

Kmeans_Personalidad

July 23, 2022

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
from sklearn.cluster import KMeans
from sklearn.metrics import pairwise_distances_argmin_min
from mpl_toolkits.mplot3d import Axes3D
plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')
```

```
[2]: dataframe = pd.read_csv(r" analisis.csv")
dataframe.head()
```

```
[2]:
```

	usuario	op	co	ex	ag	ne \
0	3gerardpique	34.297953	28.148819	41.948819	29.370315	9.841575
1	aguerosergiokun	44.986842	20.525865	37.938947	24.279098	10.362406
2	albertochicote	41.733854	13.745417	38.999896	34.645521	8.836979
3	AlejandroSanz	40.377154	15.377462	52.337538	31.082154	5.032231
4	alfredocasero1	36.664677	19.642258	48.530806	31.138871	7.305968

	wordcount	categoria
0	37.0945	7
1	78.7970	7
2	49.2604	4
3	80.4538	2
4	47.0645	4

```
[3]: dataframe.describe()
```

```
[3]:
```

	op	co	ex	ag	ne	wordcount \
count	140.000000	140.000000	140.000000	140.000000	140.000000	140.000000
mean	44.414591	22.977135	40.764428	22.918528	8.000098	98.715484
std	8.425723	5.816851	7.185246	7.657122	3.039248	44.714071
min	30.020465	7.852756	18.693542	9.305985	1.030213	5.020800
25%	38.206484	19.740299	36.095722	17.050993	6.086144	66.218475
50%	44.507091	22.466718	41.457492	21.384554	7.839722	94.711400
75%	49.365923	26.091606	45.197769	28.678866	9.758189	119.707925

```
max      71.696129   49.637863   59.824844   40.583162   23.978462   217.183200
```

```

      categoria
count  140.000000
mean    4.050000
std     2.658839
min     1.000000
25%     2.000000
50%     3.500000
75%     7.000000
max     9.000000
```

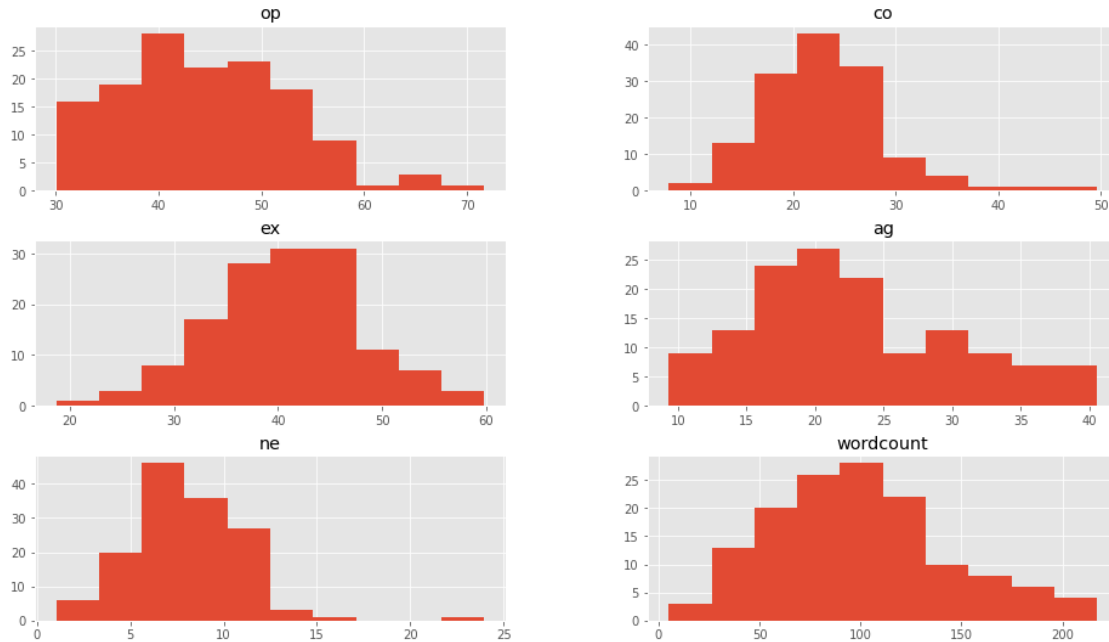
```
[4]: dataframe.groupby('categoria').size()
```

