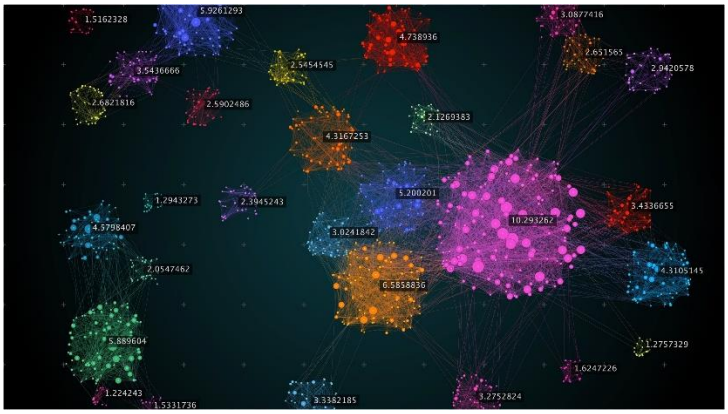
الفصل الثالث

K-means

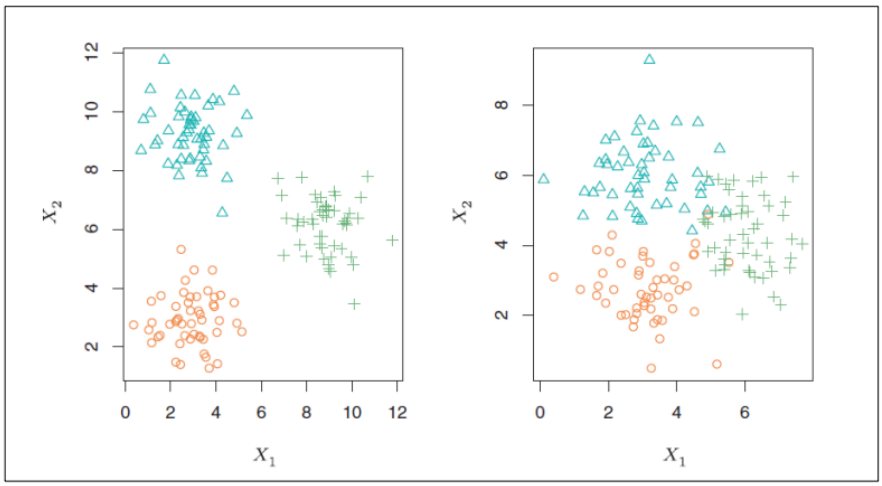
K-Means

تعتبر خوارزمية K-Means من الخوارزميات تعلم الآلة غير الخاضعة للإشراف , و التي تستخدم لحل مسائل العنقدة و التصنيف على شكل عناقيد حيث ان مجموعة البيانات داخل العنقود الواحدة متجانسة لكنها غير متجانسة مع مجموعة البيانات داخل العناقيد الأخرى .



التعلم دون اشارف unsupervised Learning : و فيه تجمع الخوارزمية البيانات المتشابهة الى مجموعات (تقوم الشبكة بإدخال أشعة الدخل المتشابهة مع بعضها تلقائيا بدون استخدام معطيات تدريب و هي نوع من أنواع شبكات التنظيم الذاتي) تعتبر من أهم أنواع العلم بدون اشراف Clustering .

التجميع و العنقدة : هو عملية تجميع الكائنات التي تمتلك سمات متشابهة مع بعضها ضمن مجموعات تدعى العناقيد اذا تختار الآلة أفضل طريقة لفرز الميزات التي تراها مناسبة ,تعد عملية التجميع فعليا عملية تصنيف و لكن المفارقة هنا انها لا تحتوي على أزواج محددة مسبقا .

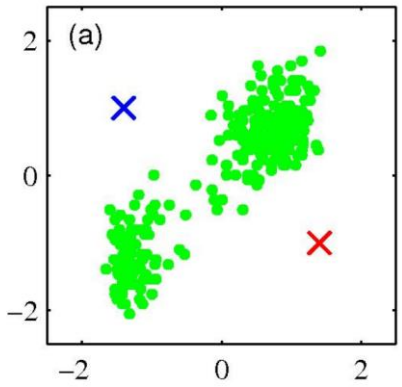


خطوات خوارزمية ال K-means :

تستخدم هذه الخوارزمية لتجميع عدة بيانات اعتمادا على خصائصها الى K تجميع , و تتم عملية التجميع من خلال تقليل المسافات بين البيانات و مركز التجميع centroid cluster.

