Assignment 4 Report

Arnav Bhutani
Alison Jones
Justin Parks

Introduction:

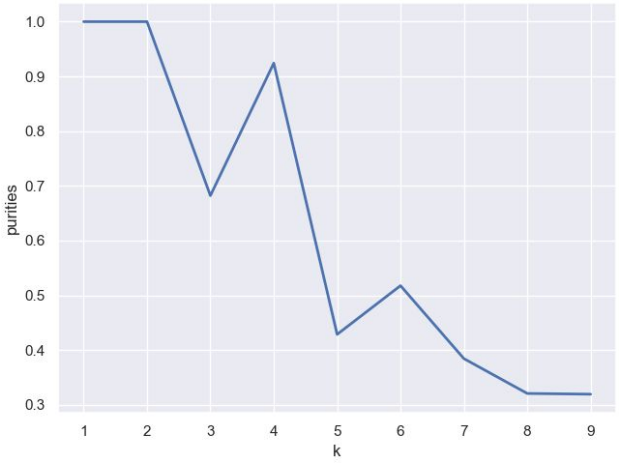
For this project, we split the work up between the two parts, with Justin taking most of the K-means functionality and Arnav taking the lead on the PCA section. Alison doubled up with Arnav for PCA, but also ran support for Justin. For a general estimate of contribution, Arnav did 40%, Alison did about 25%, and Justin did about 35% of this project.

K-Means:

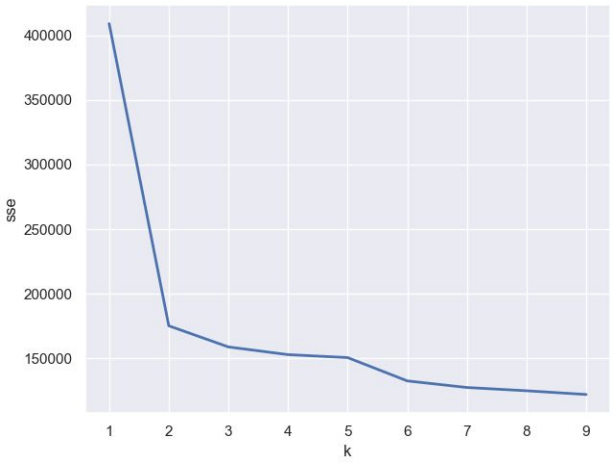
1.1 - For this problem, when attempting to seed the clustered, I used a method of sampling without replacement in order to make sure no two clusters accidentally initialized to the same value as another. This proved pretty useful, as I didn't run into many zero-cluster issues.

1.2 - For this, I implemented a simple for-loop that looped 5 times and averaged the three values received from the single loop, train_sses_vs_iter, train_sses_vs_k, and train_purities_vs_k. As the random number generator is seeded, no repeat cluster centers were generated. Below are the plots generated for the three subparts:

Purities vs K



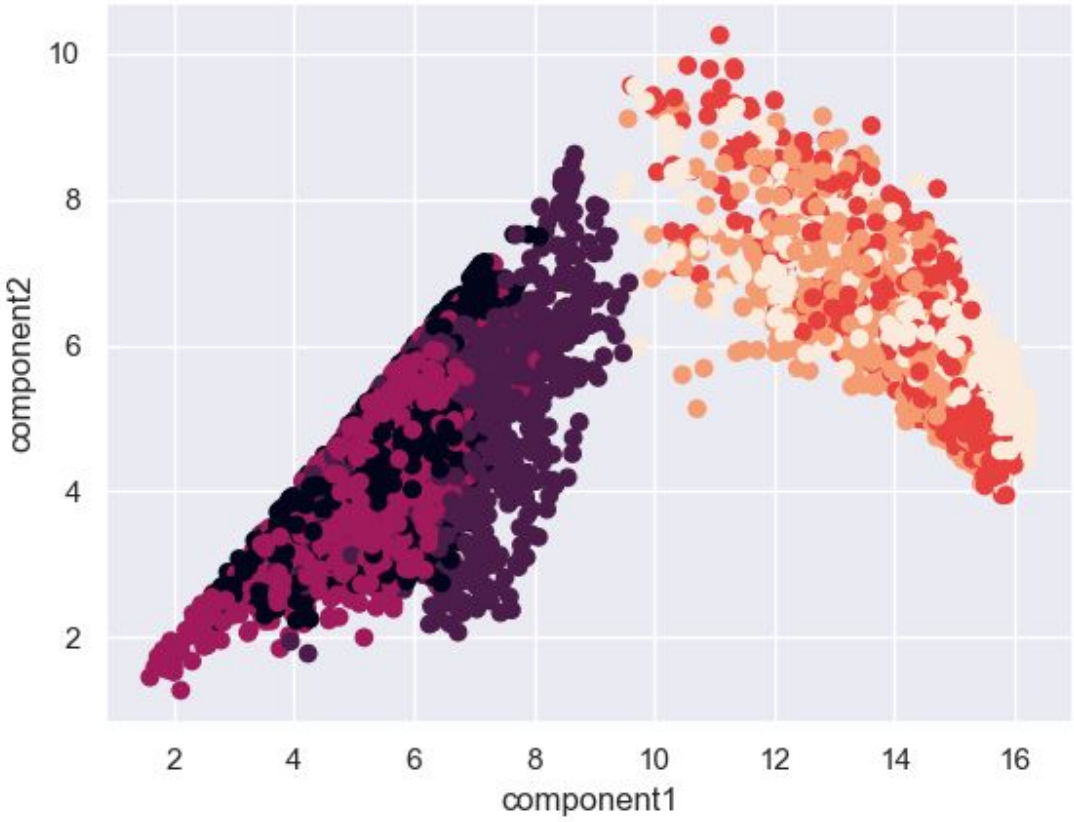
SSE vs K



PCA:

2.1 -

2.2 - As you can see the data has been split very cleanly by the two components. They are divided very evenly between the six classifications.



2.3 - Unable to get this working. Here are some code notes for this section.

```
        sorted_array = np.sort(self.eig_vals)
    popped_val = sorted_array[-1]
    sorted_array = sorted_array[:-1]
    some_value = np.matmul(x, self.eig_vecs)
    some_value = some_value * self.retain_ratio
    while some_value >= 0:
        popped_val = sorted_array[-1]
        sorted_array = sorted_array[:-1]
        Some_value-popped_val
```

OR

```
removed_indices = []
    some_array = np.matmul(x, self.eig_vecs)
    print("past here")
    for i in range(len(some_array)):
```

```
  print(i)
  sum = np.sum(some_array[i])
  print("sum: ", sum)
  while sum >= 0:
     min = np.amin(some_array[i])
     sum - some_array[i][min]
     removed_indices.append(min)
     some_array = np.delete(some_array[i], min)
  print(removed_indices[i])
```

2.4 - N/A