Question 1

1. **FALSE**: By definition, for a given test example, Bayes' Classifier outputs the label that is most 'probable'. Most probable doesn't necessaliy mean the correct label.
2. **TRUE**: At a given x, we predict using the 'discrete' points in the box for that x. Since the picking of point is discretised, the resultant output is also expected to be 'discrete'.
3. **FALSE**: Naive bayes would achieve Bayes error rate, which we know from (1), is not zero.
4. **FALSE**: 'Decision Trees" would typically help in classification/decision problems. Logistic regression would give probability in the output as desired.
5. **FALSE**: $\phi(\alpha x) = \alpha^2\phi(x)$ which is clearly non-linear.
6. **FALSE**: What if the optimal regressor is required to be non-linear.
7. **TRUE**: In the optimal solution for unconstrained problem, the indicator would ensure that $f_i(x) \leq 0$.
8. **FALSE**: For $d = 1$, it is a convex set.

Question 2

(a) Each coin is a Bernoulli variable with probability of 1 as b; thus, the likelihood of the data is:

$$\mathrm{P}\left(x_1, \ldots, x_n \,|\, b\right) = \prod_{i=1}^{n} b^{x_i}(1-b)^{1-x_i} = b^{\sum_{i=1}^{n} x_i}(1-b)^{n-\sum_{i=1}^{n} x_i} = b^k(1-b)^{n-k},$$

where k is defined to be total number of 1's: $\sum_{i=1}^{n} x_k$. This implies that the log-likelihood of the data is:

$$\log\left(\mathrm{P}\left(x_1, \ldots, x_n \,|\, b\right)\right) = k\log(b) + (n-k)\log(1-b).$$

One can take the derivative of this equation with respect to b in order to maximize the log-likelihood:

$$\frac{\mathrm{d}}{\mathrm{d}b}\left(\log\left(\mathrm{P}\left(x_1, \ldots, x_n \,|\, b\right)\right)\right) = \frac{k}{b} - \frac{n-k}{1-b}.$$

The optimal estimator $\hat{b}$ is given when the derivative equals 0:

$$\frac{k}{\hat{b}} - \frac{n-k}{1-\hat{b}} = 0 \rightarrow \frac{k}{\hat{b}} = \frac{n-k}{1-\hat{b}} \rightarrow k - k\hat{b} = n\hat{b} - k\hat{b} \rightarrow k = n\hat{b} \rightarrow \hat{b} = \frac{k}{n} = \frac{\sum_{i=1}^{n} x_i}{n}.$$

(b) An estimator is defined to be unbiased if the expectation of the estimator is equal to the parameter itself.

$$\mathrm{E}\left[\hat{b}\right] = \mathrm{E}\left[\frac{\sum_{i=1}^{n} x_i}{n}\right] = \frac{\mathrm{E}\left[\sum_{i=1}^{n} x_i\right]}{n} = \frac{\sum_{i=1}^{n} \mathrm{E}\left[x_i\right]}{n} = \frac{\sum_{i=1}^{n} \left(b \cdot 1 + (1-b)\cdot 0\right)}{n} = \frac{\sum_{i=1}^{n} b}{n} = \frac{nb}{n} = b$$

Therefore, the estimator is unbiased.

(c) The variance of one flip of the coin is equal to:

$$\begin{aligned}\mathrm{V}\left[x\right] &= \mathrm{E}\left[\left(x - \mathrm{E}\left[x\right]\right)^2\right] = b\left(1-b\right)^2 + \left(1-b\right)\left(0-b\right)^2 = b\left(1-2b+b^2\right) + b^2\left(1-b\right) \\ &= b - 2b^2 + b^3 + b^2 - b^3 = b - b^2 = b\left(1-b\right).\end{aligned}$$

(d) From part (iv) from Question 1 on Homework 1, it is known that if $g : \mathbb{R}^d \to \mathbb{R}^k$ is a differentiable function and the maximum likelihood estimator for $\theta \in \mathbb{R}^d$ is θ_{ML}, then the maximum likelihood estimator of $g(\theta)$ is $g(\theta_{\text{ML}})$. In this case, take $d = k = 1$ and $g(b) = b(1-b)$. Therefore, the maximum likelihood estimator of the variance of one coin flip is: $\hat{b}\left(1-\hat{b}\right) = \frac{\sum_{i=1}^{n} x_i}{n}\left(1 - \frac{\sum_{i=1}^{n} x_i}{n}\right)$.