أما خطوات هذه الخوارزمية فهي :

1- تحديد عدد التجميعات K , و هي تعتبر خطوة تهيئة أولية .
2- تحديد احداثيات مراكز التجميعات centroid عشوائيا لأول مرة , و يكون محسوبا لباقي المرات (متوسط النقاط التي تنتمي للمركز).



- Pick *K* random points as cluster centers (means)

Shown here for *K*=2

3- حساب المسافة بين كل البيانات و بين أول مركز , و يتم استخدام البعد الإقليدي , يعطى البعد الإقليدي d_{ij} بين المثالين I , J بالعلاقة التالية : $d_{ij} =$

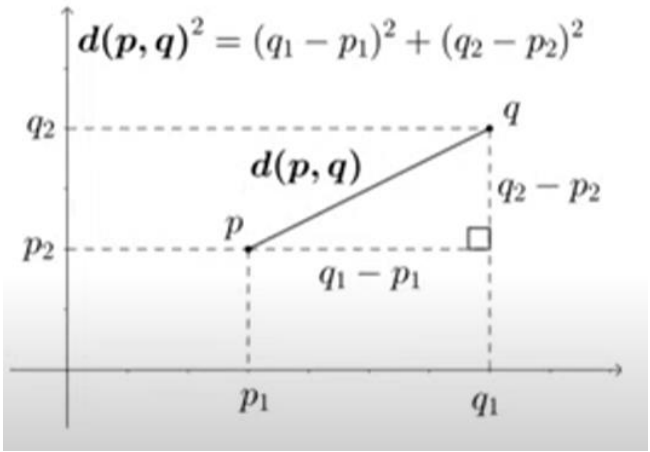
$$\sqrt{\sum_{k=1}^{n}(X_{ik} - X_{jk})^2}$$

حيث n : عدد خصائص المثال .

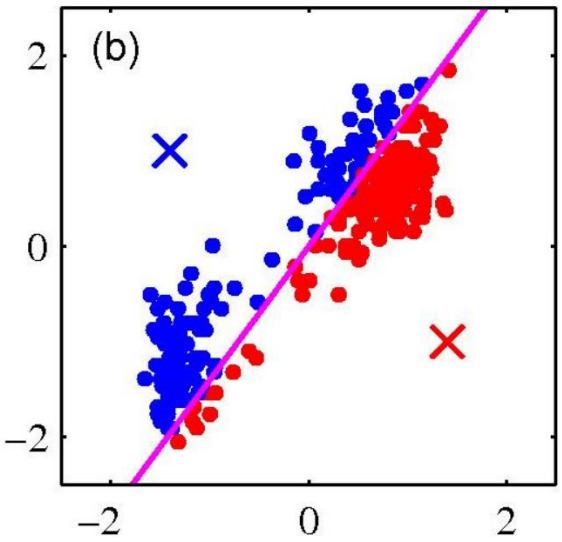
X_{ik} احداثيات الخاصية K للمثال i .

X_{jk} احداثيات الخاصية K للمثال j (يكون في العادة احداثيات المركز).

L2 norm

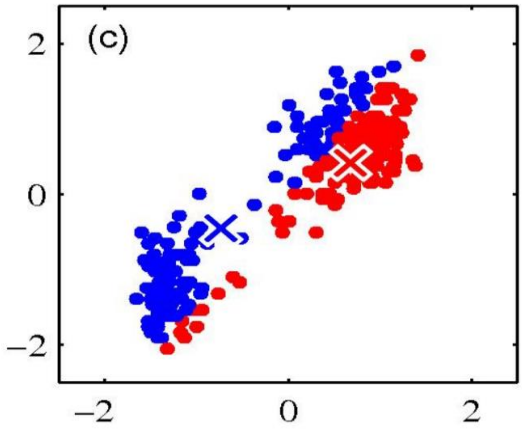


4- نقوم بنسب البيانات حسب قربها من المركز حيث تكون أقصر مسافة بين النقطة و المركز و نكرر العملية لباقي المراكز.



