Example exam in Statistical Machine Learning: suggested solutions

1. i) False. Since k-NN with \( k = 1 \) always has zero training error
   ii) False. Parametric models don’t have to be linear in the parameters, e.g. a neural network is a parametric nonlinear model.
   iii) False. Both the training error and the test error can be seen as random variables (since they depend on the random training data) and thus this is not always the case, even though it is commonly so.
   iv) True. Ridge regression adds a penalty term to the cost function which penalizes large parameter values, and this will shrink the estimated parameters towards zero.
   v) False. Conditionally on the training data \( \mathcal{T} \), the ensemble members are (for a fixed input \( x \)) independent random variables (the randomness coming from the bootstrapping). Since the bagged model is computed as the average of these ensemble members, the law of large numbers ensures that the estimated model converges to the “mean model” as \( B \to \infty \),

   \[
   \frac{1}{B} \sum_{b=1}^{B} \hat{f}^*_{\mathcal{B}}(x) \to \mathbb{E}[\hat{f}^*_{\mathcal{B}}(x) \mid \mathcal{T}].
   \]

   Since this “mean model” does not depend on \( B \), we conclude that the flexibility of the bagged model does not increase with \( B \) and it will thus not overfit as \( B \) becomes large.
   vi) False. A classification problem is not limited to 2 classes.
   vii) False. The number of cars sold have an ordinal aspect and this is thus best viewed as a regression problem.
   viii) False. Cross-validation is based on an approximation of the test error and uses (part of) the training data for estimating this. Cross-validation is extensively used in “real world” settings!
   ix) True. To see why this is the case, consider first a tree with a single split which has 1 internal node (the root) and 2 leaf nodes, so the claim is true. Next, for each consecutive split we will create one additional internal node and one additional leaf node (or, more precisely, we create two new leaf nodes and transform a previous leaf node into an internal node). Thus, the claim remains true for any tree.
x) True. The model is linear in the parameters.

2. (a) A linear classifier can have a nonlinear decision boundary in the original input space if the inputs are transformed using a nonlinear transformation. In this case a decision boundary corresponding to a circle centered in the origin would completely separate the two classes in Figure 1 and hence achieve zero misclassification error. This decision boundary can be achieved by using \( X_1^2 + X_2^2 \) as input instead of \( X_1 \) and \( X_2 \). Note that the decision boundary for a linear classifier is always linear in the input space used.

(b) The misclassification loss is given by \( I(y \neq \hat{G}(x)) \) and the exponential loss is given by \( \exp(-yC(x)) \) where \( C(x) = x - 1.625 \). Hence we have:

<table>
<thead>
<tr>
<th>( x_i )</th>
<th>0.75</th>
<th>1</th>
<th>1.5</th>
<th>1.75</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y_i )</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

| \( C(x_i) \) | -0.875 | -0.625 | -0.125 | 0.125 | 0.375 |
| \( \hat{G}(x_i) \) | -1 | -1 | -1 | 1 | 1 |
| \( y_iC(x_i) \) | -0.875 | 0.625 | 0.125 | 0.125 | 0.375 |
| \( I(y \neq \hat{G}(x)) \) | 1 | 0 | 0 | 0 | 0 |
| \( \exp(-yC(x)) \) | 2.40 | 0.54 | 0.88 | 0.88 | 0.69 |

The total misclassification loss is 1 and the total exponential loss is 2.40 + 0.54 + 0.88 + 0.88 + 0.69 = 5.39

(c) We have the following parameters: \( \hat{\mu}_F = 25, \hat{\mu}_P = 40, \hat{\Sigma}_F = \hat{\sigma}_F^2 = 7^2, \hat{\Sigma}_P = \hat{\sigma}_P^2 = 10^2 \). Since 60% passed the exam \( \hat{\pi}_F = 0.4 \) and \( \hat{\pi}_P = 0.6 \). For LDA, the average within-class variance \( \hat{\Sigma} \) is required. Since \( N \gg 1 \):

\[
\hat{\Sigma} \approx \frac{1}{N} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T = 0.4\hat{\sigma}_F^2 + 0.6\hat{\sigma}_P^2
\]

