

## Assignment: Clustering Image Data

The comments/sections provided are your cues to perform the assignment. You don't need to limit yourself to the number of rows/cells provided. You can add additional rows in each section to add more lines of code.

If at any point in time you need help on solving this assignment, view our demo video to understand the different steps of the code.

Happy coding!

## Clustering Image Data

### DESCRIPTION

Analyzing images with codes can be difficult. Therefore, it's necessary for you to make your code understand the context of an image. In doing so, the first step will be identifying the dominant colors.

[Hint: Refer the following url for image processing documentation: <http://omz-software.com/pythonista/docs/ios/PIL.html>]

**Objective:** To identify the dominant color in the image.

### Action to Perform:

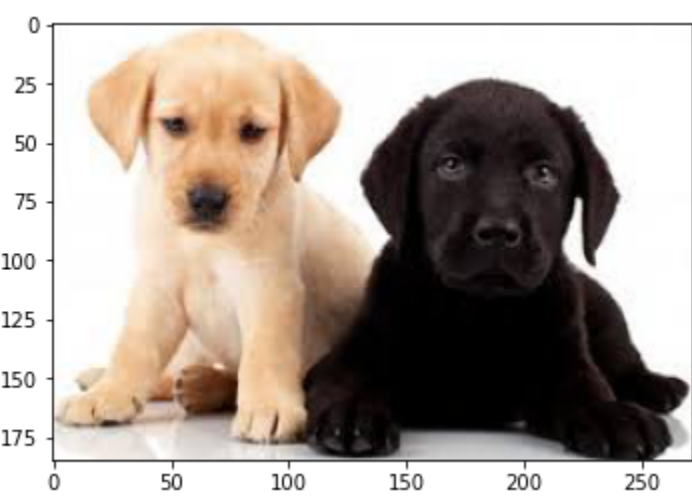
1. Open and display the image "dog.jpeg."
2. Use K-means clustering for image segmentation, which will include the following steps:
  - A. Find out the dimensions of the image and convert it to a two-dimensional array.
  - B. Use K-means clustering with k set to 3 and cluster the image. [Hint: Refer to the K-means module of scikit learn]
  - C. Predict the cluster label of every pixel in the image and plot it back as an image.
  - D. Find out the three dominant colors in the image. [Hint: The cluster centers should correspond to three dominant colors]

```
In [1]: #Step 1: Import libraries
from PIL import Image
from sklearn.cluster import KMeans
from matplotlib import pyplot as plot
import numpy as np
%matplotlib inline
```

### Open and display the image "dog.jpeg."

```
In [2]: image = Image.open('dogs.jpeg')
plot.imshow(image)
```

```
Out[2]: <matplotlib.image.AxesImage at 0x1168e448>
```



### Find out the dimensions of the image and convert it to a two-dimensional array

```
In [3]: #Step 2: Get the Image and its Corresponding RGB Values
image_np = np.asarray(image)
image_np[0:2]
```

```
Out[3]: array([[240, 240, 240],
               [255, 255, 255],
               [255, 255, 255],
               ...,
               [254, 254, 254],
               [254, 254, 254],
               [254, 254, 254]],

              [[240, 240, 240],
               [255, 255, 255],
               [255, 255, 255],
               ...,
               [254, 254, 254],
               [254, 254, 254],
               [254, 254, 254]]], dtype=uint8)
```

```
In [4]: #Step 3: Get the Image Dimensions
image_np.shape
```

```
Out[4]: (185, 272, 3)
```

```
In [5]: #Step 4: Reshape the Data
pixels=image_np.reshape(image_np.shape[0]*image_np.shape[1],image_np.shape[2])
pixels.shape
```

```
Out[5]: (50320, 3)
```

### Use K-means clustering with K set to 3 and cluster the image

[Hint: Refer to the K-means module of scikit learn]

```
In [6]: #Step 5: Define the K-means Model
kmeans = KMeans(n_clusters=3, random_state=0)
kmeans.fit(pixels)
```

```
Out[6]: KMeans(n_clusters=3, random_state=0)
```