```
[4]: categoria
1     27
2     34
3      9
4     19
5      4
6      8
7     17
8     16
9      6
dtype: int64
```

```
[5]: dataframe.drop(['categoria'],1).hist()
plt.show()
```

/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only

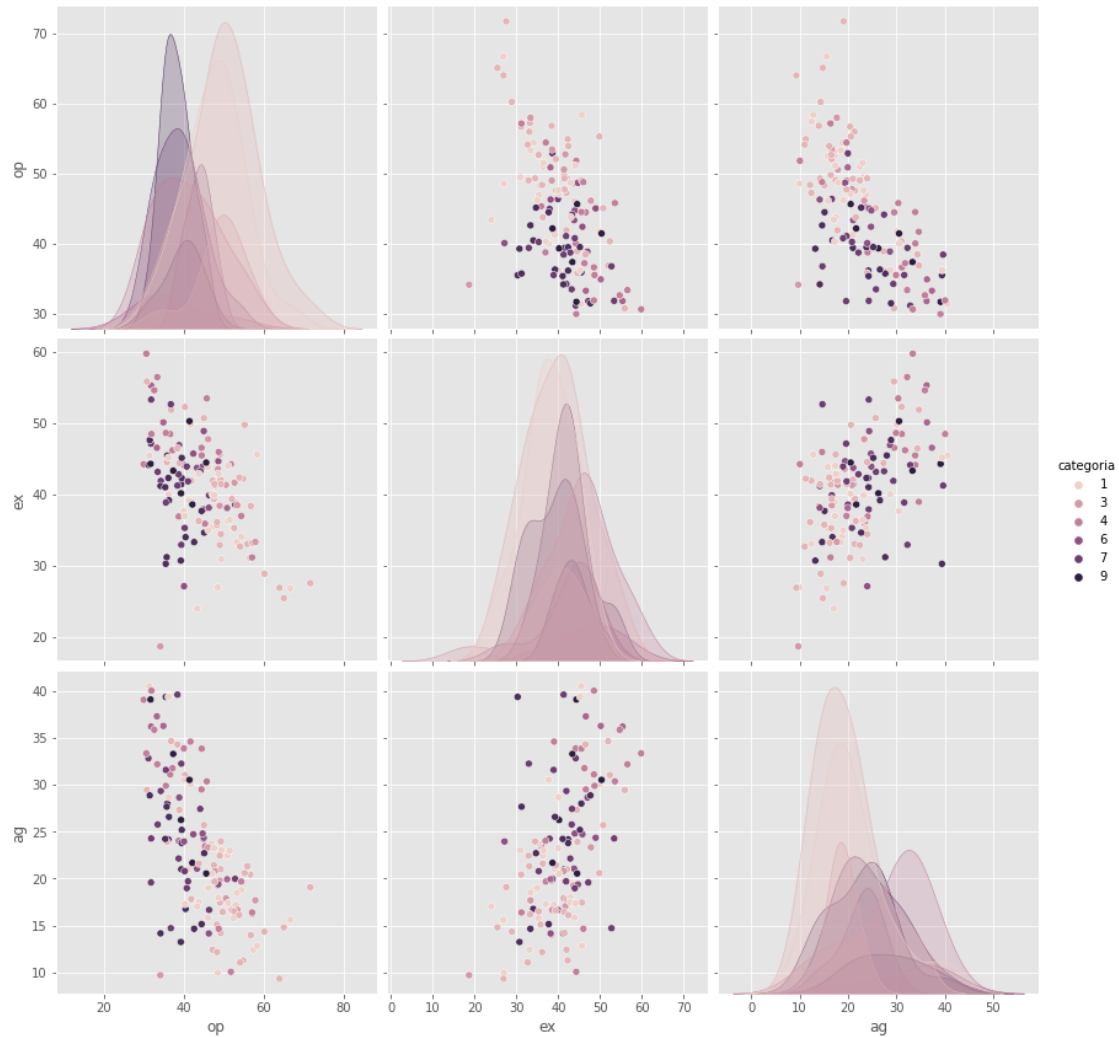
"""Entry point for launching an IPython kernel.