The discriminant function for LDA is:

\[
\delta_k(x) = x^T \hat{\Sigma}^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^T \hat{\Sigma}^{-1} \hat{\mu}_k + \log \hat{\pi}_k
\]

Hence, after inserting the estimated parameter values:

\[
\delta_F(x) = 0.314x - 4.842
\]
\[
\delta_P(x) = 0.503x - 10.561
\]

The decision boundary is given by \( \hat{\delta}_F = \hat{\delta}_P \) which gives \( 0.314x - 4.842 = 0.503x - 10.561 \iff x \approx 30 \).
(d) Same parameters as in c). The discriminant function for QDA is:

\[ \hat{\delta}_k(x) = -\frac{1}{2} \log |\hat{\Sigma}_k| - \frac{1}{2} (x - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1} (x - \hat{\mu}_k) + \log \hat{\pi}_k \]

Hence, after inserting parameter values we get:

\[ \hat{\delta}_F(x) = -\frac{1}{98} x^2 + \frac{25}{49} x - 9.240 \]
\[ \hat{\delta}_P(x) = -\frac{1}{200} x^2 + \frac{40}{100} x - 10.814 \]

The decision boundary is given by \( \hat{\delta}_F = \hat{\delta}_P \) which gives

\[ -\frac{1}{98} x^2 + \frac{50}{98} x - 9.240 = -\frac{1}{200} x^2 + \frac{2}{5} x - 10.814 \iff x \approx 31 \]

Note that there are two solutions but one is negative and thus disregarded (you can’t study for a negative number of hours).

3. (a) We want to fit the function \( \beta_0 + \beta_1 X \), i.e. a straight line, to two training data points \((x_1 = 0, y_1)\) and \((x_2 = 1, y_2)\). However, since we have only two data points we can fit a line exactly through these two points, and this is naturally the least-squares estimate (since it attains zero training error). This results in

\[ \hat{\beta}_0 = y_1, \]
\[ \hat{\beta}_1 = y_2 - y_1. \]

(The same solution is of course obtained if we instead solve the normal equations.)

(b) First we note that the two training outputs can be written as

\[ y_1 = \beta_0 + \beta_1 x_1 + \varepsilon_1 = \varepsilon_1, \]
\[ y_2 = \beta_0 + \beta_1 x_1 + \varepsilon_2 = 1 + \varepsilon_2, \]

where we have used the fact that \( x_1 = 0, x_1 = 1, \beta_0 = 0 \) and \( \beta_1 = 1 \). Here, \( \varepsilon_1 \) and \( \varepsilon_2 \) are independent Gaussian random variables with zero mean and variance \( \sigma^2 \). Thus, it follows that \( y_1 \) and \( y_2 \) are also independent random variables with \( y_1 \sim \mathcal{N}(0, \sigma^2) \) and \( y_2 \sim \mathcal{N}(1, \sigma^2) \). Inserting this into the expression for the bias
we get
\[
E \left[ \hat{f}(x) - f(x) \right] = E \left[ \hat{\beta}_0 + \hat{\beta}_1 x - (\beta_0 + \beta_1 x) \right] \\
= E [y_1 + (y_2 - y_1)x - x] \\
= E[y_1] + (E[y_2] - E[y_1])x - x \\
= 0 + (1 - 0)x - x \\
= 0.
\]

That is, the model has zero bias, which is to be expected for a linear model which is assumed to be of the correct functional form.

