

```
1  """
2  Clusters Bob Ross paintings by features.
3
4  By Walter Hickey <walter.hickey@fivethirtyeight.com>
5
6  See http://fivethirtyeight.com/features/a-statistical-analysis-of-the-work-of-bob-ross/
7  """
8
9  import numpy as np
10 from scipy.cluster.vq import vq, kmeans, whiten
11 import math
12 import csv
13
14 def main():
15
16     # load data into vectors of 1s and 0s for each tag
17     with open('elements-by-episode.csv', 'r') as csvfile:
18         reader = csv.reader(csvfile)
19         reader.next() # skip header
20         data = []
21         for row in reader:
22             data.append(map(lambda x: int(x), row[2:])) # exclude EPISODE and TITLE
23
24     # convert to numpy matrix
25     matrix = np.array(data)
26
27     # remove columns that have been tagged less than 5 times
28     columns_to_remove = []
29     for col in range(np.shape(matrix)[1]):
30         if sum(matrix[:,col]) <= 5:
31             columns_to_remove.append(col)
32     matrix = np.delete(matrix, columns_to_remove, axis=1)
33
34     # normalize according to stddev
35     whitened = whiten(matrix)
36     output = kmeans(whitened, 10)
37
38     print "episode", "distance", "cluster"
39
40     # determine distance between each of 403 vectors and each centroid, find closest neighbor
41     for i, v in enumerate(whitened):
42
43         # distance between centroid 0 and feature vector
44         distance = math.sqrt(sum((v - output[0][0]) ** 2))
45
46         # group is the centroid it is closest to so far, set initially to centroid 0
47         group = 0
48         closest_match = (distance, group)
49
50         # test the vector i against the 10 centroids, find nearest neighbor
51         for x in range(0, 10):
52             dist_x = math.sqrt(sum((v - output[0][x]) ** 2))
53             if dist_x < closest_match[0]:
54                 closest_match = (dist_x, x)
55
56         print i+1, closest_match[0], closest_match[1]
57
58 if __name__ == "__main__":
59     main()
```