```
[6]: sb.pairplot(dataframe.dropna(),  
    ↪ hue='categoria',size=4,vars=["op","ex","ag"],kind='scatter')
```

```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-  
packages/seaborn/axisgrid.py:2076: UserWarning: The `size` parameter has been  
renamed to `height`; please update your code.  
warnings.warn(msg, UserWarning)
```

```
[6]: <seaborn.axisgrid.PairGrid at 0x7fd13251eda0>
```



```
[7]: X = np.array(dataframe[["op","ex","ag"]])
      y = np.array(dataframe['categoria'])
      X.shape
```

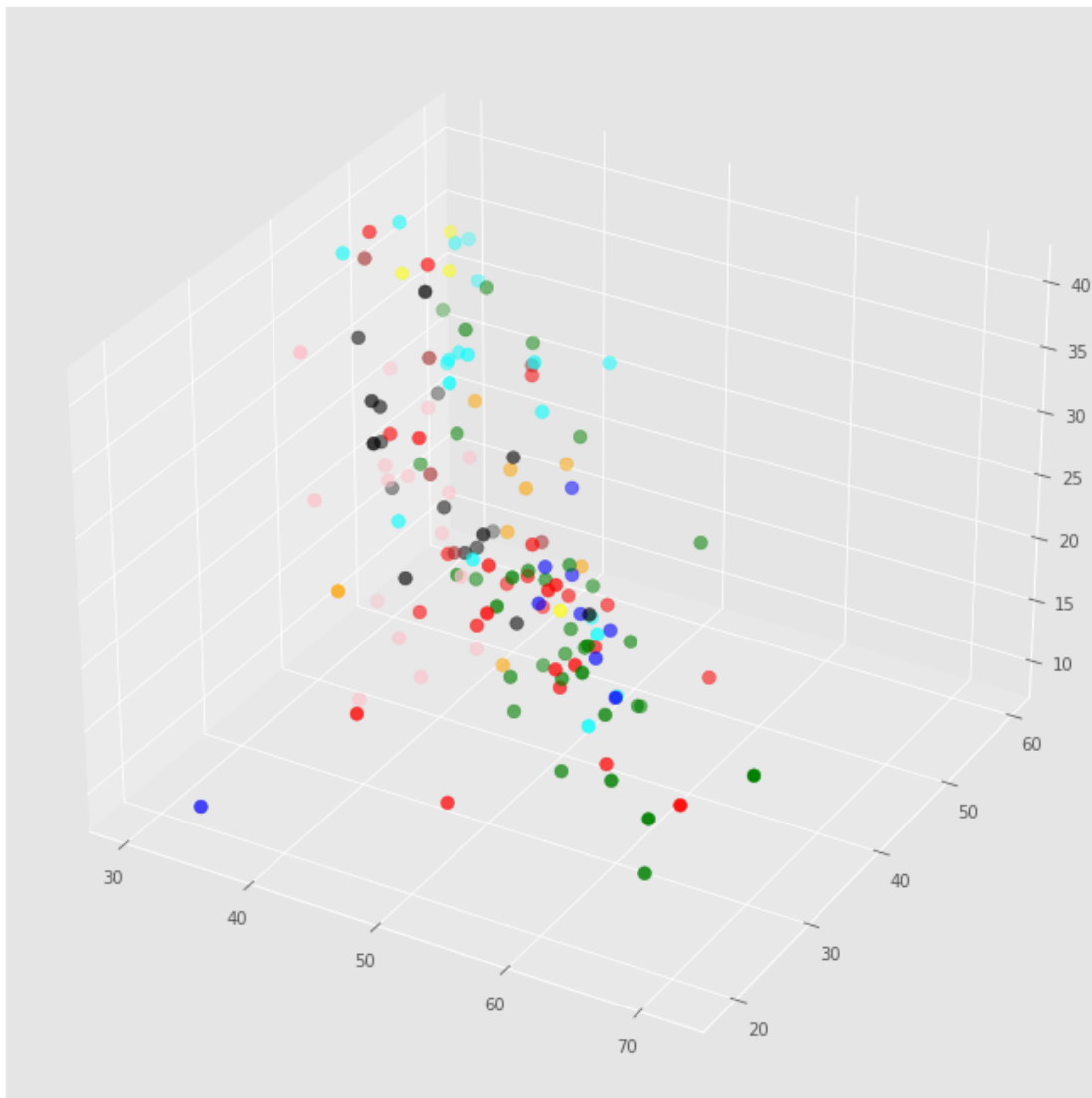
```
[7]: (140, 3)
```