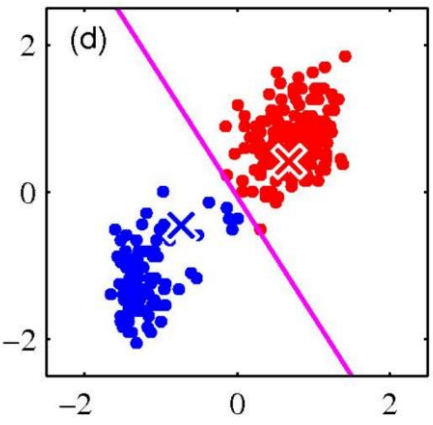
5- نقوم بإيجاد موقع المراكز الجديد و ذلك من خلال حساب مجموع القيم داخل كل مجموعة على عددها فيصبح للمركز موضع جديد .

$$Ci = \frac{1}{|Ni|} \sum xi$$



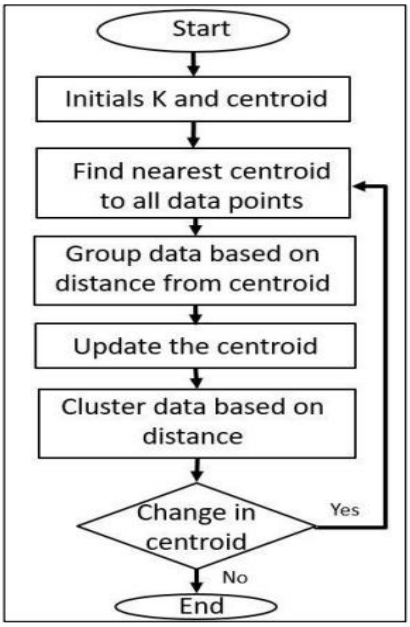
6- تكرار الخطوات من 2 الى 4 حتى حصول الاستقرار (عدم وجود كائنات تنتقل ضمن التجميعات), أو حتى التكرار عدد معين من المرات .

نلاحظ عند تكرار العملية ان تم إضافة نقاط x جديدة الى المجموعة كانت تنتمي سابقة الى مجوعة أخرى لان المسافة بين النقاط اختلفت باختلاف موضع المركز centroid .



يعتمد أداء هذه الخوارزمية على المواقع الأولية لمركز التجميعات centroid , و من المستحسن تنفيذ هذه الخوارزمية عدة مرات مع اختلاف المراكز في كل مرة عن المرات السابقة .

الشكل يظهر المخطط التدفقي لخوارزمية K-means :



كيفية تحديد قيمة K: optimal number of k

في K-means لدينا العناقيد و هي مجموعات – وكل عنقود هو مجموعة – لها نقطة وسطى centroid خاصة به(حيث يتم حسابها من مجموع البيانات على عددها داخل المجموعة الواحدة) .

أنه مجموع المسافة المربعة بين متوسط النقطة (تسمى Centroid) وكل نقطة من المجموعة.

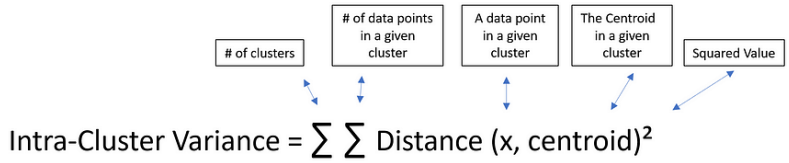
كلما كانت القيمة أصغر ، كان التجميع أفضل. (within-cluster sums of squares)

$$\text{WCSS} = \sum_{C_k}^{C_n} \Big(\sum_{d_i in C_i}^{d_m} distance(d_i, C_k)^2 \Big)$$

