1.

(c)

	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8	Cluster 9
Cluster Size	111	105	84	35	89	147	88	85	99	157
Most Common Label	[3]	[8]	[2]	[3]	[1]	[7]	[6]	[0]	[1]	[9]
Precent of Most Common Label	50.45%	40.00%	84.52%	65.71%	56.18%	44.22%	79.55%	94.12%	61.62%	40.13%
Error	0.50	0.60	0.15	0.34	0.44	0.56	0.20	0.06	0.38	0.60

The error on cluster is calculated by taking the number of wrong labels in the cluster and dividing it by the cluster size

By this calculation we can deduct that the algorithm classified correctly wrong about 419 samples, therefore the error is

$$\frac{419}{100} = 0.419 = 41.9\%$$

(d)

	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8	Cluster 9
Cluster Size	291	1	1	1	1	1	1	1	1	1
Most Common Label	[7]	[9]	[1]	[7]	[4]	[9]	[2]	[7]	[8]	[6]
Precent of Most Common Label	11.34%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Error	0.89	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

The same way we calculated the error before, now we get that the error is $\frac{258}{300} = 0.863.. \approx 86.3\%$

Clearly k-means worked better for this problem.

(e)

Kmeans:

	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Cluster Size	156	142	179	270	166	87
Most Common Label	[3]	[1]	[8]	[7]	[6]	[0]
Precent of Most Common Label	37.18%	69.72%	36.87%	34.44%	53.01%	82.76%
Error	0.63	0.30	0.63	0.66	0.47	0.17