```
[8]: fig = plt.figure()
      ax = Axes3D(fig)
      colores=['blue','red','green','blue','cyan','yellow','orange','black','pink','brown','purple']
      asignar=[]
      for row in y:
          asignar.append(colores[row])
      ax.scatter(X[:, 0], X[:, 1], X[:, 2], c=asignar,s=60)
```

/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-

packages/ipykernel_launcher.py:2: MatplotlibDeprecationWarning: Axes3D(fig) adding itself to the figure is deprecated since 3.4. Pass the keyword argument auto_add_to_figure=False and use fig.add_axes(ax) to suppress this warning. The default value of auto_add_to_figure will change to False in mpl3.5 and True values will no longer work in 3.6. This is consistent with other Axes classes.

[8]: <mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x7fd135b92e48>

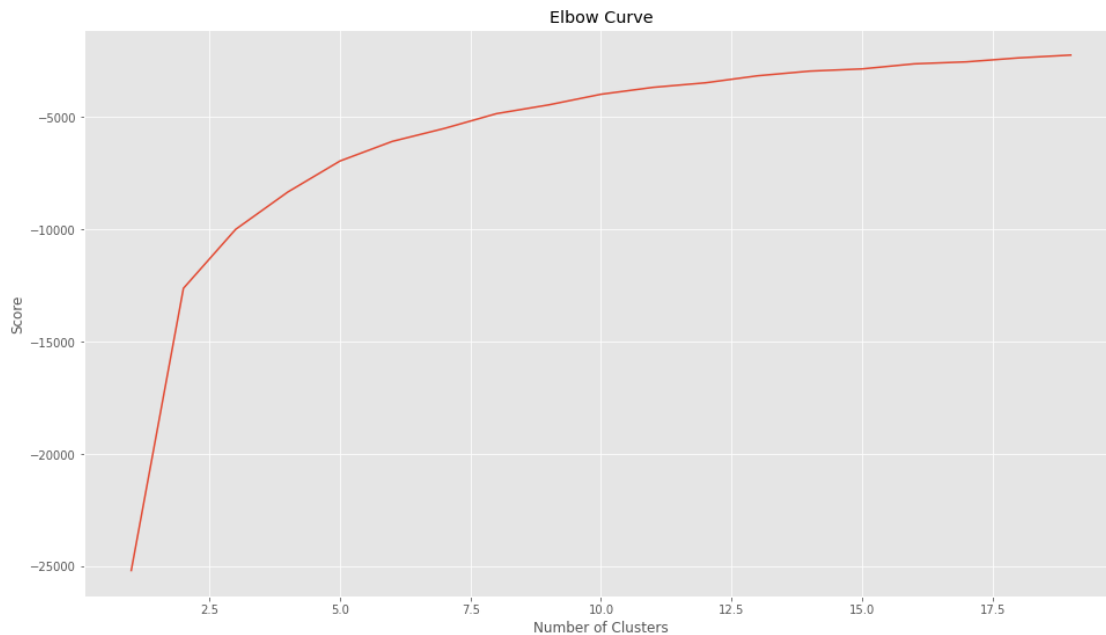


