12) Mean shift

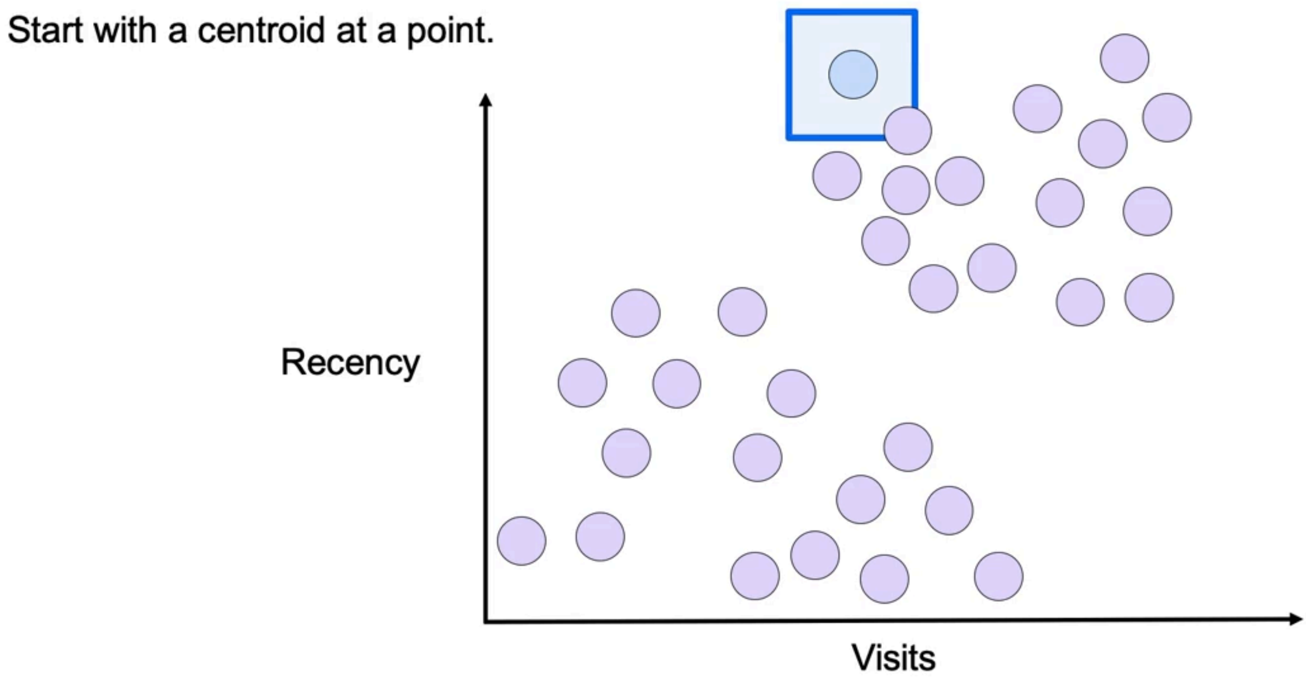
- Similar to k-means, as in we will be partitioning our points closest to the nearest cluster centroid
- In meanshift, the centroid is the point of highest local density (like k-means, where the centroid was the mean of the clutser)
- Algorithm ends when all points are assigned to a cluster
- We calculate local density by evaluating weighted mean around each point
- **Strengths:**
  - Model free, ie does not assume number or shape of clusters
  - Can use just one parameter (ie window size aka bandwidth)
  - Robust to outliers
- **Weakness**:
  - Result heavily depends on the window size aka bandwidth
  - Selection of window size is not easy :(
  - Can be slow to implement