Using the same expressions we can calculate the variance
\[
\text{Var}(\hat{f}(x)) = E \left[ (\hat{f}(x) - E[\hat{f}(x)])^2 \right] \\
= E \left[ (y_1 + (y_2 - y_1)x - x)^2 \right] \\
= E \left[ ((1 - x_1)\varepsilon_1 + x_2\varepsilon_2)^2 \right] \\
= E \left[ (1 - x_1)^2\varepsilon_1^2 + x_2^2\varepsilon_2^2 + 2(1 - x_1)x_2\varepsilon_1\varepsilon_2 \right] \\
= \{ (1 - x_1)^2 + x_2^2 \} \sigma^2 = 0.82\sigma^2,
\]
where we have used the fact that $E[\varepsilon_1^2] = E[\varepsilon_2^2] = \sigma^2$ and $E[\varepsilon_1\varepsilon_2] = E[\varepsilon_1]E[\varepsilon_2] = 0$ for the penultimate equality.

(c) We want to find the condition of $\text{MSE}^{\text{alt}}$ being smaller than $\text{MSE}^{\text{orig}}$. Recall that $\text{MSE} = \text{Bias}^2 + \text{Variance} + \text{Irreducible error}$, and insert that into the expression, i.e.
\[
\text{MSE}^{\text{alt}} < \text{MSE}^{\text{orig}} \iff \frac{0.4^2 + 0.5\sigma^2 + \sigma^2}{\sigma^2} < \frac{0.82\sigma^2 + \sigma^2}{\sigma^2} \iff \frac{0.4^2}{0.32\sigma^2} < \frac{0.16}{\sigma^2} \iff \frac{0.32}{\sigma^2} < \sigma^2 \iff 0.5 < \sigma^2.
\]

4. (a) The first split, i.e.
\[\text{at the root node, will divide the whole input space into two half-spaces. This must therefore correspond to the split point } b. \] 
\[\text{The two child nodes of the root will similarly correspond to the splits of these two half-spaces (if they are indeed split), which in our case must correspond to split points } a \text{ and } d, \] 
\[\text{respectively. Finally, we have split point } c \text{ which further divides} \]
one of regions. The resulting tree is thus as follows, with splitting criteria and final regions specified at the internal and leaf nodes, respectively.

\[ \begin{align*} &X_2 \leq b \\ &X_1 \leq d & X_1 \leq a \\ &X_2 \leq c & \quad \quad R_2 & R_3 \\ & & R_5 & R_4 \end{align*} \]

(b) This problem can be solved in two ways, either by using the formula for probabilistic linear regression (page 2, under "Linear regression and regularization") or by expressing the posterior \( p(\beta_0|y) \) in terms of the data and prior distributions utilizing Bayes’ theorem.

**Formula for probabilistic linear regression:**

For a probabilistic linear regression model with \( \varepsilon \sim \mathcal{N}(0, \sigma^2) \) and prior distribution \( p(\beta) = \mathcal{N}(\beta|\mu_0, \Sigma_0) \) the posterior distribution is

\[ p(\beta|y) = \mathcal{N}(\beta|\mu_N, \Sigma_N) \]

with

\[
\begin{align*}
\mu_N &= \Sigma_N (\Sigma_0^{-1} \mu_0 + \sigma^{-2} X^T y) \\
\Sigma_N &= (\Sigma_0 + \sigma^{-2} X^T X)^{-1}
\end{align*}
\]

In the given model of the system the noise has mean value not equal to zero, so to use the above formula the model \( Y = \beta_0 + \varepsilon \) must be rewritten according to

\[ Y = \beta_0 + \mu + \tilde{\varepsilon}, \quad \tilde{\varepsilon} \sim \mathcal{N}(0, \sigma^2) \]
which can in turn be rewritten as

\[ \tilde{Y} = Y - \mu = \beta_0 + \tilde{\epsilon}, \quad \tilde{\epsilon} \sim \mathcal{N}(0, \sigma^2) \]

which is now on the appropriate form to apply the formula for the posterior. With parameters \( \mu_0 = 0, \Sigma_0 = \sigma_0^2, X = 1, y = \tilde{y} = y - \mu \), the posterior mean and variance is given by

\[ \Sigma_N = \frac{\sigma_0^2 \sigma^2}{\sigma_0^2 + \sigma^2}, \quad \mu_N = \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2} (y - \mu) \]