Error: $\frac{524}{1000} = 0.524 = 52.4\%$

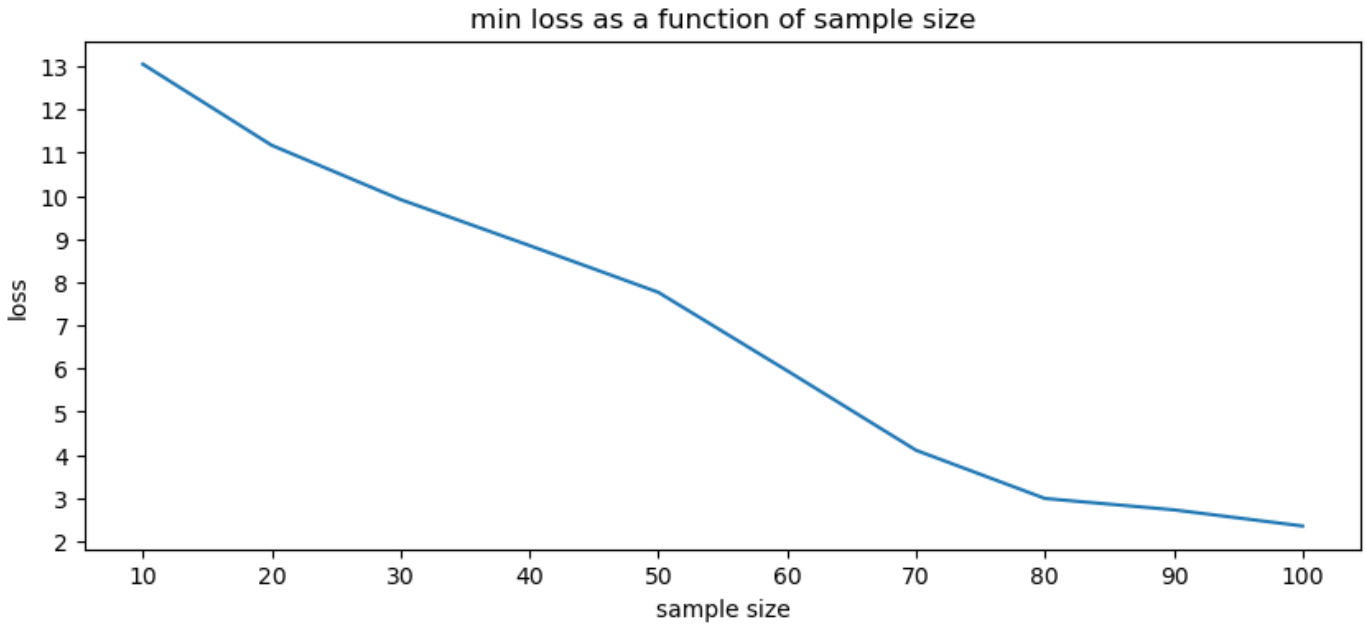
Single-linkage

	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Cluster Size	295	1	1	1	1	1
Most Common Label	[7]	[2]	[4]	[3]	[7]	[8]
Precent of Most Common Label	11.86%	100.00%	100.00%	100.00%	100.00%	100.00%
Error	0.88	0.00	0.00	0.00	0.00	0.00

Error: $\frac{260}{300} = 0.866.. \approx 86.6\%$

We can see that the k means algorithm got better results for k=10. It makes sense since we know the data is originally divided into 10 sets, each represent a digit.

2.(a)



(b)

As the training set grows, we expected the loss on the distribution to reduce. Based on what we've seen in class: $\ell(h,D) \leq \ell(h,S) + O\left(\frac{(B^2R^2)}{\sqrt{m}}\right)$. we used the λ that minimizes $\ell(h,S)$ and as m grows, $\left(\frac{(B^2R^2)}{\sqrt{m}}\right)$ gets smaller.

(c) yes, we can see it in the plot.

(d)

We saw in class that the bays-optimal regressor for **squared loss** is $h_{bayes}(x) = \mathbb{E}[Y|X=x]$.

$$\mathbb{E}[Y|X=x] = \mathbb{E}[\langle w,x\rangle + \eta|X] = \langle w,x\rangle + \mathbb{E}[\eta|X] \underset{\substack{\eta\sim N(0,\sigma)\\ \Rightarrow \mathbb{E}(\eta)=0\\ \Rightarrow \mathbb{E}[\eta|X]=0}}{=} \langle w,x\rangle \Rightarrow \underset{(squared\ loss)}{h_{bayes}} = \langle w,x\rangle$$

bays-optimal regressor for **absolute loss** is $MEDIAN_{(X,Y)\sim D}[Y|X=x]$

$$MEDIAN_{(X,Y)\sim D}[Y|X=x] = MEDIAN_{X\sim D}[\langle w,x\rangle + \eta|X=x]$$

And since $\langle w,x\rangle$ is a scalar, not random variable:

$$= \langle w,x\rangle + MEDIAN_{X\sim D}[\eta|X=x] = \langle w,x\rangle \Rightarrow \underset{(absolute\ loss)}{h_{bayes}} = \langle w,x\rangle$$

Where the last equality is based on the fact that $\eta\sim N(0,\sigma)$ and median of gaussian random variable is its mean.

3.(a)

According to GD algorithm:

$$w^{(t+1)} = w^{(t)} - \eta \nabla f(w^{(t)})$$

Let $g(w) := \lambda \|w\|$

$z_i(w) \coloneqq (\langle w, x_i \rangle - y_i)^2$

$f(w) \coloneqq g(w) + \Sigma_{i=1}^{m} z_i(w)$

we will find $\nabla f(w)$.

$\nabla g(w) = \lambda \cdot \frac{1}{2} \cdot \frac{2w}{\|w\|} = \frac{\lambda w}{\|w\|}$