```
[9]: Nc = range(1, 20)
kmeans = [KMeans(n_clusters=i) for i in Nc]
kmeans
score = [kmeans[i].fit(X).score(X) for i in range(len(kmeans))]
```

```

score
plt.plot(Nc,score)
plt.xlabel('Number of Clusters')
plt.ylabel('Score')
plt.title('Elbow Curve')
plt.show()

```



```

[10]: kmeans = KMeans(n_clusters=5).fit(X)
centroids = kmeans.cluster_centers_
print(centroids)

[[35.90241306  47.56828232  33.58748762]
 [59.25320707  31.29770653  15.9896352 ]
 [43.3644665   31.89755772  19.50586767]
 [49.99285115  40.87133549  17.47411785]
 [39.94672753  42.82349753  23.72594859]]

```

```

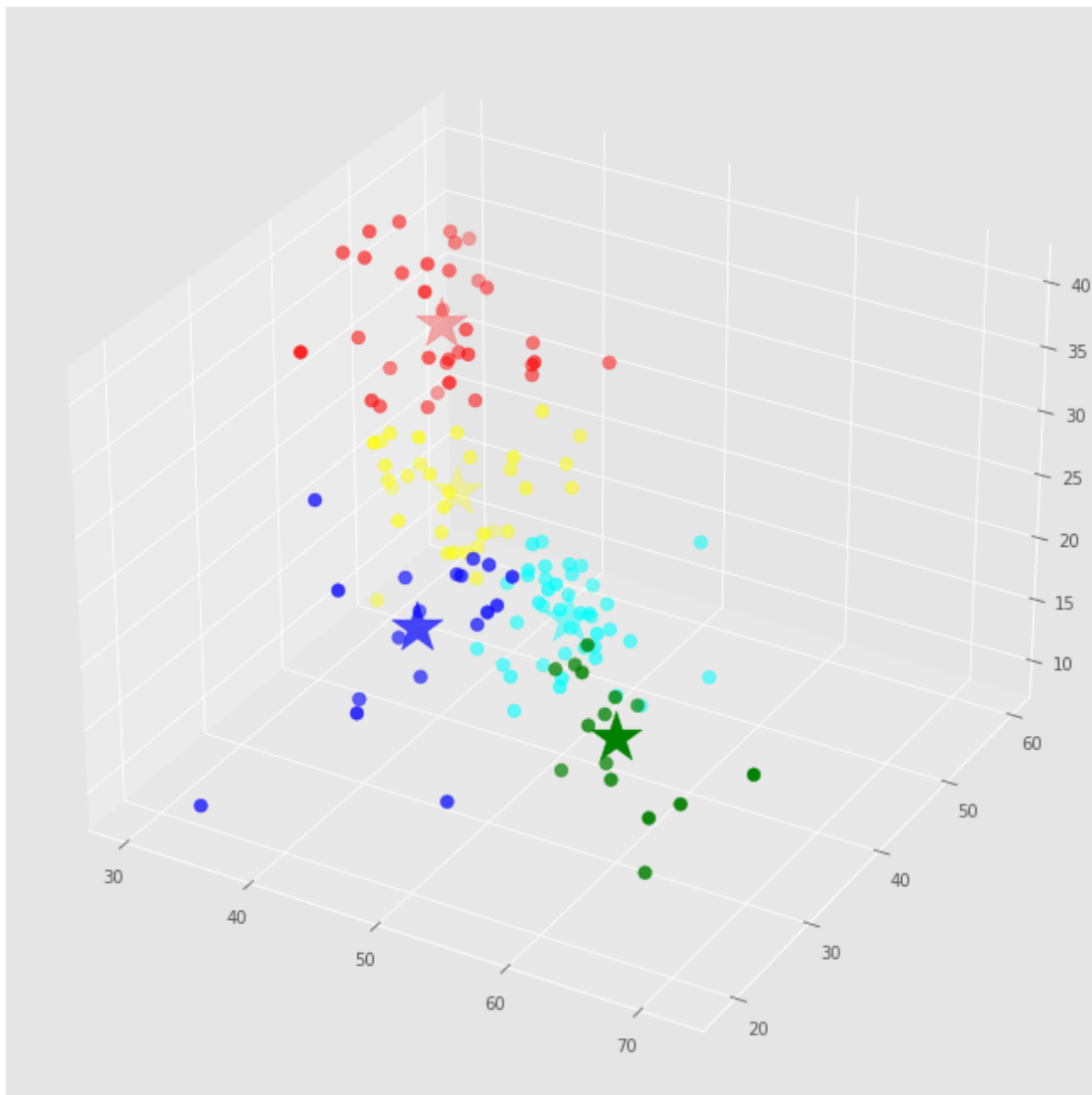