(e) To answer this question, Bayes' rule can be utilized; in particular,

$$\mathrm{P}(b \mid x_1, \ldots, x_n) = \frac{\mathrm{P}(x_1, \ldots, x_n \mid b)\,\mathrm{P}(b)}{\mathrm{P}(x_1, \ldots, x_n)}.$$

Because $\mathrm{P}(x_1, \ldots, x_n)$ is constant with respect to b, one can instead maximize: $\mathrm{P}(x_1, \ldots, x_n \mid b)\,\mathrm{P}(b)$. Stated formally, this means that: $\arg\max_b(\mathrm{P}(b \mid x_1, \ldots, x_n)) = \arg\max_b(\mathrm{P}(x_1, \ldots, x_n \mid b)\,\mathrm{P}(b))$. Hence, the goal is to maximize $\mathrm{P}(x_1, \ldots, x_n \mid b)\,\mathrm{P}(b)$ with respect to b; one can start by writing the quantity that needs to be maximized:

$$\mathrm{P}(x_1, \ldots, x_n \mid b)\,\mathrm{P}(b) = 2b \cdot b^k (1-b)^{n-k},$$

where k is defined exactly as before. Taking the logarithm of both sides, one obtains:

$$\log\left(\mathrm{P}(x_1, \ldots, x_n \mid b)\,\mathrm{P}(b)\right) = \log(2b) + k\log(b) + (n-k)\log(1-b),$$

and because $\log(\cdot)$ is a monotonically increasing function, maximizing the quantity on the left is the same as maximizing the original quantity. Proceeding as before, one can take the derivative of this equation with respect to b in order to maximize the term on the left:

$$\frac{\mathrm{d}}{\mathrm{d}b}\left(\log\left(\mathrm{P}(x_1, \ldots, x_n \mid b)\,\mathrm{P}(b)\right)\right) = \frac{1}{b} + \frac{k}{b} - \frac{n-k}{1-b} = \frac{k+1}{b} - \frac{n-k}{1-b}.$$

The optimal estimator $\hat{b}$ is given when the derivative equals 0:

$$\frac{k+1}{\hat{b}} - \frac{n-k}{1-\hat{b}} = 0 \to \frac{k+1}{\hat{b}} = \frac{n-k}{1-\hat{b}} \to k+1-k\hat{b}-\hat{b} = n\hat{b}-k\hat{b} \to k+1 = (n+1)\hat{b} \to \hat{b} = \frac{k+1}{n+1} = \frac{1+\sum_{i=1}^{n} x_i}{1+n}.$$

Question 3

1. A is the training curve while B is the testing curve because the accuracy of B drops at the middle.
2. For trees of same sizes, new accuracies will drop. However, the new training curve will evenatually goes to 100%.
3. The new test curve will resemble the new training curve, and will not drop in the middle. With the new training data, it will not be biased toward a certain portion of the whole underdlying distribution.

Question 4

Part 1

i. C, error exists.

ii. A, passes through origin.

iii. B, max margins.

Part 2

i. B, same boundary.

ii. D, effects of translating space not known.

iii. A, no constraint on passing through origin.

iv. D, just translating (ii) and scaling (i).

v. D, effects of adding nonlinearity not known.

vi. C, less capacity.

Question 5

Part 1

Alice uses L2 norm. Now let us consider Bob. Bob uses $2e^{||x-x'||+1}$. Let us define distance between i and j in Alice's frame is $d_{alice,i,j}$.