The posterior pdf is given by

\[ p(\beta_0|y) = \mathcal{N}(\beta_0|\mu_N, \Sigma_N) = \frac{1}{(2\pi)^{p/2} |\Sigma_N|^{1/2}} \exp \left( -\frac{1}{2} (\beta_0 - \mu_N)^T \Sigma_N^{-1} (\beta_0 - \mu_N) \right) \]

and the log-posterior is hence given by (ignoring terms not depending on \( \beta_0 \))

\[ \log p(\beta_0|y) = -\frac{1}{2} (\beta_0 - \mu_N)^T \Sigma_N^{-1} (\beta_0 - \mu_N) \]

Inserting expressions for \( \Sigma_N \) and \( \mu_N \) above gives

\[ \log p(\beta_0|y) = -\frac{\sigma_0^2 + \sigma^2}{2\sigma_0^2 \sigma^2} \left( \beta_0 - \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2} (y - \mu) \right)^2 \]

Hence \( a = \frac{\sigma_0^2 + \sigma^2}{2\sigma_0^2 \sigma^2} \) and \( b = \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2} (y - \mu) \) in the given expression for the log-posterior.

**Using Bayes’ theorem:**

The pdf for the data distribution and the prior respectively are given by

\[ p(y|\beta_0) = \mathcal{N}(\beta_0 + \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{(y - (\beta_0 + \mu))^2}{2\sigma^2} \right) \]

\[ p(\beta_0) = \mathcal{N}(0, \sigma_0^2) = \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp \left( -\frac{\beta_0^2}{2\sigma_0^2} \right) \]

Bayes’ theorem gives

\[ p(\beta_0|y) = \frac{p(y|\beta_0)p(\beta_0)}{p(y)} \propto p(y|\beta_0)p(\beta_0) \]
On log-form this becomes
\[
\log p(\beta_0|y) = \log(p(y|\beta_0)p(\beta_0)) = \log p(y|\beta_0) + \log p(\beta_0)
\]
Inserting the expressions for \(p(y|\beta_0)\) and \(p(\beta_0)\) and removing all terms that do not depend on \(\beta_0\) gives
\[
\log p(\beta_0|y) = -\frac{(y - (\beta_0 + \mu))^2}{2\sigma^2} - \frac{\beta_0^2}{2\sigma_0^2}
\]
This expression can be rewritten as (“completing the square”)
\[
\log p(\beta_0|y) = -\frac{\sigma_0^2(y - (\beta_0 + \mu))^2 + \sigma^2\beta_0^2}{2\sigma_0^2\sigma^2} = \text{[ignore non-}\beta_0\text{ terms]}
\]
\[
= -\frac{-2\sigma_0^2y\beta_0 + \sigma_0^2\beta_0^2 + 2\sigma_0^2\mu\beta_0 + \sigma^2\beta_0^2}{2\sigma_0^2\sigma^2}
\]
\[
= -\frac{(\sigma_0^2 + \sigma^2)(\beta_0^2 - 2\frac{\sigma_0^2(y-\mu)}{\sigma_0^2+\sigma^2}\beta_0)}{2\sigma_0^2\sigma^2}
\]
\[
= \text{[ignore non-}\beta_0\text{ terms]}
\]
\[
= -\frac{\sigma_0^2 + \sigma^2}{2\sigma_0^2\sigma^2} \left(\beta_0 - \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2}(y - \mu)\right)^2
\]
Hence \(a = \frac{\sigma_0^2 + \sigma^2}{2\sigma_0^2\sigma^2}\) and \(b = \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2}(y - \mu)\) in the given expression for the log-posterior.

**Interpretation:**
It is clear that the log-posterior indeed can be written on the form \(p(\beta_0|y) = a(\beta_0 - b)^2 + \text{const. with } a \text{ and } b \text{ as stated above.}\) This expression implies that the posterior is also a Gaussian since the log-posterior is a quadratic form (i.e. its form corresponds to \(\frac{-1}{2}(x - \mu)^T\Sigma^{-1}(x - \mu)\)), where we can identify the mean as being equal to \(b\) and the variance as being equal to \(-1/(2a)\).