How it works

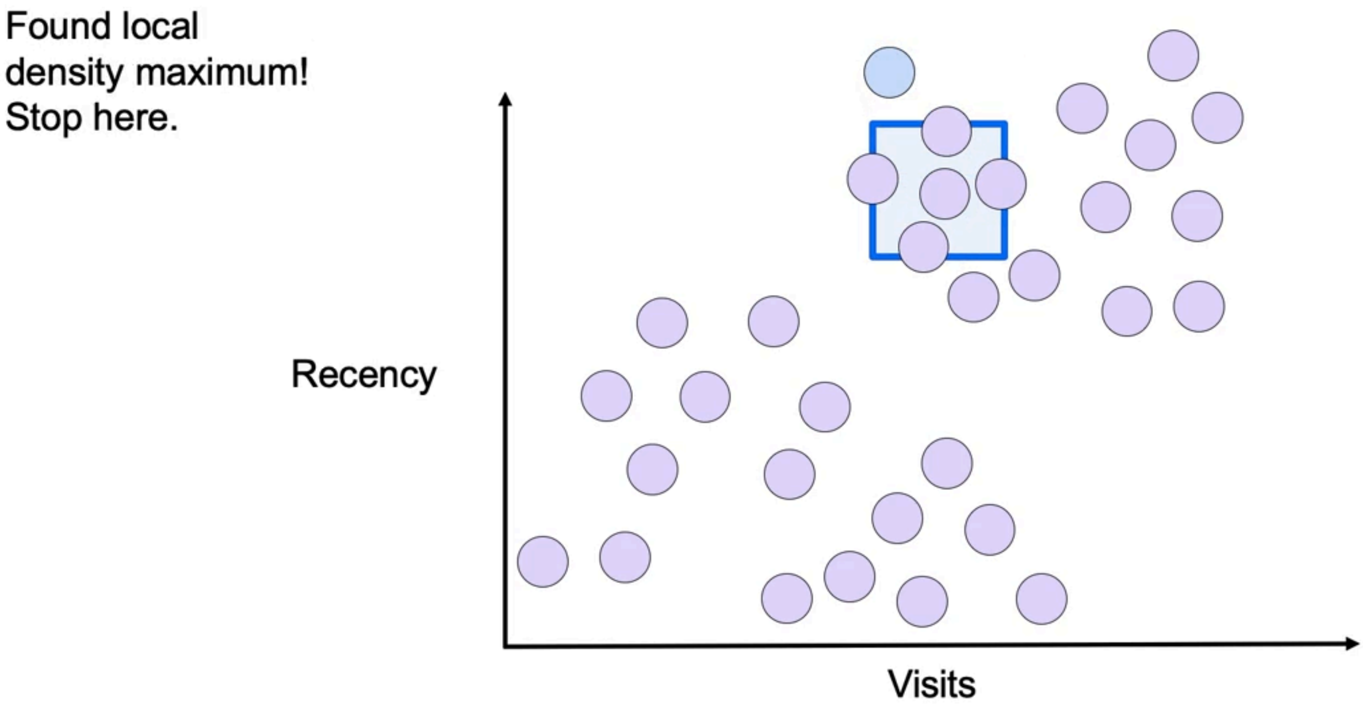
1. We choose a point and a window
2. We calculate weighted mean in that window
3. We shift the centroid of the window to the new mean
4. We keep doing step 2 and 3 until convergence (no shift), ie until local density maximum ("mode") is reached
5. Repeat steps 1-4 for all data points. Data points that lead to the same mode are grouped to the same cluster

Lets visualise how this actually works in practice:

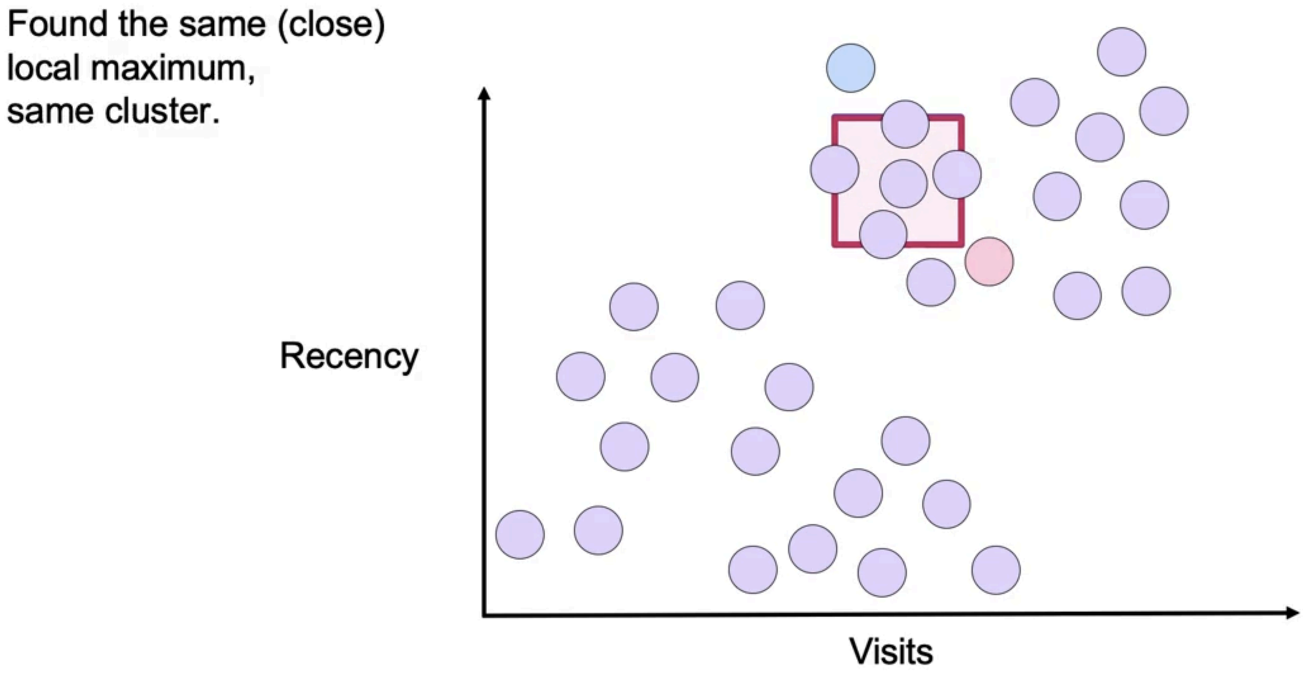
1. Start with a centroid at a point



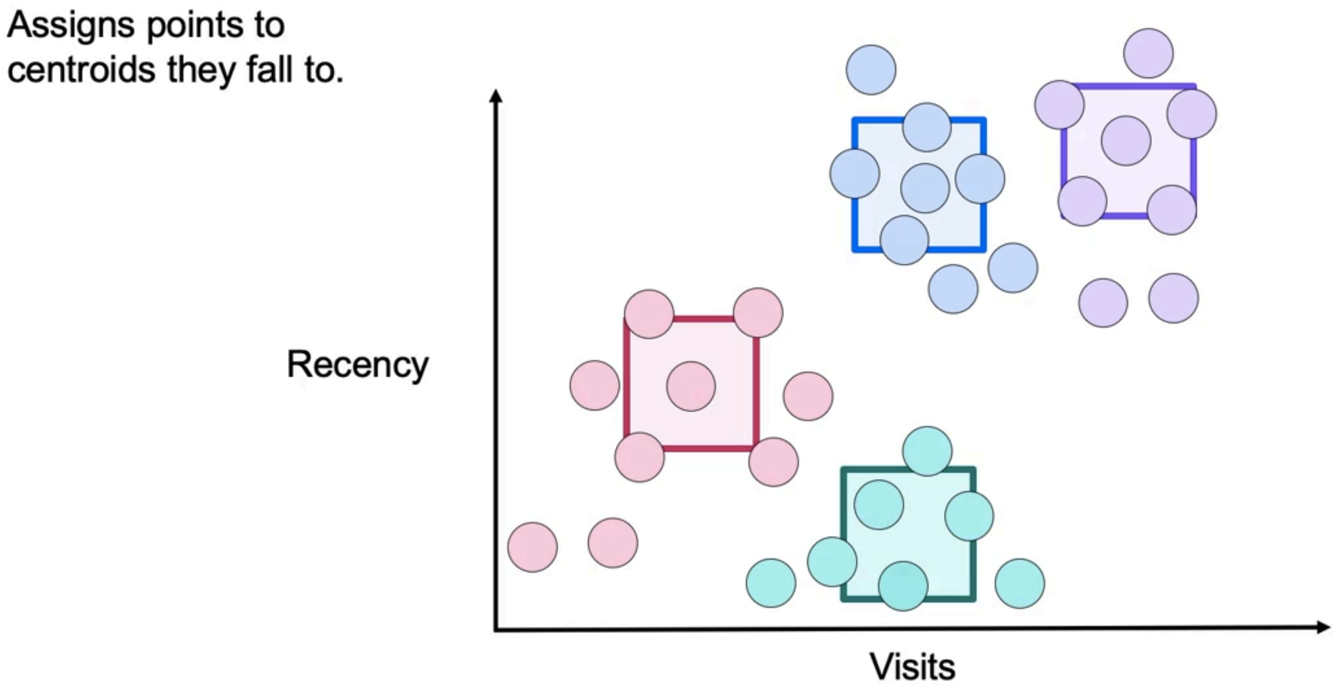
2/3/4) Sample local density and follow gradient towards the denser direction until local density maximum is reached



