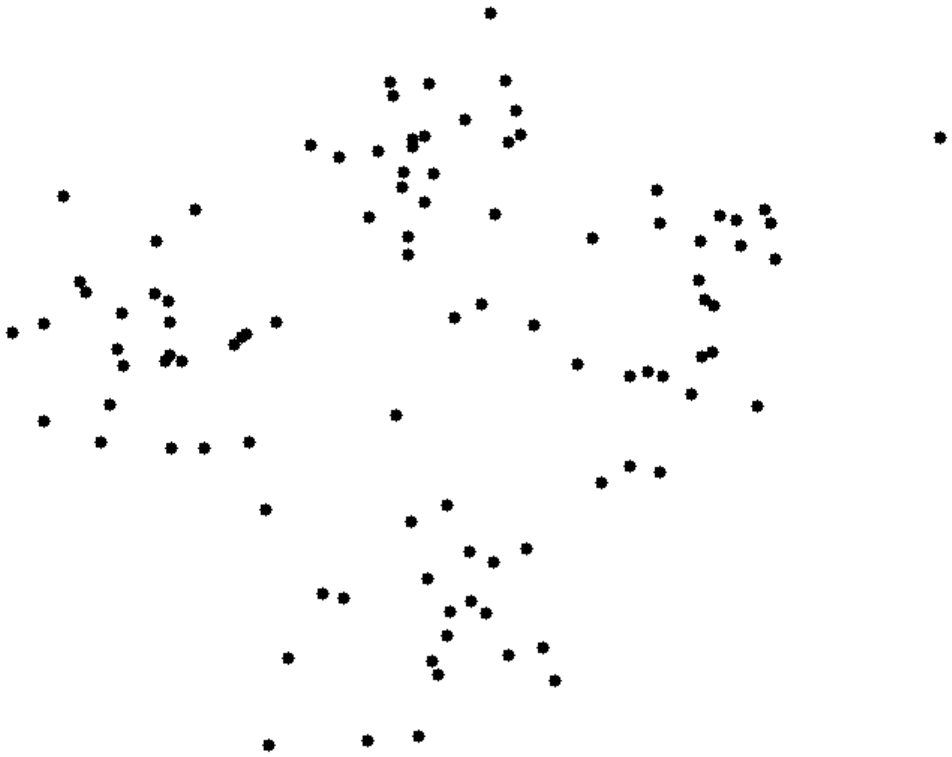
In [ ]:
```
1
```

K-means clustering with sci-kit learn

K-means clustering is an iterative clustering algorithm that is used to cluster unlabeled data. For example, the data you have may be unlabeled because it was collected by someone else who didn't have time to label everything by hand.

- **Initialization:** First, k-means randomly chooses k samples from the data to use as the initial cluster centers (k is the number of clusters).
- **Cluster Assignment:** Next, each data point is assigned to the cluster center that is closest to that data point.
- **Move centroid:** At this point each cluster center should have a set of data points associated with that cluster assignment. We will now update the cluster center to be the mean of all of the data points assigned to that cluster.
- **Iterate:** The previous step will likely move the cluster centers. If it did, we will repeat the process again and again until the centroids no longer move after the cluster assignment step.

Below is an animation showing the steps of K-means clustering for $k = 4$



Credit: Andrey A. Shabalin

To run the k-means algorithm in Scikit-learn, import `sklearn.cluster.KMeans` :

In [ ]:
```
1 from sklearn.cluster import KMeans
```

Exercise 3: First, split the dataframe into training and test data sets called `train_data` and `test_data` . Refer to the Lab 6 "Splitting data using Pandas" section for a fast way to separate data into training and test datasets.

In [ ]:
```
1
```

Since `KMeans` is an unsupervised learning algorithm, it does not use an answer dataset (like `LDA` from last week). We'll remove the species descriptions from both dataframes:

In [ ]:
```
train_data_n = train_data.drop("species", axis=1)
test_data_n = test_data.drop("species", axis=1)
```

Now we can use k-means clustering to fit a cluster called `cluster` to your penguin data matrix with $k = 2$ clusters:

In [ ]:
```
cluster = KMeans(n_clusters=2)
cluster.fit(train_data_n)
```

The documentation for Scikit-learn K-means (https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html) describes what can be done after fitting this model. For example, the attribute `labels_` returns an array of the cluster ID for every data point:

In [ ]:
```
cluster.labels_
```

Exercise 4: Create a new column, `labels`, in the `train_data` penguin data frame using the labels from the k-means model. Display the first five rows of the dataframe. Plot culmen length by culmen depth from this dataframe with points colored by the k-means cluster labels.

In [ ]:
```

```

Another helpful Scikit-learn method is `predict`, which uses the previously created clustering algorithm to predict the closest cluster for the given data. We can use this to predict the cluster for a previously unseen data point, or to look at different data points within the training data:

In [ ]:
```
cluster.predict(test_data_n)
```

Now we can compare the predicted cluster IDs to the known species names for each sample to see how well the machine learning algorithm matches the species classification.

In [ ]:
```
test_data['label'] = cluster.predict(test_data_n)
test_data.groupby(['variety','label']).size().reset_index(name='counts')
```

Exercise 5: Cluster your penguin data again using $k = 3$ clusters. Display the first five rows of the training dataset. After clustering, replace your original cluster labels in `train_data` with the new labels and plot culmen length by culmen depth (scatterpoints colored by cluster label).

In [ ]:
```

```

Exercise 6: Use this new cluster to predict the cluster ID of the test data. Display the number of samples that belong to each cluster-species group. Does it seem like there is one species that clusters better than the others?

In [ ]:
```

```

Exercise 7: Cluster only the `culmen.length` and `culmen.depth` of the penguin data matrix into $k = 3$ clusters. Display the first five rows of the training dataset. Repeat the plot above. How do the results change with fewer dimensions for the model?

In [ ]:
```

```

Lab 7 Bonus exercise

Bonus Exercise 1: Visualize the success of the clustering algorithm compared to the true species values. Plot the culmen length, culmen depth, and flipper length in a 3D plot. Each scatterpoint should be colored according to species, and the clustering results should be represented by different marker shapes (circle, X, diamond, etc). *Hint:* Lab 4 includes code for plotting a 3D plot in Bonus Exercise 2.

In [ ]:
```

```