(c) For the Gaussian process \(f \sim \mathcal{GP}(m, k)\) two inputs \(x, x'\) will give the output
\[
\left(\begin{array}{c} f(x) \\ f(x') \end{array}\right) \sim \mathcal{N}\left(\left(\begin{array}{c} m(x) \\ m(x') \end{array}\right), \left(\begin{array}{cc} k(x, x) & k(x, x') \\ k(x', x) & k(x', x') \end{array}\right)\right)
\]
Here \(x = -2, x' = -1, m(x) = 0\) and \(k(x, x') = \exp\left(-\frac{1}{2}(x - x')^2\right)\) which gives the output
\[
\left(\begin{array}{c} f(-2) \\ f(-1) \end{array}\right) \sim \mathcal{N}\left(\left(\begin{array}{c} m(-2) \\ m(-1) \end{array}\right), \left(\begin{array}{cc} k(-2, -2) & k(-2, -1) \\ k(-1, -2) & k(-1, -1) \end{array}\right)\right)
\]
\[
\sim \mathcal{N}\left(\left(\begin{array}{c} 0 \\ 0 \end{array}\right), \left(\begin{array}{cc} 1 & 0.779 \\ 0.779 & 1 \end{array}\right)\right)
\]
Hence from the mean of the process it is clear that the pdf of the process should be centered in $(0, 0)$ which is fulfilled by figure i, iv and v. From the covariance matrix it is clear that there is a positive correlation between the two outputs (off-diagonal elements are positive) and this is only fulfilled in figure i. So figure i shows the level curves of the vector.

5. (a) 

**Similarities:**
- Both methods are ensemble methods and trains many models that make a collective decision
- Both models reduce the variance compare to that of a single base model

**Differences:**
- Boosting can overfit to the training data while bagging can’t
- Boosting can reduce bias as well as then variance
- Bagging can be trained in parallel while boosting is sequential

(b) We want to minimize the expression

$$E(\alpha) := \sum_{i=1}^{N} w_i^b \exp(-\alpha y_i \hat{G}^b(x_i))$$

with respect to $\alpha$. Using the second hint of the problem formulation, we can do this by differentiating the expression, setting the derivative to zero, and solving for $\alpha$.

Before we do this, however, we make use of the first hint of the problem formulation to simplify the expression under study. Note first that using the class labels $-1$ and $1$ implies that

$$y_i \hat{G}^b(x_i) = \begin{cases} 
1 & \text{if } G^b(x_i) = y_i, \\
-1 & \text{if } G^b(x_i) \neq y_i.
\end{cases}$$

Thus, the expression (1) can be written as

$$E(\alpha) = e^{-\alpha} \sum_{i=1}^{N} w_i^b I(G^b(x_i) = y_i) + e^{\alpha} \sum_{i=1}^{N} w_i^b I(G^b(x_i) \neq y_i),$$

where we have used the indicator function to split the sum into two sums: the first ranging over all correctly classified data points,
and the second ranging over all erroneously classified data points. Furthermore, for notational simplicity we define $W_c$ and $W_e$ for the sum of weights of correctly classified data points and erroneously classified data points, respectively.

Now, we compute the derivative and set to zero:

$$\frac{dE}{d\alpha} = -e^{-\alpha}W_c + e^\alpha W_e = 0 \iff e^\alpha W_e = e^{-\alpha}W_c$$

$$\iff e^{2\alpha} = \frac{W_c}{W_e} \iff \alpha = \frac{1}{2} \log \left( \frac{W_c}{W_e} \right).$$

It remains to write the solution on the format asked for in the problem formulation. With $W := \sum_{i=1}^{N} w^b_i$ denoting the sum of all weights, we note first that $W_c = W - W_e$ and thus,

$$\alpha = \frac{1}{2} \log \left( \frac{W - W_c}{W_e} \right) = \frac{1}{2} \log \left( \frac{1 - W_e/W}{W_e/W} \right).$$

Finally, noting that $\text{err}_b = W_e/W$ completes the proof.