Observe that if $d_{alice,i,j} < d_{alice,i,k} \implies e^{d_{alice,i,j}} < e^{d_{alice,i,k}} \implies e^{d_{alice,i,j}+1} < e^{d_{alice,i,k}+1} \implies 2e^{d_{alice,i,j}+1} < 2e^{d_{alice,i,k}+1} \implies d_{bob,i,j} < d_{bob,i,k}$ So j remains nearest neighbor of i even for Bob.

Another way of thinking the same problem is that Alice and Bob use same metric but different units. So even though units are different, being near is an invariant property which does not change.

Since Carol uses L1 norm which is a different metric, Carol doesn't have same distance

Part 2

i

It is given that nearest neighbor of x is S1 has a label 1 and nearest neighbor of x is S2 has a label 1. Now in $S1 \cup S2$, nearest neighbor is either the one is S1 or S2 both of which have a label of 1. Hence we this statement is true.

ii

It is given that nearest neighbor of x is S1 has a label 1 and nearest neighbor of x is S2 has a label 1. It is given that nearest neighbor of x is S1 has a label 1 and nearest neighbor of x is S2 has a label 1. Now in $S1 \cup S2$, we know closest point will have label 1 from above. But it could happen that the second and third closest points need not be from one of the points and could both have labels 0. In this scenario, the 3 NN algorithm returns a 0 but not a 1. So this statement is false.

iii

Consider 3 nearest neightbors in S1 to have labels 0,1,1 and 3 nearest neighbors of S2 to have labels 0,1,1. We know in this case 3NN returns 1 in both S1 and S2. But now in $S1 \cup S2$, it could so happen that 0,0,1 are 3 nearest neighbors which makes 0 as the label and not 1

Part 3 and 4

A simple expansion of Euclidean distance gives $\phi(x).\phi(x) + \phi(x').\phi(x') - 2 * \phi(x).\phi(x')$
From Kernel's definition, we can replace $\phi(x).\phi(x')$ wiht K(x,x')
So for doing a KNN we now consider distance to be
$K(x,x) + K(x,x') - 2K(x,x')$

So for doing a KNN we now consider distance to be
$K(x,x) + K(x,x') - 2K(x,x')$

Question 6

Consider augmenting the dataset with d additional examples; for the i-th example, let the i-th input coordinate be c with all other coordinates equal to 0 and the output be 0. The error term is now:

$$\begin{aligned}
\left\|w\tilde{X} - \tilde{y}\right\|^2 &= \left\| \begin{bmatrix} w_1 & w_2 & \cdots & w_d \end{bmatrix} \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1n} & c & 0 & \cdots & 0 \\ X_{21} & X_{22} & \cdots & X_{2n} & 0 & c & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ X_{d1} & X_{d2} & \cdots & X_{dd} & 0 & 0 & \cdots & c \end{bmatrix} - \begin{bmatrix} y_1 & y_2 & \ldots & y_d & 0 & 0 & \ldots & 0 \end{bmatrix} \right\|^2 \\
&= \left\| \begin{bmatrix} \hat{y}_1 & \hat{y}_2 & \ldots & \hat{y}_d & cw_1 & cw_2 & \ldots & cw_d \end{bmatrix} - \begin{bmatrix} y_1 & y_2 & \ldots & y_d & 0 & 0 & \ldots & 0 \end{bmatrix} \right\|^2 \\
&= \left\| \begin{bmatrix} \hat{y}_1 - y_1 & \hat{y}_2 - y_2 & \ldots & \hat{y}_d - y_d & cw_1 & cw_2 & \ldots & cw_d \end{bmatrix} \right\|^2 \\
&= \|wX - y\|^2 + \sum_{i=1}^{d} c^2 w_i^2 = \|wX - y\|^2 + c^2 \sum_{i=1}^{d} w_i^2.
\end{aligned}$$

This is the correct general principle, but to exactly recreate the loss, c must be equal to $\sqrt{\lambda\alpha}$. Then, the new loss to be minimized $\left(\left\|w\tilde{X} - \tilde{y}\right\|^2 + \lambda(1-\alpha)\|w\|_1\right)$ is exactly equal to the old loss to be minimized; therefore, the optimization problem has been recast as a lasso regression problem.