# M6-05: k-Means Clustering in Python

*Part of the "Towards Machine Learning" Learning Badge*

***Video Walkthrough:*** https://discovery.cs.illinois.edu/m6-05/

## K-Means Clustering In Python

All models in the scikit-learn (sklearn) Python library follow the same basic design. The simplest clustering model is k-means, creating k clusters based on an average "centroid".

## Step 0: Explore a DataSet

To show a real-world example of clustering, we will discover a new dataset -- this dataset includes 15 votes by United States senators in during the 114th Congress:

| name | party | state | vote1 | vote2 | ... | vote15 |
|---|---|---|---:|---:|:---:|---:|
| Alexander | R | TN | 0 | 1 | ... | 0 |
| Ayotte | R | NH | 0 | 1 | | 0 |
| Baldwin | D | WI | 1 | 0 | | 1 |
| Barrasso | R | WY | 0 | 1 | | 0 |
| Bennet | D | CO | 0 | 0 | | 0 |
| Blumenthal | D | CT | 1 | 0 | | 1 |
| ... | | | | | | |

**Dataset URL:** https://waf.cs.illinois.edu/discovery/congress.csv

This dataset has 100 rows, one for each member of the United States Senate. Each vote is encoded where a 0 indicates a "No" vote, a 1 indicates a "Yes" vote, and a 0.5 indicates an "Abstain" or "Absent" vote.

Given that many votes are "party line" votes, we may be able to use clustering to learn the party membership based entirely on their voting history.

## Step 1: Initialize the Machine Learning Model

The first step is to always create an instance of the machine learning model, which provides Python an object to use to store the parameters of the model. For all models, we must import the package and then initialize the model:

**Python:**

```python
# Import sklearn's KMeans model:
from sklearn.cluster import KMeans

# Create a new instance of a KMeans model:
```

**Description:** Imports and creates an instance of sklearn's k-means clustering model.

## Step 2: Train the Model using `.fit(...)`

The k-means clustering algorithm is an unsupervised learning algorithm, so we only need a list of independent variables -- no dependent variable is needed!

The list of variables -- or features -- that we want to use is:

| | |
|---|---|
| **Python:** | |
| **Description:** | List of all features we will use in k-means clustering. |

Using the features, fit the k-means model:

| | |
|---|---|
| **Python:** | |
| **Description:** | Fits the k-means model with the data in the DataFrame df. |

## Step 3: Use the Model using `.predict(...)`

In the case of clustering, the predict function will predict a cluster that the senator belongs to. The prediction must be based on a list of features. In this case, we are predicting on the same dataset we used to fit the model:

| | |
|---|---|
| **Python:** | |
| **Description:** | Fits the k-means model with the data in the DataFrame df. |

**Puzzle #1:** What cluster includes most of the Democrats ("D")? Republicans ("R")?

**Puzzle #2:** How many senators were clustered incorrectly?