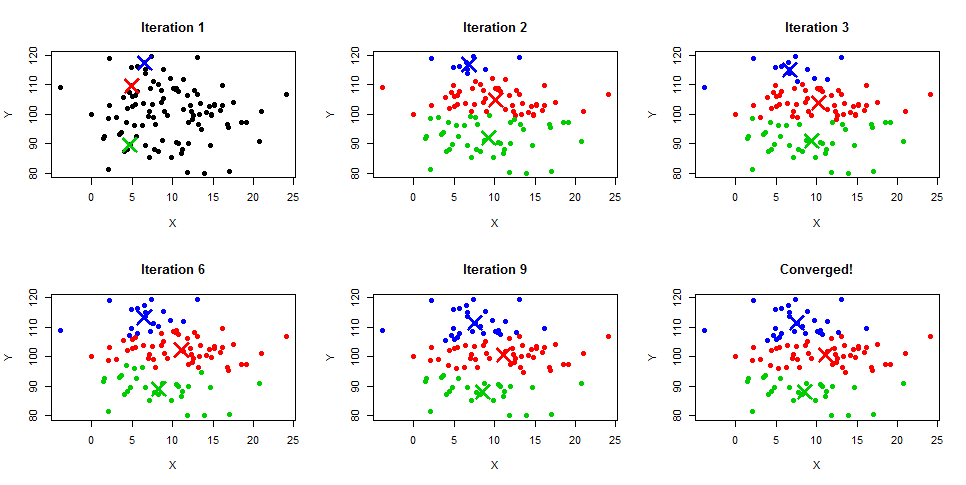
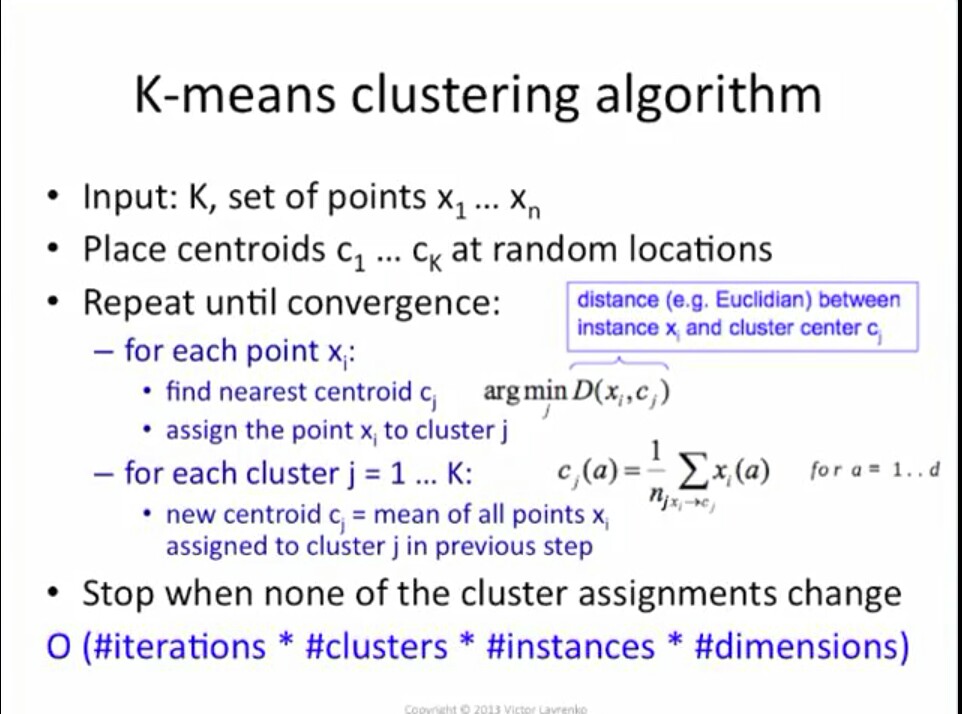
**K-Means Clustering**