Where,

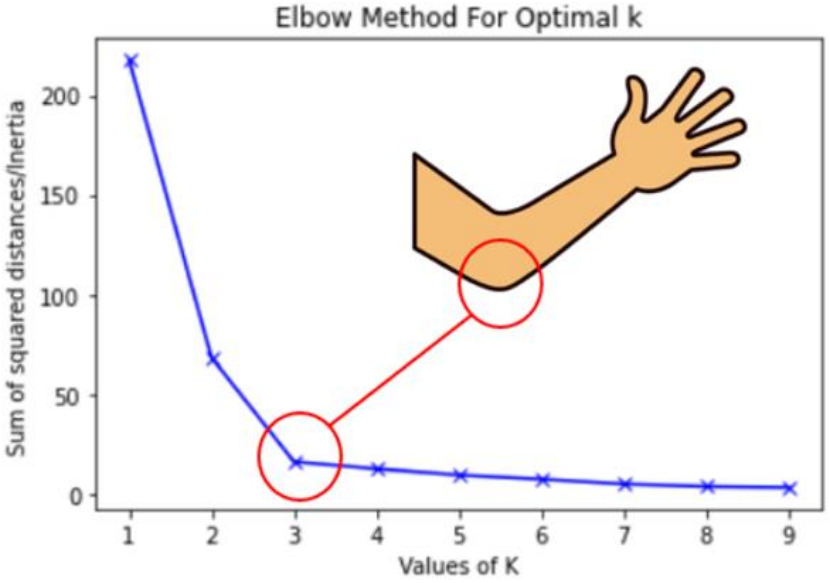
C is the cluster centroids and d is the data point in each Cluster.

المعادلة لتوضيح و هي عبارة عن دالة الخطأ التربيعي : (Intra-cluster variance)



حيث يقوم بحساب مجموع التباين (اختلاف البيانات مع المتوسط أي المركز C داخل كل مجموعة)و نكرر العملية على كل المجموعات و نقوم بجمعها .

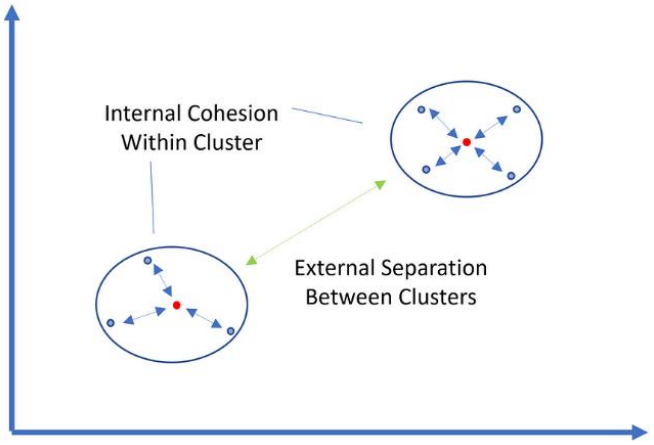
نعلم أنه مع زيادة عدد العناقيد , تستمر هذه القيمة في التناقص و لكن اذا قمت برسم النتيجة , فقد ترى أن مجموع المسافة المربعة يتناقص بشكل حاد الى قيمة معينة من K, ثم ببطء أكثر بعد ذلك . هنا , يمكننا إيجاد العدد الأمثل لعدد العناقيد .



Line plot between K and inertia

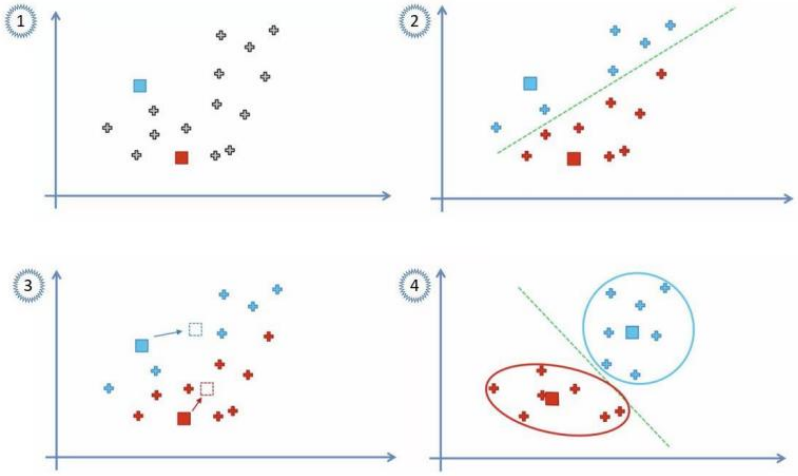
مفهوم BCSS **Between-cluster sums of squares** : هو المسافة بين كل مركز مجموعة و أخرى حيث يجب أن يكون كبيرا حتى لا تدخل النقاط المجموعة في مجموعة أخرى .

