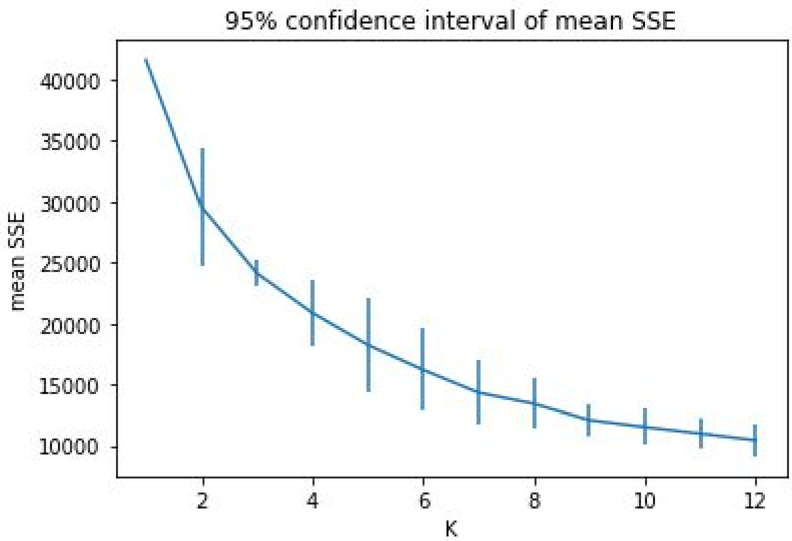
1.

(a)

k μ_k σ_k

{1: [41580.0000000011, 0.0],
2: [29536.67079071056, 2404.7619280043386],
3: [24112.013576830817, 550.3079193379692],
4: [20874.71561582976, 1339.2321869416166],
5: [18248.271369616232, 1915.8533732411145],
6: [16199.653071825407, 1686.9206236074237],
7: [14340.020933670274, 1324.1028554747916],
8: [13434.917124498412, 1019.7208566830121],
9: [12096.09477462503, 676.5987872108416],
10: [11521.567618781799, 740.8779972626403],
11: [10994.141726941036, 606.1116477030495],
12: [10447.593079708075, 655.5640900532965]}



(b)

K	μ	μ-2σ	μ+2σ
1	41580	41580	41580
2	29536.67	24727.1	34346.19
3	24112.01	23011.4	25212.63
4	20874.72	18196.3	23553.18
5	18248.27	14416.6	22079.98
6	16199.65	12825.8	19573.49
7	14340.02	11691.8	16988.23
8	13434.92	11395.5	15474.36
9	12096.09	10742.9	13449.29
10	11521.57	10039.8	13003.32
11	10994.14	9781.92	12206.37
12	10447.59	9136.46	11758.72

（c）

As k increase and approaches the total number of examples N, SSE will decrease and approach 0. In this way, we can not choose the k with the least SSE. We can not find the optimal k since k=N has the least SSE, but it is not what we want.

(d)

we could use scatter criteria. Instead of only monitoring error within clustering, we could also calculate the between cluster scatter matrix. we hope that the scatter criterion $\frac{tr(S_B)}{tr(S_W)}$ is high and we need to choose the smallest k that this criterion is not increasing fast and tend to plateau.

2.

(a) $m_1 = \frac{1}{3}\left(\begin{pmatrix}1\\1\end{pmatrix} + \begin{pmatrix}2\\2\end{pmatrix} + \begin{pmatrix}3\\3\end{pmatrix}\right) = \begin{pmatrix}2\\2\end{pmatrix}$

$m_2 = \frac{1}{5}\left(\begin{pmatrix}5\\2\end{pmatrix} + \begin{pmatrix}6\\2\end{pmatrix} + \begin{pmatrix}7\\2\end{pmatrix} + \begin{pmatrix}8\\2\end{pmatrix} + \begin{pmatrix}9\\2\end{pmatrix}\right) = \begin{pmatrix}7\\2\end{pmatrix}$

(b)

$m = \frac{1}{8}\left(3 \cdot \begin{pmatrix}2\\2\end{pmatrix} + 5 \cdot \begin{pmatrix}7\\2\end{pmatrix}\right) = \begin{pmatrix}\frac{41}{8}\\2\end{pmatrix}$

(c) for C_1:

$S_1 = \begin{pmatrix}1-2\\1-2\end{pmatrix}(1-2, 1-2) + \begin{pmatrix}0\\0\end{pmatrix}(0\ 0) + \begin{pmatrix}1\\1\end{pmatrix}(1\ 1)$

$= \begin{pmatrix}2 & 2\\2 & 2\end{pmatrix}$

for C_2:

$S_2 = \begin{pmatrix}-2\\0\end{pmatrix}(-2\ 0) + \begin{pmatrix}-1\\0\end{pmatrix}(-1\ 0) + \begin{pmatrix}0\\0\end{pmatrix}(0\ 0) + \begin{pmatrix}1\\0\end{pmatrix}(1\ 0) + \begin{pmatrix}2\\0\end{pmatrix}(2\ 0)$

$= \begin{pmatrix}10 & 0\\0 & 0\end{pmatrix}$

(d) $S_W = S_1 + S_2$

$= \begin{pmatrix}12 & 2\\2 & 2\end{pmatrix}$

(e)

$S_B = 3 \cdot \begin{pmatrix}-\frac{25}{8}\\0\end{pmatrix}(-\frac{25}{8}\ 0) + 5 \cdot \begin{pmatrix}\frac{15}{8}\\0\end{pmatrix}(\frac{15}{8}\ 0)$

$= \begin{pmatrix}\frac{375}{8} & 0\\0 & 0\end{pmatrix}$

(f) $tr(S_B) = \frac{375}{8}$

$tr(S_W) = 14$

$\frac{tr(S_B)}{tr(S_W)} = \frac{375}{112}$

3.

3. a) $(0.6)^3 + C_3^2 \cdot (0.6)^2 \cdot 0.4 = 0.648$

C_3 accuracy: 64.8%

b) $(0.6)^5 + C_5^4 (0.6)^4 \cdot 0.4 + C_5^3 (0.6)^3 \cdot (0.4)^2 = 0.68256$

C_5 accuracy: 68.256%

(C)

```
from scipy.special import comb
total = 0
for i in range(13,26):
    total += comb(25,i)*(0.6**i)*(0.4**(25-i))
print(total)
0.846232231024237
```

c) C_{25} accuracy: 84.623%

(d) each student may not totally worked independently, and each model may not all reach 60% accuracy

(e) C_{25} accuracy: 30.632% worse than guessing

(e)

```
from scipy.special import comb
total1 = 0
for i in range(13,26):
    total1 += comb(25,i)*(0.45**i)*(0.55**(25-i))
print(total1)
0.30632396592448247
```

4.

Instance	True class	Predicted ~~cla~~ class	
1	P	P	TP
2	N	P	FP
3	P	P	TP
4	P	P	TP
5	N	P	FP
6	P	P	TP
7	N	N	TN
8	N	N	TN
9	N	N	TN
10	P	N	FN

Confusion Matrix:

Prediction	Truth P	Truth N
P	4	2
N	1	3

accuracy: $\frac{4+3}{10} = 70\%$

specificity: $\frac{3}{5}$

Precision: $\frac{4}{4+2} = \frac{2}{3}$

recall: $\frac{4}{5} = 80\%$

F1 score: $2 \cdot \frac{\frac{4}{5} \cdot \frac{2}{3}}{\frac{4}{5}+\frac{2}{3}} = \frac{8}{11}$