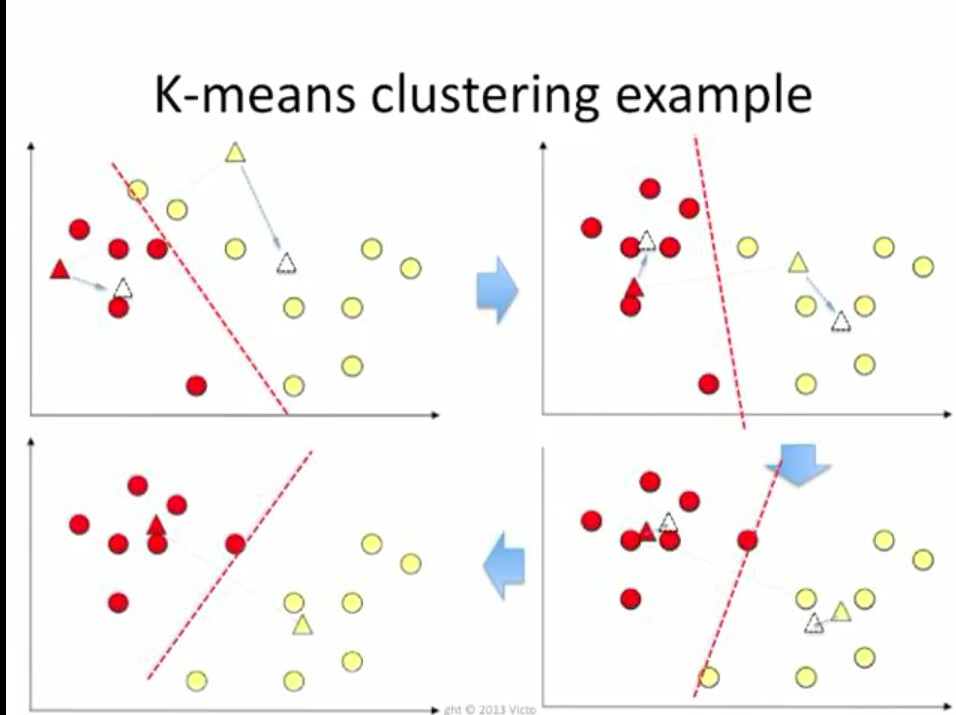
**Short note about K –Means Clustering**:

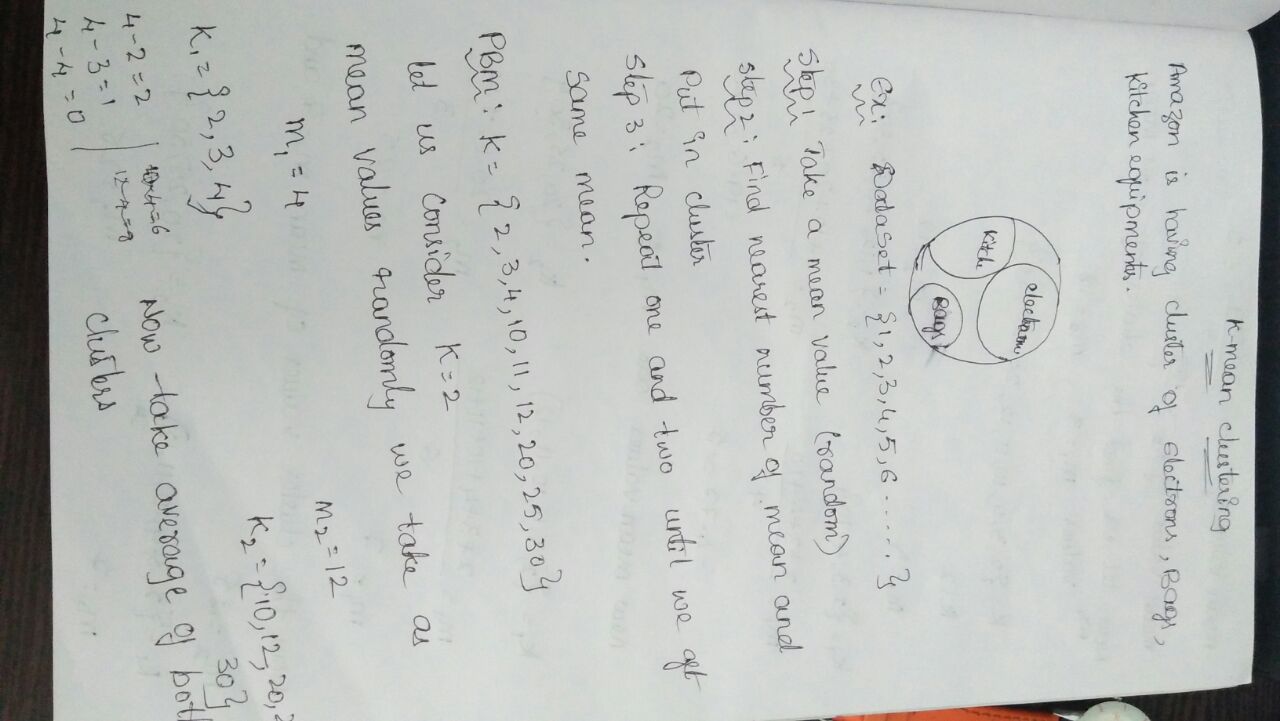
The K-means algorithm starts by placing K points (centroids) at random locations in space. We then perform the following steps iteratively: (1) for each instance, we assign it to a cluster with the nearest centroid, and (2) we move each centroid to the mean of the instances assigned to it. The algorithm continues until no instances change cluster membership.

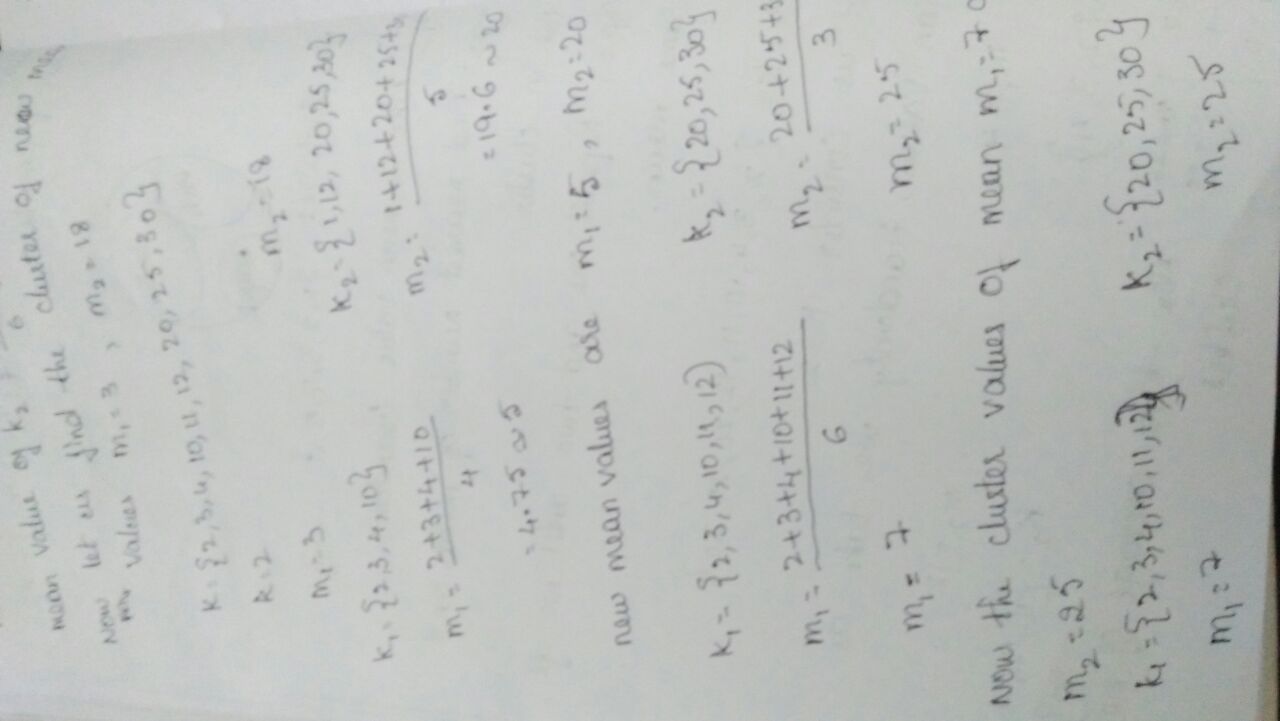
k-means clustering aims to [partition](https://en.wikipedia.org/wiki/Partition_of_a_set) n observations into k clusters in which each observation belongs to the [cluster](https://en.wikipedia.org/wiki/Cluster_(statistics)) with the nearest [mean](https://en.wikipedia.org/wiki/Mean), serving as a prototype of the cluster. This results in a partitioning of the data space into [Voronoi cells](https://en.wikipedia.org/wiki/Voronoi_cell).











