EDA_Color_of_Poster

April 19, 2017

1 EDA on Colors of Poster

1.1 Here we did some exploratory data analysis on color of posters. There are three tasks accomplished: 1) Download movie posters of horror, romance and scifi movies. 2) Decompose the color of each pixel and visualize the frequency of each color (number of pixels showing this color) using HSV color scheme (3D visualization). 3) Decompose the color of each pixel and visualize the frequency of each bin of color (1,000 bins roughly represent every color) using RGB color scheme (2D visualization).

1.2 The 1,000 bins in task 3 come from 10 bins (in R value) * 10 bins (in G value) * 10 bins (in B value). We can also transform it into 1,000 features, where the value of each feature is the number of pixels with the RGB value that falls into this certain bin. Later on we can use this information for classification models.

In [30]:
```
import wget
import json
import http.client
import time
```

In [31]:
```
##### Request movie data: genre == Romance #####

conn = http.client.HTTPSConnection("api.themoviedb.org")
payload = "{}"
conn.request("GET", "/3/discover/movie?with_genres=10749&primary_release_y
res = conn.getresponse()
encoding = res.info().get_content_charset('utf8')
data = json.loads(res.read().decode(encoding))
```

In [33]:
```
##### Download movie poster: genre == Romance #####

for i in range(len(data["results"])):
    poster_path = data["results"][i]["poster_path"]
    url = "https://image.tmdb.org/t/p/w185"
    url = url + poster_path
    url
    wget.download(url,"./romance_185/")
    time.sleep(1)
```

```
100% [...] 12014 / 12014
```

```
In [16]: ##### Request movie data: genre == Horror #####

         conn = http.client.HTTPSConnection("api.themoviedb.org")
         payload = "{}"
         conn.request("GET", "/3/discover/movie?with_genres=27&primary_release_year
         res = conn.getresponse()
         encoding = res.info().get_content_charset('utf8')
         data = json.loads(res.read().decode(encoding))
```

```
In [17]: ##### Download movie poster: genre == Horror #####

         for i in range(len(data["results"])):
             poster_path = data["results"][i]["poster_path"]
             url = "https://image.tmdb.org/t/p/w185"
             url = url + poster_path
             url
             wget.download(url,"./horror_185/")
             time.sleep(1)
```

```
100% [...] 5950 / 5950
```

```
In [128]: ##### Request movie data: genre == Scifi #####

          conn = http.client.HTTPSConnection("api.themoviedb.org")
          payload = "{}"
          conn.request("GET", "/3/discover/movie?with_genres=878&primary_release_ye
          res = conn.getresponse()
          encoding = res.info().get_content_charset('utf8')
          data = json.loads(res.read().decode(encoding))
```

```
In [130]: ##### Download movie poster: genre == Scifi #####

          for i in range(len(data["results"])):
              poster_path = data["results"][i]["poster_path"]
              url = "https://image.tmdb.org/t/p/w185"
              url = url + poster_path
              url
              wget.download(url,"./scifi_185/")
              time.sleep(1)
```

```
100% [...] 17015 / 17015
```

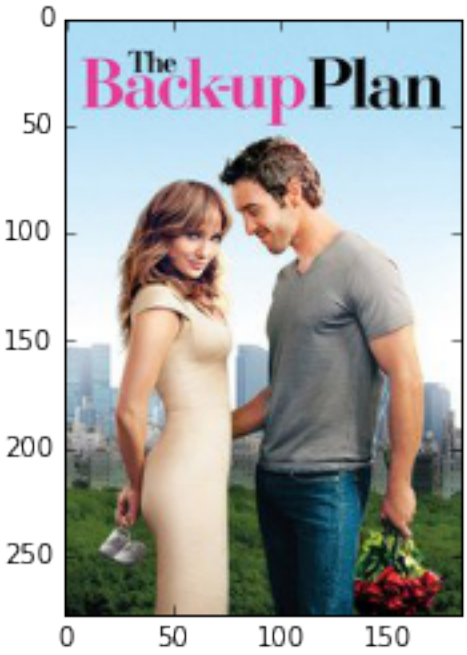
```
In [23]: import numpy as np
         import mpl_toolkits.mplot3d.axes3d as p3
         import matplotlib.pyplot as plt
         import colorsys
         from PIL import Image
```