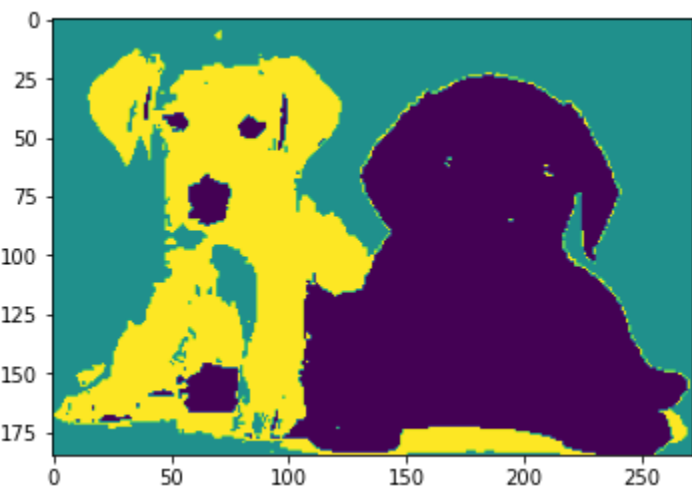
```
In [7]: #Step 6: Define the Cluster Centres
cluster_centers=kmeans.cluster_centers_
cluster_centers
```

```
Out[7]: array([[ 29.48793962,  21.56369305,  20.46157633],
               [251.43356433, 247.54268354, 242.92964748],
               [210.41334653, 174.54062526, 143.39313701]])
```

### Predict the cluster label of every pixel in the image and plot it back as an image.

```
In [8]: #Step 7: Predict the cluster label of every pixel
labels = kmeans.predict(pixels)
labels = labels.reshape((185,272))
plot.imshow(labels)
```

```
Out[8]: <matplotlib.image.AxesImage at 0x11b15850>
```

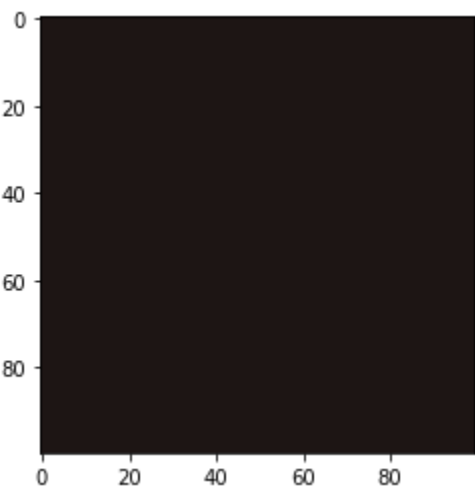


### Find out the three dominant colors in the image.

[Hint: The cluster centers should correspond to three dominant colors]

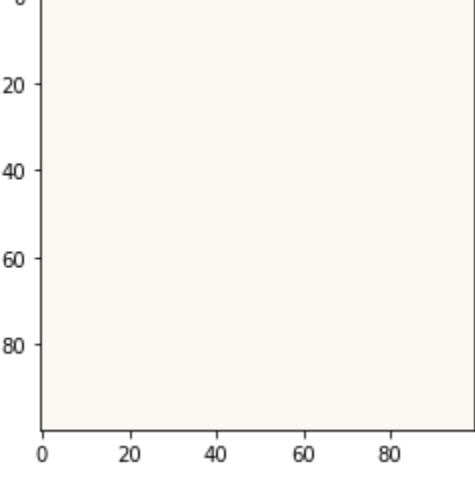
```
In [9]: #Step 7: Predict the three Dominant colors in the image
#Plot the First Dominant Color
blob = np.ones((100,100,3))
blob1 = blob * cluster_centers[0]
plot.imshow(blob1.astype(np.uint8))
```

```
Out[9]: <matplotlib.image.AxesImage at 0x11b574c0>
```



```
In [10]: #Plot the Second Dominant Color
blob2 = blob * cluster_centers[1]
plot.imshow(blob2.astype(np.uint8))
```

```
Out[10]: <matplotlib.image.AxesImage at 0x11b8b298>
```



```
In [11]: #Plot the Third Dominant Color
blob3 = blob * cluster_centers[2]
plot.imshow(blob3.astype(np.uint8))
```

```
Out[11]: <matplotlib.image.AxesImage at 0x11bd0b8>
```