$\nabla z_i(w) = 2(\langle w, x_i \rangle - y_i) * x_i$

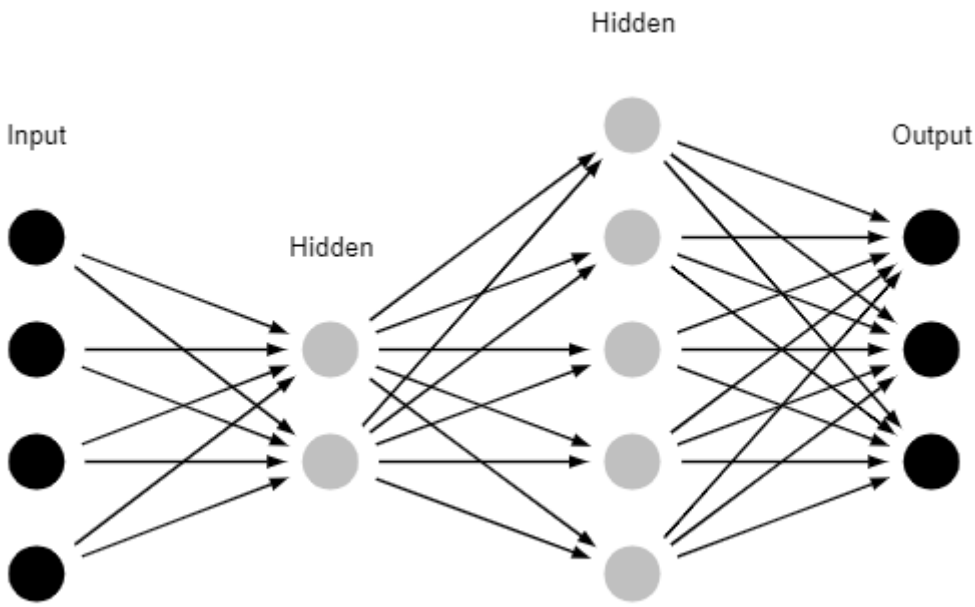
$\nabla f(w) = \nabla g(w) + \Sigma_{i=1}^{m} \nabla \mathrm{z_i}(w) = \dfrac{\lambda w}{\|w\|} + \Sigma_{i=1}^{m} 2(\langle w, x_i \rangle - y_i) * x_i$

Hence, $w^{(t+1)} = w^{(t)} - \eta \left(\frac{\lambda w}{\|w^{(t)}\|} + \Sigma_{i=1}^{m} 2\left(\langle w^{(t)}, x_i \rangle - y_i\right) * x_i \right)$

(b)

$$w^{(t+1)} = w^{(t)} - \eta \left(\frac{\lambda w}{\|w^{(t)}\|} + 2\left(\langle w^{(t)} | x_i \rangle - y_i\right) * x_i \right)$$

4.(a)



(b).

Since the input layer has 4 neurons, $X = \mathbb{R}^4$

(c)

Since $\psi(o_1, o_2, o_3) = argmax_{i\in[k]} o_i$, $Y = \{1,2,3\}$

(d)

$$|w| = 4 * 2 + 2 * 5 + 5 * 3 \ = 33 \Rightarrow w \in \mathbb{R}^{33}$$

$H = \{h_w(x) | w \in \mathbb{R}^{33}\}$

Then $h_w\big((x_1, x_2, x_3, x_4)\big) =$

$$\underset{12\leq i\leq 14}{\operatorname{argmax}}\Big(\sum_{j=7}^{11} w_{j,i} \cdot \sigma\Big(\sum_{k=5}^{6} w_{m,j} \cdot \sigma\Big(\sum_{l=1}^{4} w_{t,m} \cdot x_t\Big)\Big)\Big)$$

5.(a)

For each sample x there are d attributes, and we can test 3 options for θ, hence there are $3d + 2$ options for inner vertex (+2 for label). In a tree of depth n there are at most 2^{n+1} nodes, thus $|\overline{H}_n| \leq (3d + 2)^{2^{n+1}}$

(b)

Danny is wrong, since ID3 is heuristic algorithm and not ERM it does not guarantee minimal error. After performing pruning, we can't guarantee we will get minimal error. Thus PAC assumption on the sample size does not apply.

6.(a)

The assumption **does not hold**.