In [13]:
```
##### The function decompose and visualize colors of posters #####
##### HSV color scheme, 3D visualization #####

def color_decomposition(img_file):
    img = img_file.load()
    [xs, ys] = img_file.size
    max_intensity = 100
    hues = {}
    for x in range(0, xs):
        for y in range(0, ys):
            [r, g, b] = img[x, y]

            r /= 255.0
            g /= 255.0
            b /= 255.0

            [h, s, v] = colorsys.rgb_to_hsv(r, g, b)

            if h not in hues:
                hues[h] = {}
            if v not in hues[h]:
                hues[h][v] = 1
            else:
                if hues[h][v] < max_intensity:
                    hues[h][v] += 1
    h_ = []
    v_ = []
    i = []
    colours = []

    for h in hues:
        for v in hues[h]:
            h_.append(h)
            v_.append(v)
            i.append(hues[h][v])
            [r, g, b] = colorsys.hsv_to_rgb(h, 1, v)
            colours.append([r, g, b])

    %matplotlib inline
    fig = plt.figure()
    ax = p3.Axes3D(fig)
    ax.scatter(h_, v_, i, s=5, c=colours, lw=0)
    ax.set_xlabel('Hue')
    ax.set_ylabel('Value')
    ax.set_zlabel('Intensity')
    fig.add_axes(ax)
    plt.show()
```

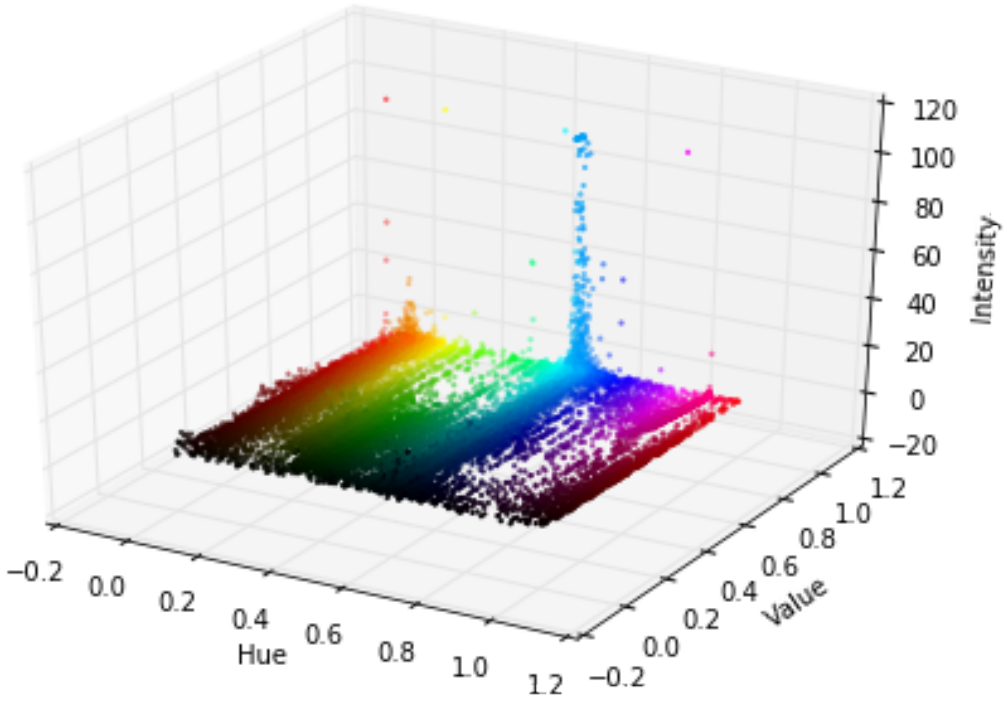
In [27]:
```
from matplotlib.pyplot import imshow
```

In [28]:
```
##### One example poster of romance movie #####

romance_1 = Image.open("./romance_185/1.jpg")
%matplotlib inline
pil_im = Image.open('./romance_185/1.jpg')
imshow(np.asarray(pil_im))
```