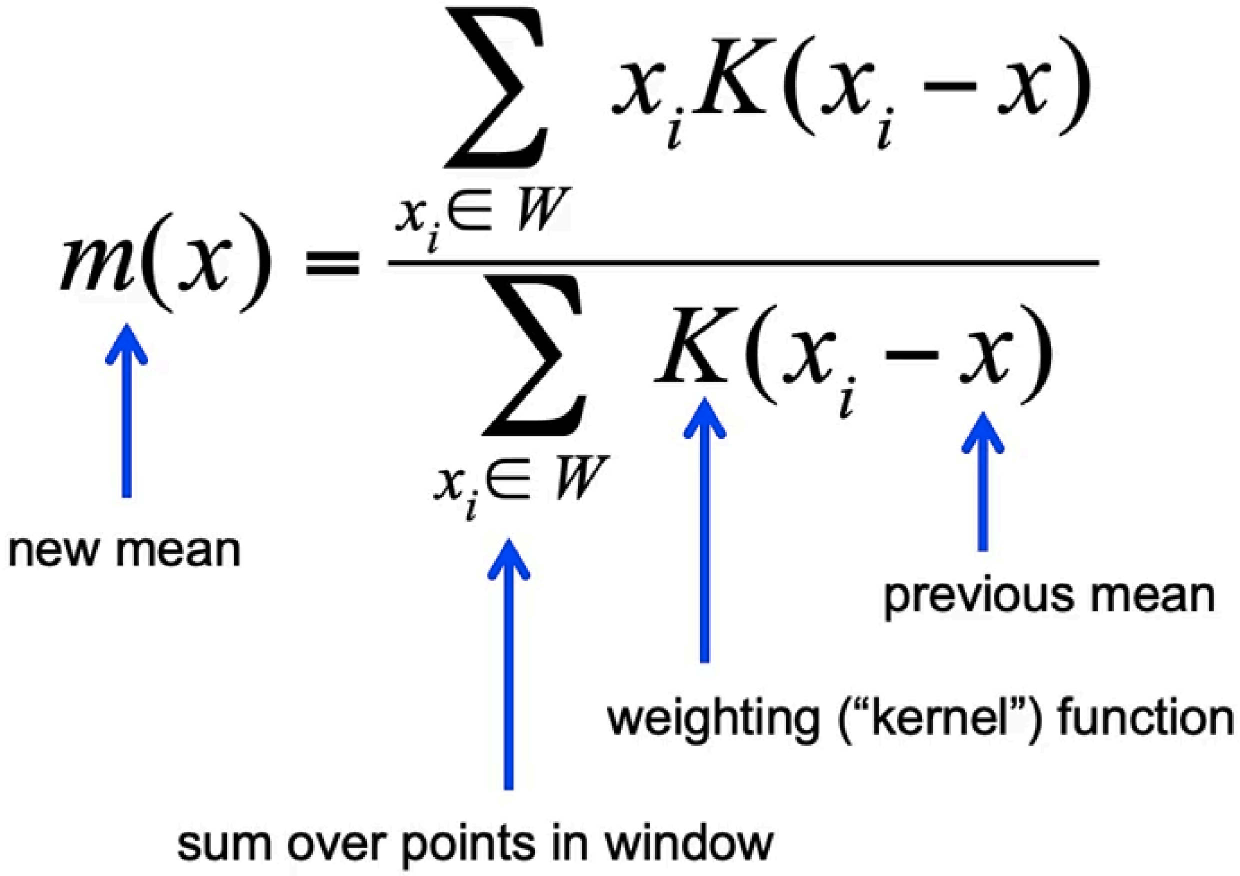
5. Repeat the steps for other points



So in our dataset, we end up with 4 local maximas like so:



Weighted mean



Code

```python
from sklearn.cluster import MeanShift

model = MeanShift(bandwidth=None)
model.fit(X1)
y_pred = model.predict(X2)
```