Naïve bayes assumption is $P[X = x|Y = y] = \Pi_{i=1}^{n} P[X_i = x_i|Y = y]$

But in the given distribution,

$P(x_1 = -1|Y = -1) = \frac{5}{60} + 0$ and $P(x_2 = +1|Y = -1) = \frac{4}{60} + 0$

$$\Rightarrow P(x_1 = -1|Y = -1) * P(x_2 = +1|Y = -1) > 0$$

And $P[X = (-1, +1) \mid Y = -1] = \frac{P\left[Y = -1 \middle| X = (-1, +1)\right] * P[X = (-1, +1)]}{P[Y = -1]} = \frac{\frac{0}{\frac{24}{60}+0} * P[X = (-1, +1)]}{\frac{5}{60}+0+\frac{11}{60}+\frac{4}{60}} = 0$

Which means $0 = [X = x|Y = y] \neq \Pi_{i=1}^{n} P[X_i = x_i|Y = y] > 0$ for some X, Y

(b)

$$P[Y = 1] = \frac{6}{60} + \frac{24}{60} + \frac{2}{60} + \frac{8}{60} = \frac{2}{3} \Rightarrow P[Y = -1] = \frac{1}{3}$$

From previous section, we know we can't use the naïve bayes assumption.

Hence $h_{bayes} = \underset{y \in Y}{argmax}\ (P[Y = y] * P[X = x|Y = y])$

$$= \underset{y \in Y}{argmax} \left(P[Y = y] * \frac{P[X = x \cap Y = y]}{P[Y = y]} \right) = \underset{y \in Y}{argmax}(P[X = x \cap Y = y])$$

For $x = (-1, -1)$ we get:

$P[X = x \cap Y = 1] = \frac{6}{60} > P[X = x \cap Y = -1] = \frac{5}{60} \Rightarrow$ for $x = (-1, -1)$ the predictor will return 1.

For $x = (-1, 1)$ we get:

$P[X = x \cap Y = 1] = \frac{24}{60} > P[X = x \cap Y = -1] = 0 \Rightarrow$ for $x = (-1, 1)$ the predictor will return 1.

For $x = (1, -1)$ we get:

$P[X = x \cap Y = 1] = \frac{2}{60} < P[X = x \cap Y = -1] = \frac{11}{60} \Rightarrow$ for $x = (1, -1)$ the predictor will return -1

For $x = (1, 1)$ we get:

$P[X = x \cap Y = 1] = \frac{8}{60} > P[X = x \cap Y = -1] = \frac{4}{60} \Rightarrow$ for $x = (1, 1)$ the predictor will return 1

Therefore ,

$$h_{bayes}(x) = \begin{cases} -1, x = (1, -1) \\ 1, \quad else \end{cases}$$

7.(a)

From the definition of x_t we can see that there's a linear dependence between the first and second coordinates to the third and fourth coordinates, hence the degree of X is 2 and therefore the degree of $X^T X$ is 2 as well. Since $X^T X \in M_{4\times 4}$ and the degree of $X^T X$ is 2, therefore it has 2 eigenvalues 0.

it has kernel of dimension 2, and from the dimension theorem ($suppose\ A \in M_{n\times n}\ we\ get\ \ n = \dim\big(image(A)\big) + \dim(ker(A))$

we know that the eigenvalues of $X^T X$ are non-negative. thus, the 2 smallest eigenvalues of $X^T X$ are both 0 and the distortion is $0 + 0 = 0$

(b)

Let $x_1 \coloneqq (1,0,1,0)$, $x_2 \coloneqq (0,1,1,1), x_3 \coloneqq (2,1,5,9)\ x_4 = (2,0,4,4)$

Note that x_1, x_2, x_3, x_4 all satisfy $x_t(3) = \big(x_t(1)\big)^2 + \big(x_t(2)\big)^3 \wedge\ \ x_t(4) = \big(x_t(3) - x_t(1)\big)^2$

For $m = 3, X = \begin{pmatrix} [x_1] \\ [x_2] \\ [x_3] \\ [x_4] \end{pmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 2 & 1 & 5 & 9 \\ 2 & 0 & 4 & 4 \end{bmatrix} \Rightarrow X^T X = \begin{bmatrix} 9 & 2 & 19 & 26 \\ 2 & 2 & 6 & 10 \\ 19 & 6 & 43 & 62 \\ 26 & 10 & 62 & 98 \end{bmatrix}$

After performing gaussian elimination on $X^T X$ we get $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

We got all the eigenvalues are 1, therefore the distortion is sum of positives and > 0.