Out[28]:
```
<matplotlib.image.AxesImage at 0xb03f6a0>
```



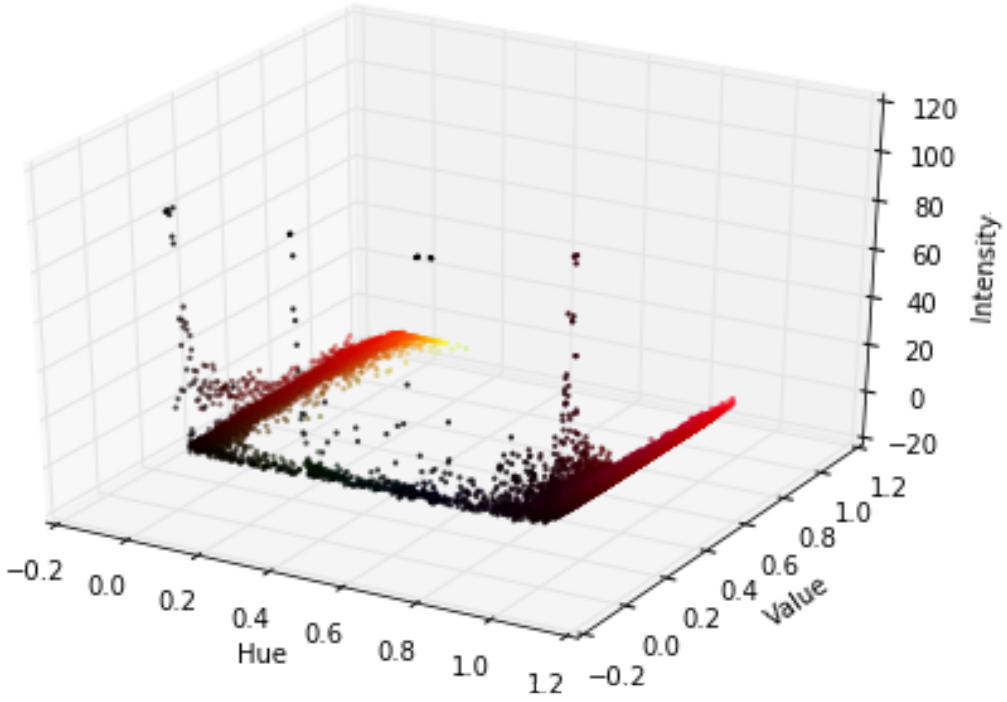
In [15]:
```
##### The color decomposition of this romance poster #####

color_decomposition(romance_1)
```



In [29]:
```
##### One example poster of horror movie #####

horror_1 = Image.open("./horror_185/1.jpg")
%matplotlib inline
pil_im = Image.open('./horror_185/1.jpg')
imshow(np.asarray(pil_im))
```

Out[29]:
```
<matplotlib.image.AxesImage at 0xaf65c18>
```

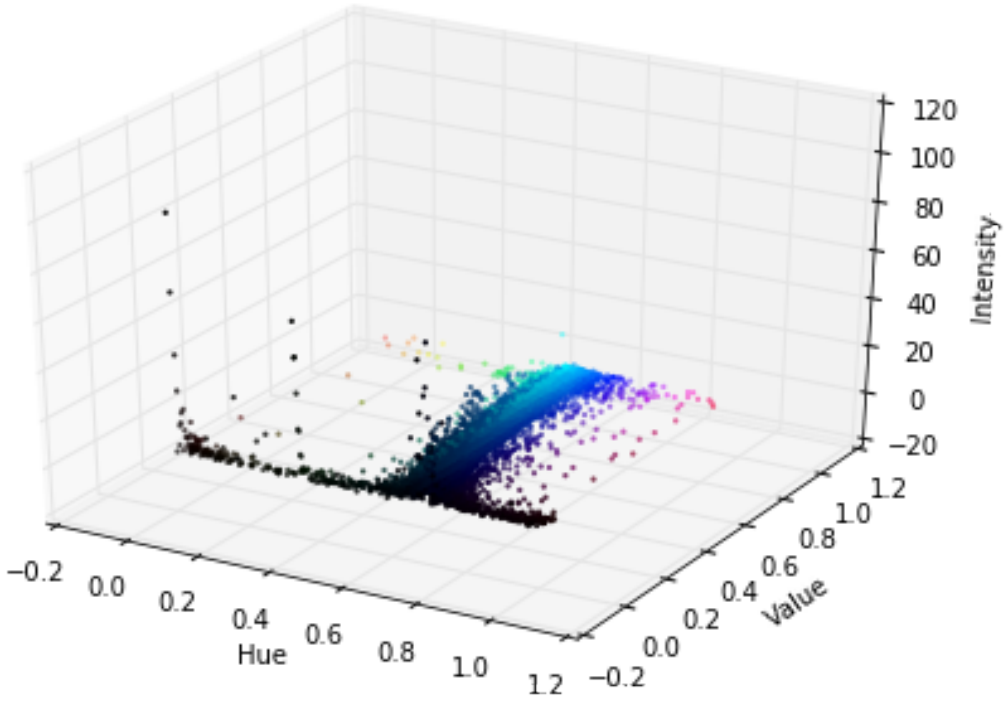
In [19]:
```
##### The color decomposition of this horror poster #####

color_decomposition(horror_1)
```