Hence the mean values are same ,we consider the clustering values as

k1=(2,3,4,10,11,12},k2={20,25,30}.

**Use Cases-**

* Market Price and cost modeling.
* Customer Segmentation.
* Insurance claim and fraud Detection.
* Elgibility to apply for credit card
* Chances of getting selected in Interview
* Chances of apply use cases without error.
* To mention the Population Growth.
* To check the Credit Risk in banks.
* Finding the transactions happened in ATM per day.
* Finding the changes in Pollution

Python-

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import sklearn

from sklearn.cluster import KMeans

from sklearn import preprocessing

from sklearn.preprocessing import scale

#

## Importing the dataset

dataset = pd.read\_csv('C:\\Users\\Rama\\Desktop\\Social\_Network\_Ads.csv')

X = dataset.iloc[:, [2,4]].values

#y = dataset.iloc[:, 4].values

#data management

data\_clean=dataset.dropna()

#subset clustering variables

cluster=data\_clean[['Age','Purchased']]

#standardise clustering variables to have mean=0,standard deviation=1

clustervar=cluster.copy()

clustervar['Age']=preprocessing.scale(clustervar['Age'].astype('float64'))

#clustervar['EstimatedSalary']=preprocessing.scale(clustervar['EstimatedSalary'].astype('float64'))

clustervar['Purchased']=preprocessing.scale(clustervar['Purchased'].astype('float64'))

# Splitting the dataset into the Training set and Test set

from sklearn.cross\_validation import train\_test\_split

clus\_train,clus\_test=train\_test\_split(clustervar,test\_size=.3,random\_state=123)

#k-mean cluster analysisfor 1-9 clusters

from scipy.spatial.distance import cdist

clusters=range(1,10)

meandist=[]

for k in clusters:

model=KMeans(n\_clusters=8)

model.fit(clus\_train)

clusassign=model.predict(clus\_train)

meandist.append(sum(np.min(cdist(clus\_train,model.cluster\_centers\_,'euclidean'),axis=1))

/ clus\_train.shape[0])

clf = KMeans(n\_clusters=3)

clf.fit(clus\_train)

centroids = clf.cluster\_centers\_

labels = clf.labels\_

colors = ["g.","r.","y."]

for i in range(len(clus\_train)):

plt.plot(X[i][0], X[i][1], colors[labels[i]], markersize = 10)

plt.scatter(centroids[:, 0],centroids[:, 1], marker = "x", s=150, linewidths = 5, zorder = 10)

plt.show()