[11]: labels = kmeans.predict(X)
C = kmeans.cluster_centers_
colores=['red','green','blue','cyan','yellow']
asignar=[]
for row in labels:
    asignar.append(colores[row])
fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(X[:, 0], X[:, 1], X[:, 2], c=asignar,s=60)

```

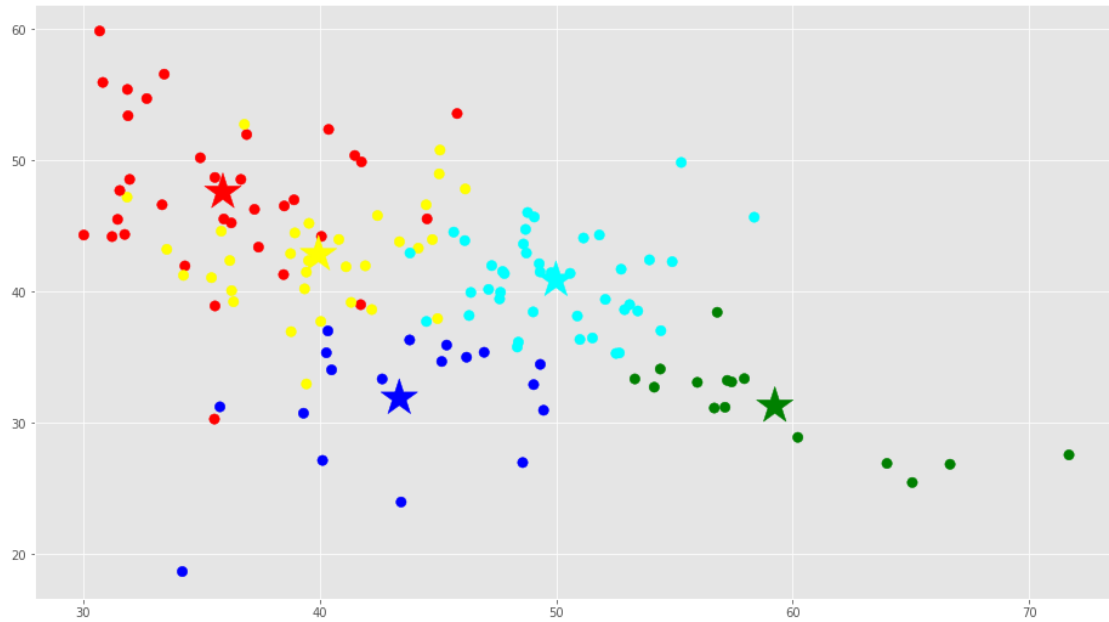
```
ax.scatter(C[:, 0], C[:, 1], C[:, 2], marker='*', c=colores, s=1000)
```

/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/ipykernel_launcher.py:8: MatplotlibDeprecationWarning: Axes3D(fig) adding itself to the figure is deprecated since 3.4. Pass the keyword argument `auto_add_to_figure=False` and use `fig.add_axes(ax)` to suppress this warning. The default value of `auto_add_to_figure` will change to `False` in `mpl3.5` and `True` values will no longer work in 3.6. This is consistent with other Axes classes.

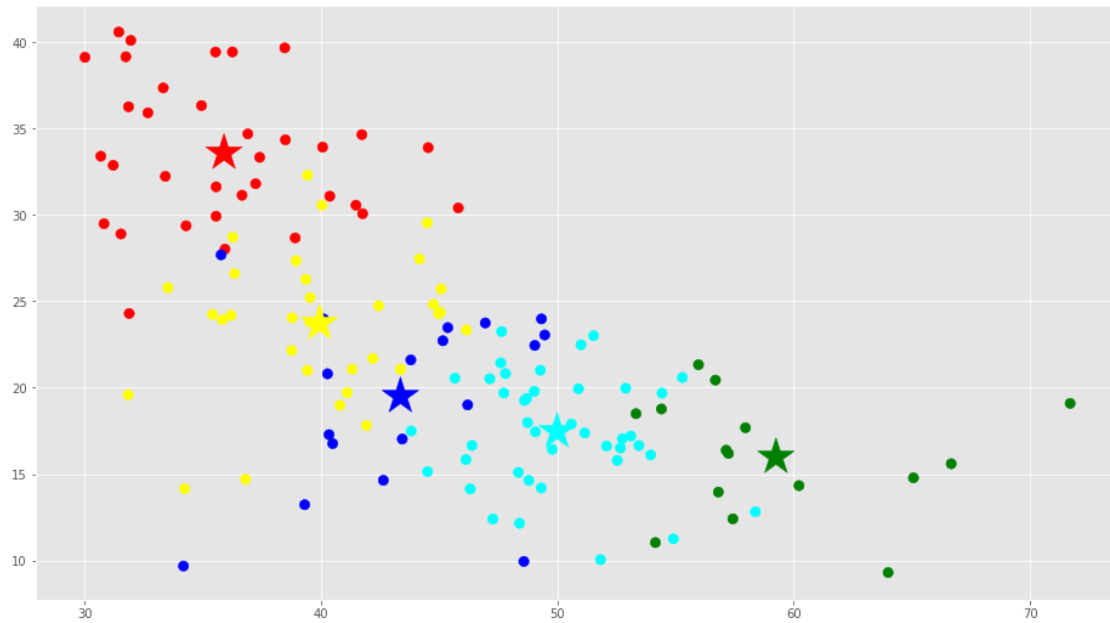
[11]: <mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x7fd13767d7b8>



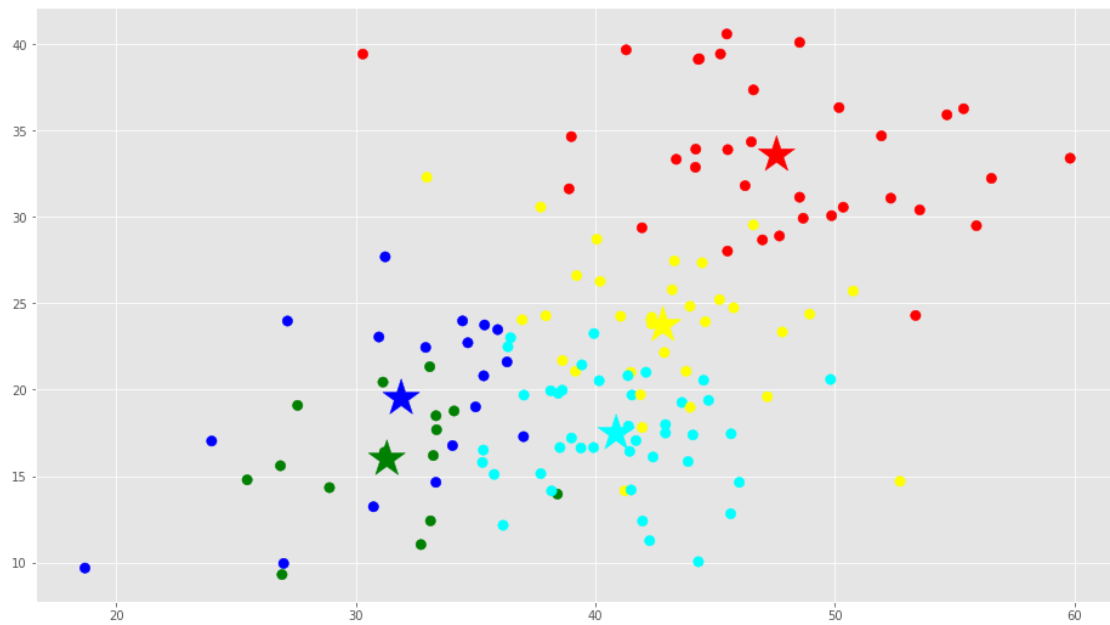
```
[12]: f1 = dataframe['op'].values
      f2 = dataframe['ex'].values
      plt.scatter(f1, f2, c=asignar, s=70)
      plt.scatter(C[:, 0], C[:, 1], marker='*', c=colores, s=1000)
      plt.show()
```