حيث تقوم فكرة ال Kmeans بتقليل المسافة بين النقاط و المركز داخل المجموعة و تكبير المسافة بين كل مركز و مركز



مثال مبسط يشرح طريقة عملها :

بفرض أننا نريد تقسيم النقاط التالية الى مجموعات حسب قربها من بعضها كما في الشكل :



- مزايا و مساوئ K-means :
- المزايا

1- ذات فعالية عالية .
2- سهلة التنفيذ
3- تتعامل مع القيم المستمرة و القيم المتقطعة (الاسمية)
4- تتكيف بسهولة مع البيانات الجديدة

- المساوئ :

1- حساسة للحالة الأولية , يؤدي اختيار حالات أولية متعددة الى إعطاء نتائج مختلفة لتجميعات.
2- شكل التجمع دائري لأنه يعتمد على حساب المسافة
3- عنقدة القيم المتطرفة. حيث يمكن سحب و ابعاد القيم الوسطى centroids بواسطة القيم المتطرفة أو قد تحصل القيم المتطرفة على مجموعة خاصة بها بدلا من تجاهلها لذلك يجب وضع إزالة القيم المتطرفة أو قصها قبل العنقدة في الاعتبار .

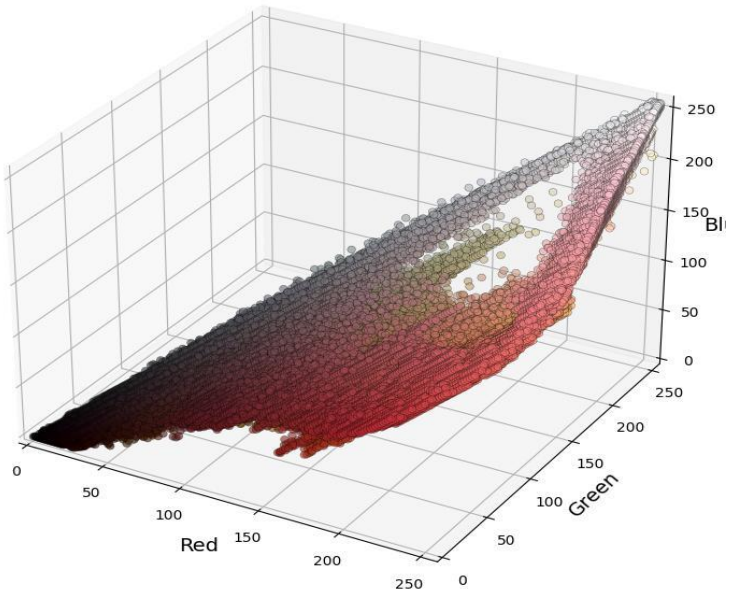
التطبيق العملي لل Kmeans :

ادخال NumPy, pandas, matplotlib, OpenCV, Sklearn.cluster المكتبات

```
from collections import Counter
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import numpy as np
import cv2
img_name = "/content/real_test_apple.jpg"
raw_img = cv2.imread(img_name)
raw_img = cv2.cvtColor(raw_img,cv2.COLOR_BGR2RGB)
img = cv2.resize(raw_img,(416,416),interpolation = cv2.INTER_AREA
)
img
```