8.(a)

Let $\chi = \{1,2,3\}$ (there's a mistake in the question for the definition of $\mathbb{P}_{X\sim D_\theta}$ so we defined χ as followed) and $\Theta \subseteq [0,1]^3$ $such\ that\ for\ \theta(1) + \theta(2) + \theta(3) = 1$

(a)

Let $\Theta' = \{\theta \in \Theta \mid \theta(1) = 3\theta(2)\}$

Thus we can represent $\theta \in \Theta'$ as:

$\theta = (3\theta_2, \theta_2, 1 - 4\theta_2)$

We define for $1 \leq i \leq 3$: $k_i \coloneqq \Sigma_{j=1}^{m} \mathbb{I}[x_j = i]$

Thus

$\mathbb{P}_{S'\sim D_\theta}[S' = S] = \Pi_{i=1}^{m} \mathbb{P}_{X\sim D_\theta}[X = x_i] = \Pi_{i=1}^{m}(3\theta_2 \mathbb{I}[x_i = 1] + \theta_2 \mathbb{I}[x_i = 2] + (1 - 4\theta_2)\mathbb{I}[x_i = 3])$

$= \Pi_{x_i=1} 3\theta_2 \Pi_{x_i=2} \theta_2 \Pi_{x_i=3}(1 - 4\theta_2) = (3\theta_2)^{k_1}(\theta_2)^{k_2}(1 - 4\theta_2)^{k_3}$

Thus,

$L(S, \theta) = \log\ \mathbb{P}_{S'\sim D_\theta^m}[S' = S] = \log\left((3\theta_2)^{k_1}(\theta_2)^{k_2}(1 - 4\theta_2)^{k_3}\right)$

$= \log\left((3\theta_2)^{k_1}\right) + log\ \left((\theta_2)^{k_2}\right) + \log\left((1 - 4\theta_2)^{k_3}\right) = \log\left(3^{k_1}\right) + \log\left(\theta_2^{k_1}\right) + \log\left(\theta_2^{k_2}\right) + \log\left((1 - 4\theta_2)^{k_3}\right)$

$= \log\left(3^{k_1}\right) + k_1 \log(\theta_2) + k_2 \log(\theta_2) + k_3 \log(1 - 4\theta_2)$

And

$$\frac{D\left(L(S, \theta)\right)}{D\theta_2} = \frac{k_1}{\theta_2} + \frac{k_2}{\theta_2} - \frac{4k_3}{1 - 4\theta_2}$$

Now, we'll demand the derivative to be 0 to find maximum value

$$\frac{D\left(L(S, \theta)\right)}{D\theta_2} = 0$$

$$\frac{k_1 + k_2}{\theta_2} - \frac{4k_3}{1 - 4\theta_2} = 0$$

$$\frac{(k_1 + k_2)(1 - 4\theta_2)}{\theta_2(1 - 4\theta_2)} - \frac{4k_3\theta_2}{\theta_2(1 - 4\theta_2)} = 0$$

$$k_1 - 4\theta_2 k_1 + k_2 - 4\theta_2 k_2 - 4k_3\theta_2 = 0$$

$$4\theta_2 k_1 + 4\theta_2 k_2 + 4\theta_2 k_3 = k_1 + k_2$$

$$\theta_2(4k_1 + 4k_2 + 4k_3) = k_1 + k_2$$

$$\theta_2 = \frac{k_1 + k_2}{(4k_1 + 4k_2 + 4k_3)} = \frac{k_1 + k_2}{(4k_1 + 4k_2 + k_3)} = \frac{k_1 + k_2}{4(k_1 + k_2 + k_3)} \underset{k_1+k_2+k_3=m}{=} \frac{k_1 + k_2}{4m}$$