```
[13]: f1 = dataframe['op'].values
      f2 = dataframe['ag'].values
      plt.scatter(f1, f2, c=asignar, s=70)
      plt.scatter(C[:, 0], C[:, 2], marker='*', c=colores, s=1000)
      plt.show()
```

```
[14]: f1 = dataframe['ex'].values
f2 = dataframe['ag'].values
plt.scatter(f1, f2, c=asignar, s=70)
plt.scatter(C[:, 1], C[:, 2], marker='*', c=colores, s=1000)
plt.show()
```



```
[15]: copy = pd.DataFrame()
copy['usuario']=dataframe['usuario'].values
copy['categoria']=dataframe['categoria'].values
copy['label'] = labels;
copy.head()
```

```
[15]:      usuario  categoria  label
0   3gerardpique         7      0
1  aguerosergiokun         7      4
2  albertochicote         4      0
3  AlejandroSanz          2      0
4  alfredocaserol         4      0
```

```
[16]: cantidadGrupo = pd.DataFrame()
cantidadGrupo['color']=colores
cantidadGrupo['cantidad']=copy.groupby('label').size()
cantidadGrupo
```

```
[16]:      color  cantidad
0     red         34
1   green         15
2    blue         18
3    cyan         41
4  yellow         32
```

```
[17]: group_referrer_index = copy['label'] ==0
group_referrals = copy[group_referrer_index]
diversidadGrupo = pd.DataFrame()
diversidadGrupo['categoria']=[0,1,2,3,4,5,6,7,8,9]
diversidadGrupo['cantidad']=group_referrals.groupby('categoria').size()
diversidadGrupo
```

```
[17]:      categoria  cantidad
0           0         NaN
1           1          3.0
2           2          4.0
3           3         NaN
4           4         12.0
5           5          3.0
6           6          1.0
7           7          5.0
8           8          3.0
9           9          3.0
```

```
[18]: closest, _ = pairwise_distances_argmin_min(kmeans.cluster_centers_, X)
closest
```

```
[18]: array([ 98,  82, 123,  21,  64])
```

```
[19]: users=dataframe['usuario'].values  
      for row in closest:  
          print(users[row])
```

```
maria_patino  
JudgeJudy  
SJP  
carmenelectra  
ierrejon
```

```
[20]: X_new = np.array([[45.92,57.74,15.66]]) #davidguetta  
      new_labels = kmeans.predict(X_new)  
      print(new_labels)
```

```
[3]
```

```
[ ]:
```