حيث نقوم بتحميل و بتحويل الصورة من BGR الى RGB وثم نقوم بتصغير الصورة و طباعة array حيث تكون 3d مصفوفة من ثلاثة أبعاد نقوم بتحويلها الى 2d حيث اذا كانت الصورة (m,n,3) نحوله (m*n,3)

```
array([[[72, 72, 72],
        [76, 76, 76],
        [52, 52, 52],
        ...,
        [79, 73, 73],
        [74, 68, 68],
        [63, 55, 56]],

       [[64, 64, 64],
        [57, 57, 57],
        [61, 61, 61],
        ...,
        [42, 36, 36],
        [44, 38, 38],
        [42, 35, 35]],

       [[82, 82, 82],
        [85, 85, 85],
        [93, 93, 93],
        ...,
        [33, 27, 27],
        [25, 19, 19],
        [40, 33, 33]],

       ...,

       [[73, 73, 73],
        [77, 77, 77],
        [84, 84, 84],
        ...,
        [88, 86, 87],
        [71, 69, 70],
        [91, 90, 91]],

       [[74, 74, 74],
        [59, 59, 59],
        [98, 98, 98],
        ...,
        [67, 65, 66],
        [68, 66, 67],
        [66, 64, 65]],
```




```
img = img.reshape(img.shape[0]*img.shape[1],3)
img
```

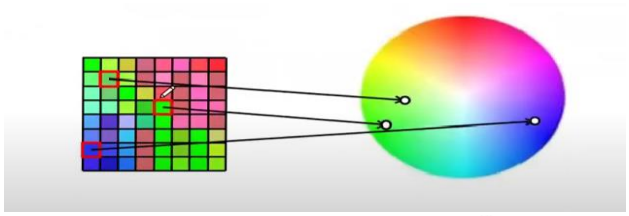
```
array([[72, 72, 72],
       [76, 76, 76],
       [52, 52, 52],
       ...,
       [94, 92, 93],
       [61, 59, 60],
       [81, 79, 80]], dtype=uint8)
```

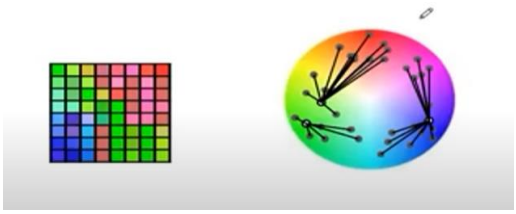
```
#https://scikitlearn.org/stable/modules/generated/sklearn.cluster
.KMeans.html
clf = KMeans(n_clusters=5)
# (fit_predict) //Compute cluster centers and predict cluster ind
ex for each sample and the funcation return
# Index of the cluster each sample belongs to.
#labelsndarray of shape (n_samples,)
color_labels = clf.fit_predict(img)
print("labels are  = ")
print(color_labels)
#(clf.cluster_centers_) =  Coordinates of cluster centers
center_colors = clf.cluster_centers_
print("centers  are  = ")
print(center_colors)
```

حيث نقوم بحساب cluster و fun تقوم بارجاع labels لكل بكسل `fit_predict`

بعدها نقوم بإسناد مواقع المراكز cluster coordinates `cluster_centers_`

```
/usr/local/lib/python3.10/dist-packages/sklearn/cl
  warnings.warn(
labels are  =
[3 3 3 ... 4 3 3]
centers  are  =
[[194.8828266  171.27224767 170.62182424]
 [176.23071195  69.92525345  62.29302938]
 [228.88952568 111.86870765  96.86340109]
 [ 68.45251322  59.50250919  58.447468  ]
 [103.45008789 102.04662781 101.10580689]]
```





```
#https://docs.python.org/3/library/collections.html#collections.C
ounter
count = Counter(color_labels)
count
```

```
Counter({3: 37274, 4: 32417, 2: 61083, 0: 7203, 1: 35079})
```

العداد count هو فئة فرعية من dict لعد العناصر, حيث اسندنا كل بكسل label للمركز مع عدد البيانات التي داخل كل مجموعة .

```
ordered_colors = [center_colors[i] for i in count.keys()]
print(ordered_colors)
hex_colors = [rgb_to_hex(ordered_colors[i]) for i in count.keys()
]
print(hex_colors)
count.keys()
```

