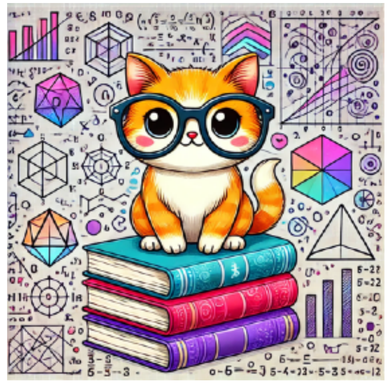
Recall K-Means Algorithm



metric that assesses the performance of K-means (smaller values better)

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|^2$$

actual data point n

`minimizes` J

1. Choose **k** random points as cluster centers
2. For each data point, assign it the cluster whose centroid is the closest
3. Using these assignments, recalculate the centers
4. Reiterate from step (2) until **convergence**:
   - cluster membership does not change
   - center only changes very very little

`hard assignmen`

$$\frac{dJ}{d\mu_k} = 2 \sum_{n=1} r_{nk}(x_n - \mu_k) = 0 \Longrightarrow \mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}} = \frac{1}{N_k} \sum_n r_{nk} x_n$$

`optimal value for` μ_k `minimizing our loss is the mean of all data points in that cluster`

GMM: EM Algorithm

1. Choose **k** random points to be cluster centers (or estimate using k-means...)
2. For each data point, calculate the **probability** of belonging to each cluster
3. Using these probability weights, recalculate the **means + variances** (and weights)
4. Repeat 2 and 3 until **distributions converge**