In [133]:
```
##### One example poster of horror movie #####

scifi_1 = Image.open("./scifi_185/1.jpg")
%matplotlib inline
pil_im = Image.open('./scifi_185/1.jpg')
imshow(np.asarray(pil_im))
```

Out[133]: <matplotlib.image.AxesImage at 0xcf15710>

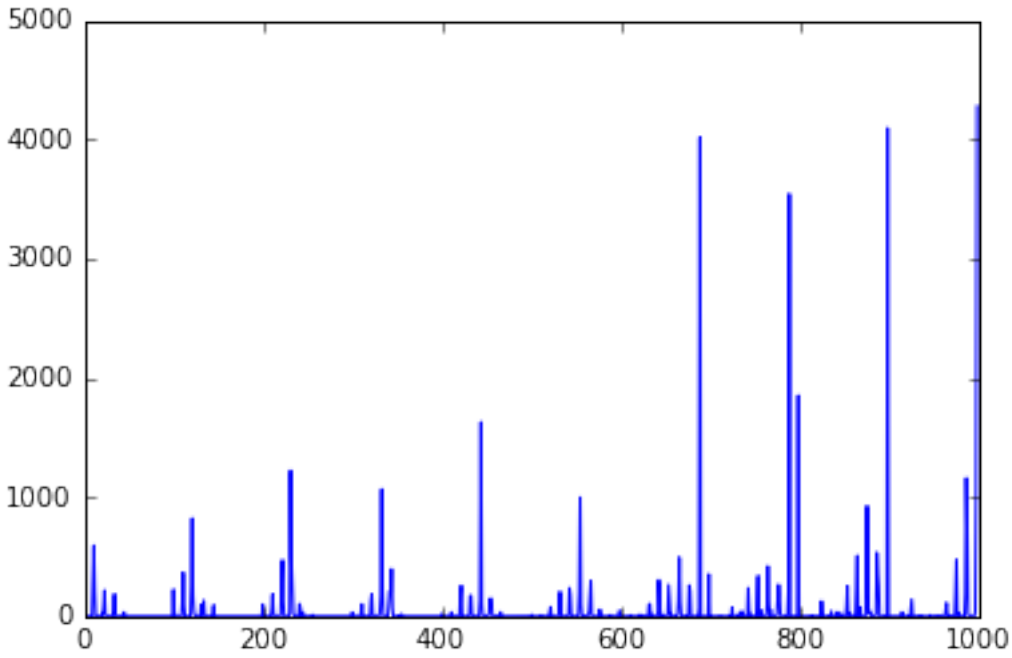
In [134]:
```
##### The color decomposition of this horror poster #####

color_decomposition(scifi_1)
```



In [115]:
```
##### The function that decompose the color of poster #####
##### RGB scheme (1,000 bins to represent colors), 2D visualization ####

def create_rgb_feature(img_file):
    img = img_file.load()
    [xs, ys] = img_file.size
    c = [[[0,0,0] for col in range(ys)] for row in range(xs)]
    for x in range(0, xs):
        for y in range(0, ys):
            [r, g, b] = img[x, y]
            c[x][y]=[r,g,b]

    flat_image = np.array(sum(c,[]))
    H, edges = np.histogramdd(flat_image, bins = (10, 10, 10))
    flat_H = [item for first in H for second in first for item in second]
    return flat_H
```