قمنا باسناد قيمة مفاتيح القاموسcount الى قيم rgb أي ابعاد centers و من ثم قيمة hex لون

```
[array([68.45251322, 59.50250919, 58.447468  ]), array([103.45008789, 102.04662781, 101.10580689]), array([228.8
['#c2abaa', '#b0453e', '#e46f60', '#443b3a', '#676665']
dict_keys([3, 4, 2, 0, 1])
```

```
labels=list(color_labels)
percent=[]
percent_1=[]
for i in range(len(center_colors)):
  j=labels.count(i)
  j=j/(len(labels))
  percent.append(round((j*100),1))
  percent_1.append(str(round((j*100),1))+'%')
```

حساب نسبة عدد البيانات (الألون) داخل كل مجموعة

```
res = {}
for key in hex_colors:
    for value in percent:

        res[key] = value
        percent.remove(value)
        break

# Printing resultant dictionary
print("Resultant dictionary is : " + str(res))
```

```
Resultant dictionary is : {'#c2abaa': 4.2, '#b0453e': 20.3, '#e46f60': 35.3, '#443b3a': 21.5, '#676665': 18.7}
```

قمنا باسناد كل قيمة في hex للمراكز الى المفتاح key ضمن القاموس أي نسبة عدد البينات التي تحتويها كل مجموعة .

```
key_max = max(res ,key = lambda x:res[x])
print(key_max)
```

```
#e46f60
```

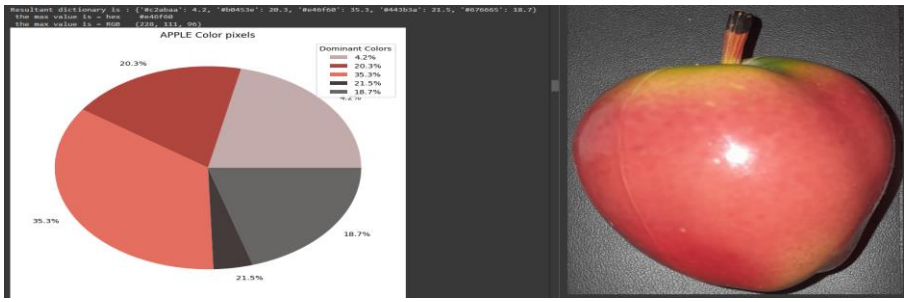
دالة max لاعطاء المفتاح(اللون) المقابل لأكبر نسبة من عدد البيانات

```
def hex_to_rgb(hex):
  rgb = []
  for i in (0, 2, 4):
    decimal = int(hex[i:i+2], 16)
    rgb.append(decimal)

  return tuple(rgb)
```

دالة تحويل hex value الى قيمة RGB .

```
fig =plt.figure(figsize=(12,8))
fig.patch.set_facecolor('white')

plt.pie(count.values(),labels=percent_1,colors = hex_colors)
plt.legend(title = "Dominant Colors")
plt.title("APPLE Color pixels")
print("Resultant dictionary is : " + str(res))
x = key_max.split("#")
print(" the max value is = hex   ", key_max,"\n the max value is
= RGB  ",hex_to_rgb(x[1]))
```

لطباعة pie plot يحتوي على النسب و اللون الأعظمي في الصورة (اللون الأعظمي أو المهيمن جاء من center of cluster حيث تكون قيمته هي القيمة المتوسطة لجميع الألوان داخل المجموعة).



K-Means Clustering: Explain It To Me Like I'm 10 | by Shreya Rao | Towards Data Science

K-Means Clustering Explained (neptune.ai)

Understanding K-means Clustering in Machine Learning | by Education Ecosystem (LEDU) | Towards Data Science

sklearn.cluster.KMeans — scikit-learn 1.2.2 documentation

Dominant colors in an image using k-means clustering | by Shivam Thakkar | buZZrobot | Medium

How-To: OpenCV and Python K-Means Color Clustering (pyimagesearch.com)

Finding Dominant Color in the Artistic Painting using Data Mining Technique (International Research Journal of Engineering and Technology (IRJET))

Color Separation in an Image using KMeans Clustering using Python | by Sai Durga Kamesh Kota | Analytics Vidhya | Medium