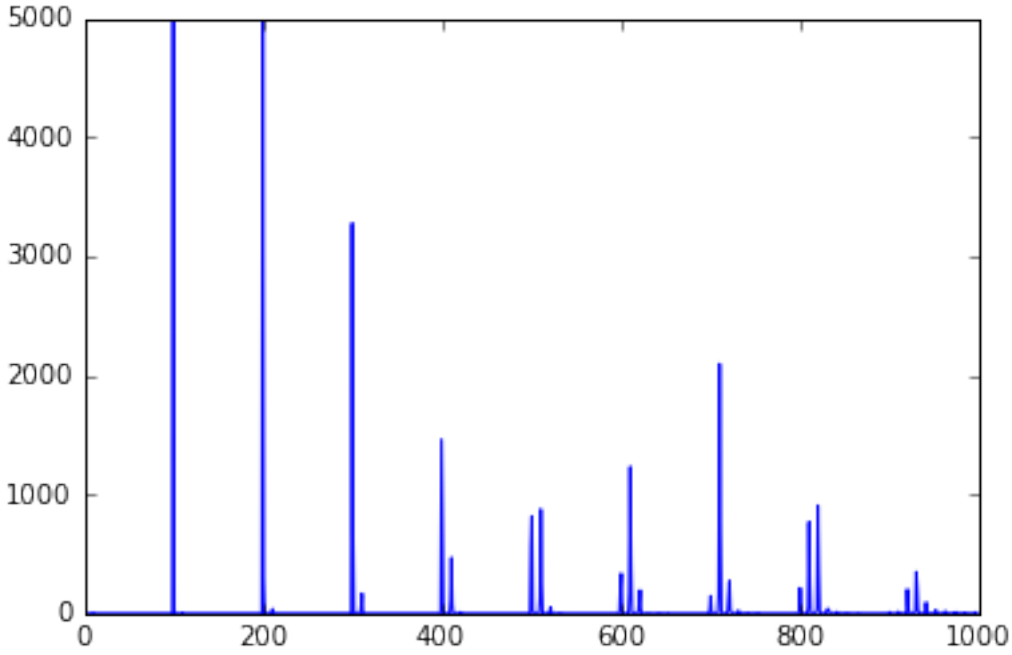
In [126]:
```
##### The frequency of 1,000 color bins of this romance poster #####

plt.plot(range(1000),create_rgb_feature(romance_1))
plt.ylim(0,5000)
```

Out[126]: (0, 5000)



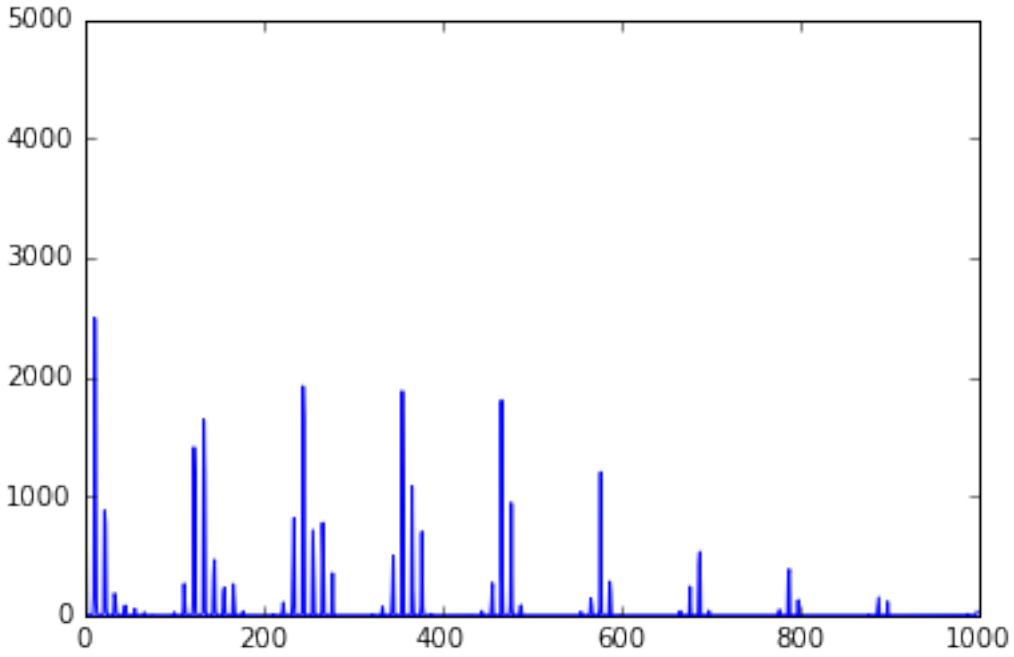
In [127]:
```
##### The frequency of 1,000 color bins of this horror poster #####

plt.plot(range(1000),create_rgb_feature(horror_1))
plt.ylim(0,5000)
```

Out[127]: (0, 5000)



In [135]:
```
##### The frequency of 1,000 color bins of this scifi poster #####

plt.plot(range(1000),create_rgb_feature(scifi_1))
plt.ylim(0,5000)
```

Out[135]: (0, 5000)



In [ ]: