This exam consists of 12 parts. I'll score it at 10 points per problem/part and add your best 8 scores to get an exam score (out of 80 points possible). Some parts will go faster than others, and you'll be wise to do them first.
1. Below are 3 class-conditional distributions for a predictor \( x \) in a \( K = 3 \) class 0-1 loss classification problem. Suppose that probabilities of \( y = k \) for \( k = 1, 2, 3 \) are \( \pi_1 = .4, \pi_2 = .3, \) and \( \pi_3 = .3 \). For each value of \( x \) give the corresponding value of the optimal (Bayes) classifier \( f^{opt} \) (fill in the last row of the table).

\[
\begin{array}{ccccccc}
\text{x} & 1 & 2 & 3 & 4 & 5 & 6 \\
\text{y = 1} & .08 & .04 & .08 & .04 & .04 & .12 \\
\pi_1 & .2 & .1 & .2 & .1 & .3 \\
\text{y = 2} & .03 & .03 & .09 & .09 & .03 & .03 \\
\pi_2 & .1 & .3 & .3 & .1 & .1 \\
\text{y = 3} & .06 & .03 & .06 & .06 & .06 & .03 \\
\pi_3 & .2 & .1 & .2 & .2 & .1 \\
\end{array}
\]

\[
\arg\max_y \pi_y f_y(x)
\]

\[
\begin{array}{ccccccc}
\text{f^{opt}(x)} & 1 & 1 & 2 & 2 & 3 & 1 \\
\end{array}
\]
2. A training set of size \( N = 3000 \) produces counts of \((x, y)\) pairs as in the table below. (Assume these represent a random sample of all cases.) For each value of \( x \) give the corresponding value of an approximately optimal 0-1 loss (Bayes) classifier \( \hat{f}^{\text{opt}} \) (fill in the last row of the table).

<table>
<thead>
<tr>
<th>( x )</th>
<th>( y = 1 )</th>
<th>( y = 2 )</th>
<th>( y = 3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>95</td>
<td>305</td>
<td>150</td>
</tr>
<tr>
<td>2</td>
<td>155</td>
<td>105</td>
<td>190</td>
</tr>
<tr>
<td>3</td>
<td>145</td>
<td>195</td>
<td>160</td>
</tr>
<tr>
<td>4</td>
<td>205</td>
<td>140</td>
<td>155</td>
</tr>
<tr>
<td>5</td>
<td>105</td>
<td>195</td>
<td>150</td>
</tr>
<tr>
<td>6</td>
<td>150</td>
<td>155</td>
<td>245</td>
</tr>
</tbody>
</table>

\[
\hat{f}^{\text{opt}}(x) = \text{argmax}_y \left( \frac{\hat{f}_y(x)}{\frac{1}{N}} \right) = \text{argmax}_y \left( \frac{\text{count of observations with } y}{N} \right) \left( \frac{\text{count of observations with } (x, y)}{\text{count of observations with } y} \right) = \text{argmax}_y \left( \text{count of observations with } (x, y) \right)
\]
Below is a cartoon representing the results of 3 different runs of support vector classification software on a set of training data representing $K = 3$ different classes in a problem with input space $\mathbb{R}^2$. Each pair of classes was used to produce a linear classification boundary for classification between those two. (Labeled arrows tell which sides of the lines correspond to classification to which classes.) 7 different regions are identified by Roman numerals on the cartoon. Indicate (fill in the table at the bottom of the page) values of an OVO (one-versus-one) classifier $\hat{f}^{\text{ovo}}$ for this situation. (In each cell, write exactly one of the numbers 1, 2, or 3, or "?" if there is no clear choice for that region.)

<table>
<thead>
<tr>
<th>Region</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
<th>V</th>
<th>VI</th>
<th>VII</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{f}(x)$</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td>3</td>
<td>3</td>
<td>?</td>
<td>2</td>
</tr>
</tbody>
</table>
4. At a particular input vector of interest in a SEL prediction problem, say \( x \), the conditional mean of \( y \mid x \) is 3. Two different predictors, \( \hat{f}_1(x) \) and \( \hat{f}_2(x) \) have biases (across random selection of training sets of fixed size \( N \)) at this value of \( x \) that are respectively .1 and −.5. The random vector of predictors at \( x \) (randomness coming from training set selection) has covariance matrix

\[
\text{Var}\left(\begin{pmatrix} \hat{f}_1(x) \\ \hat{f}_2(x) \end{pmatrix}\right) = \begin{pmatrix} 1 & .25 \\ .25 & 1 \end{pmatrix}
\]

If one uses a linear combination of the two predictors

\[
\hat{f}_{\text{ensemble}}(x) = a\hat{f}_1(x) + b\hat{f}_2(x)
\]

there are optimal values of the constants \( a \) and \( b \) in terms of minimizing the expected (across random selection of training sets) squared difference between \( \hat{f}_{\text{ensemble}}(x) \) and 3 (the conditional mean of \( y \mid x \)). Write out (but do not try to optimize) an explicit function of \( a \) and \( b \) that (in theory) could be minimized in order to find these optimal constants.

\[
E \left( 3 - (a\hat{f}_1 + b\hat{f}_2)^2 \right) = (3 - E(a\hat{f}_1 + b\hat{f}_2))^2 + \text{Var}(a\hat{f}_1 + b\hat{f}_2)
\]

\[
= (3 - a(3.1) - b(2.5))^2 + (a, b) \begin{pmatrix} 1 & .25 \\ .25 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}
\]

This is a quadratic in \( a, b \).
5. Below is a plot of a simple \((x, y)\) training data set. Plotted also are the (smooth) conditional mean function and two sets of predictions made from the data. One set is from a single tree predictor and the other set is from a random forest predictor based on a fairly large number of trees. (Since the input is only one-dimensional there can be no random selection of features for each tree in the forest, and so the latter is really simply a bagged tree.)

In this picture, for \(x\) near 1.0, both sets of predictions are substantially below the conditional mean. Do you think this is purely a random phenomenon that would be corrected "on average" across many different training sets? If so, why? If not, why not?

Is this likely just "random/the luck of the draw" for this training set?  YES  or  NO  (circle one)  

WHY?

As one splits on \(x\) down to some number of points in a final leaf on the tree, \(x = 1.0\) will always be represented by a number of \((x, y)\) pairs with \(x < 1\) and Thus \(E[y|x] < E[y|x = 1]\). Since \(\hat{f}(x)\) will be an arithmetic average of the corresponding \(y's\) one will have \(\hat{f}(i) < E[y|x = 1]\) and \(\hat{f}(1)\) (and thus \(\hat{f}(x)\) for \(x\) close to 1) will have negative bias.
6. Consider a 2-class 0-1 loss classification problem with \{-1,1\} coding of \( y \). For input \( x \in \mathbb{R}^2 \) and a parameter \( \gamma > 0 \), based on a training set of size \( N \) consider the classifier

\[
\hat{f}(x) = \begin{cases} 
1 & \text{if } \sum_{i \text{ with } y_i = 1} \exp(-\gamma \|x - x_i\|^2) > \sum_{i \text{ with } y_i = -1} \exp(-\gamma \|x - x_i\|^2) \\
-1 & \text{if } \sum_{i \text{ with } y_i = 1} \exp(-\gamma \|x - x_i\|^2) < \sum_{i \text{ with } y_i = -1} \exp(-\gamma \|x - x_i\|^2)
\end{cases}
\]

On what basis might one expect that for large \( N \) this classifier is approximately optimal?

For what "voting function" \( g(x) \) is \( \hat{f}(x) = \text{sign}(g(x)) \)? Is this a linear combination of radial basis functions?

Clearly,

\[
\sum_{i \text{ with } y_i = 1} \exp(-\gamma \|x - x_i\|^2) - \sum_{i \text{ with } y_i = -1} \exp(-\gamma \|x - x_i\|^2)
\]

is a l.c. of radial basis functions (with coefficients all \( \pm 1 \)).

Why will \( \hat{f}(x) \) typically not be of the form of a support vector machine based on a Gaussian kernel?

Support vector machines based on the Gaussian kernel

\[
K(\bar{x}, \bar{x}) = \exp(-\gamma \|\bar{x} - \bar{x}\|^2)
\]

do have voting functions that are l.c.'s of radial basis functions. But much of the whole "idea" of support vector machines is that typically only a few of the coefficients are non-zero. Above, \( \gamma \) are non-zero.
7. Overall, only a very small fraction of people presented with a certain merchandizing offer will respond to it. A set of 5 qualitative predictors (potential personal traits) is thought to be related to response. Values for these 5 predictors are obtained from a group of 96 people who responded to the offer and from a group of 604 who did not. Treating the input \( x_j \) as taking the value 1 if a subject has trait \( j \) and 0 otherwise, a model for

\[
p(x) = \text{probability of responding to the offer given personal characteristics } x
\]

of the form

\[
\log \left( \frac{p(x)}{1 - p(x)} \right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5
\]

was fit (via maximum likelihood) to the 700 training cases yielding results

\[
\hat{\beta}_0 = -3.42, \hat{\beta}_1 = 0.41, \hat{\beta}_2 = 1.76, \hat{\beta}_3 = -0.03, \hat{\beta}_4 = 0.13, \hat{\beta}_5 = 2.09
\]

a) Treating the 700 subjects (that were used to fit the logistic regression model) as a random sample of people (which they WERE NOT) give a linear function \( g(x) \) such that

\[
\hat{f}(x) = I\left[ g(x) > 0 \right]
\]

is an approximately optimal (0-1 loss) classifier (\( y = 0 \) indicating response to the offer).

\[
\hat{f}(x) = I\left[ \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 > 0 \right]
\]

\[
= I\left[ -3.42 + 0.41 x_1 + 1.76 x_2 - 0.03 x_3 + 0.13 x_4 + 2.09 x_5 > 0 \right]
\]

\( \text{or } I\left[ -(\text{something}) < 0 \right] \)

b) Continuing with the logistic regression model, properly adjust your answer to a) in order to provide an approximately optimal (0-1 loss) classifier for a case where a fraction \( \pi_0 = 1/1000 \) of all potential customers would respond to the offer.

All that needs changing from this case-control study is \( \hat{\beta}_0 \) from \( \hat{\beta}_0 = -3.42 \) to \( \hat{\beta}_0 = -3.42 - \ln\left(\frac{96}{604}\right) + \ln\left(\frac{1}{999}\right) \)

\[
= -8.49
\]

Note BTW that since all \( x_j \)'s are 0-1 variables this means that an approximately optimal 0-1 loss classifier NEVER classifies to "respond." But this is a case where losses probably should not be symmetric (missing a responder should be more serious than including a non-responder).
Below are hypothetical counts from a small training set in a 2-class classification problem with a single input, \( x \in \mathbb{R} \) (and we'll treat \( x \) as integer-valued). Although it is easy to determine what an approximately optimal (0-1 loss) classifier is here, for purposes of this exam problem, instead consider use of the AdaBoost.M1 algorithm to produce a classifier. (Use "stumps"/two-node trees that split \( \text{between integer values} \) as base classifiers.) Find an \( M = 3 \) term version of the AdaBoost voting function. (Give \( \hat{f}_1, \alpha_1, \hat{f}_2, \alpha_2, \hat{f}_3, \) and \( \alpha_3 \). The \( \hat{f} \)'s are of the form \( \text{sign}(x - \#) \) or \( \text{sign}(\# - x) \).)

\[
\begin{array}{ccc}
  x = 1 & x = 2 & x = 3 \\
  y = -1 & 5 & 4 & 6 \\
  y = 1 & 3 & 5 & 2
\end{array}
\]

So take \( f_1(x) = \text{sign}(x - 3.5) \) and \( \alpha_1 = \ln \frac{15}{10} = \ln \left( \frac{3}{2} \right) \)

**Step 1:** Splitting at .5 is an error rate of \( \frac{5}{8} \) is possible, at 1.5 an error rate of \( \frac{2}{3} \) is possible, at 2.5 an error rate of \( \frac{1}{3} \) is possible, and at 3.5 an error rate of \( \frac{10}{23} \) is possible.

So take \( f_2(x) = \text{sign}(2.5 - x) \) and \( \alpha_2 = \ln \frac{19}{12} = \ln \left( \frac{3}{2} \right) \)

**Step 2:** Upweight \( y = 1 \) values by a factor of \( \exp \alpha_2 = \frac{3}{2} \) to get the table below:

<table>
<thead>
<tr>
<th>15/2</th>
<th>12/2</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>9/2</td>
<td>15/2</td>
<td>9/2</td>
</tr>
</tbody>
</table>

So take \( f_3(x) = \text{sign}(x - 3.5) \) and \( \alpha_3 = \ln \left( \frac{30}{33} \right) \)

**Step 3:** Upweight cells by a factor of \( \exp \alpha_3 = \frac{3}{2} \) to get the table below:

<table>
<thead>
<tr>
<th>15/2</th>
<th>12/2</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>9/2</td>
<td>15/2</td>
<td>9/2</td>
</tr>
</tbody>
</table>

\( \hat{f}_1(x) = \text{sign}[x - 3.5] \) and \( \alpha_1 = \ln \left( \frac{3}{2} \right) \), \( \hat{f}_2(x) = \text{sign}[2.5 - x] \) and \( \alpha_2 = \ln \left( \frac{3}{2} \right) \)

\( \hat{f}_3(x) = \text{sign}[x - 3.5] \) and \( \alpha_3 = \ln \left( \frac{30}{33} \right) \).
Below is a 9 point data set with $p = 1$. Use agglomerative hierarchical clustering first with single linkage and then with complete linkage to find $K = 3$ clusters in these values. List for each agglomeration step all groups of more than one value. (You don't need to list every value in the data set.)

**Single Linkage**

Group at 1st step: \[\{2.7, 3.0\}\]

Groups(s) at 2nd step:
\[\{2.7, 3.0\}, \{4.9, 5.3\}\]

Groups(s) at 3rd step:
\[\{2.7, 3.0\}, \{4.9, 5.3\}, \{1.0, 1.5\}\]

Groups(s) at 4th step:
\[\{2.7, 3.0\}, \{4.9, 5.3, 5.9\}, \{1.0, 1.5\}\]

Groups(s) at 5th step:
\[\{2.7, 3.0\}, \{4.0, 4.9, 5.3, 5.9\}, \{1.0, 1.5\}\]

Groups(s) at 6th step:
\[\{2.7, 3.0, 4.0, 4.9, 5.3, 5.9\}, \{1.0, 1.5\}\]

**Complete Linkage**

Group at 1st step:
\[\{2.7, 3.0\}\]

Groups(s) at 2nd step:
\[\{2.7, 3.0\}, \{4.9, 5.3\}\]

Groups(s) at 3rd step:
\[\{2.7, 3.0\}, \{4.9, 5.3\}, \{10, 1.5\}\]

Groups(s) at 4th step:
\[\{2.7, 3.0\}, \{4.9, 5.3, 5.9\}, \{10, 1.5\}\]

Groups(s) at 5th step:
\[\{2.7, 3.0, 4.0\}, \{4.9, 5.3, 5.9\}, \{10, 1.5\}\]

Groups(s) at 6th step:
\[\{2.7, 3.0, 4.0\}, \{4.9, 5.3, 5.9, 7.0\}, \{10, 1.5\}\]
This series of questions concerns a (purposely bizarre) 3-class classification problem with $p = 2$. A training set has $N = 300$ cases in it, equally distributed among classes 1, 2, and 3 and is portrayed graphically below.

Below are marginal density estimates for the 3 class-conditional joint densities.
a) Consider a "naïve Bayes" 0-1 loss classifier for the case of \( \pi_1 = \pi_2 = \pi_3 = \frac{1}{3} \).

Show some calculations and say what this classifier would produce for a classification of an input vector \((z, w) = (1, 0)\).

\[
\begin{align*}
\hat{\pi}_1 f_1(1, 0) &= \frac{1}{3} (.96) (.41) \\
\hat{\pi}_2 f_2(1, 0) &= \frac{1}{3} (.39) (.50) \\
\hat{\pi}_3 f_3(1, 0) &= \frac{1}{3} (.41) (.08)
\end{align*}
\]

\( \implies \hat{y} = 2 \) (largest product)

These are the correct answers to the problem as shown on the previous page. A coding error made the plots for \( y = 2 \) wrong. See the corrected version of the Exam for the right plots.

What about the plots on the previous page makes it seem unlikely that the naïve Bayes classifier will be an effective one?

It is clear from the scatterplot that the conditional distributions of \( w \mid z \) for a given class are not close to being free from dependence upon \( z \). So the independence assumption of naïve Bayes classification is not a good one.
b) Below is a series of pairs of plots and options to choose between pairs of them. Circle the number of the plot in each pair most likely fulfill the description.

Which neural net classifier probably has the smaller penalty weight on coefficient size (all else being equal)?

Which neural net classifier probably has the simpler network structure (all else being equal)?
Which (OVO) Gaussian SVM probably has the smaller $C^*$ penalty weight/larger numbers of support vectors (all else being equal)?

Which (OVO) Gaussian SVM classifier is probably built on the kernel with the smaller "standard deviation"/larger "gamma" (all else being equal)?
Which OVO polynomial kernel SVM is probably built on the higher order polynomial kernel (all else being equal)?

Which random forest classifier is probably built on trees with the finer (smaller) groups at the end nodes?
11. In a fake transaction database there are 5 transactions with items from the set of letters A through G. These are:

<table>
<thead>
<tr>
<th>Transaction #</th>
<th>Items Included</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A,B,D,G</td>
</tr>
<tr>
<td>2</td>
<td>B,C,E,G</td>
</tr>
<tr>
<td>3</td>
<td>A,C,D,F</td>
</tr>
<tr>
<td>4</td>
<td>C,D,E,G</td>
</tr>
<tr>
<td>5</td>
<td>A,B,C,G</td>
</tr>
</tbody>
</table>

(a) Find all item sets of support at least 0.4.

\[
\{\{A\}, \{B\}, \{C\}, \{D\}, \{E\}, \{G\}, \{A,B\}, \{A,C\}, \{A,D\}, \{A,G\}, \\
\{B,C\}, \{B,G\}, \{C,D\}, \{C,E\}, \{C,G\}, \{D,G\}, \{E,G\}, \\
\{A,B,G\}, \{B,C,G\}, \{C,D,G\}\]

(b) Based on the 3 item set with the largest support, what are the confidence, expected confidence and lift of the associated conjunctive rules?

**Thus there are 3 such sets. Pick \{B,C,G\}.**

**Rule \{B\} \Rightarrow \{C,G\}**

- Support: \(\frac{2}{5}\)
- Confidence: \(\frac{2}{3}\)
- Expected confidence: \(\frac{2}{5}\)
- Lift: \(\frac{5}{3}\)

**Rule \{C,G\} \Rightarrow \{B\}**

- Support: \(\frac{2}{5}\)
- Confidence: \(\frac{2}{3}\)
- Expected confidence: \(\frac{3}{5}\)
- Lift: \(\frac{5}{3}\)

**Rule \{B,G\} \Rightarrow \{C\}**

- Support: \(\frac{2}{5}\)
- Confidence: \(\frac{2}{3}\)
- Expected confidence: \(\frac{4}{5}\)
- Lift: \(\frac{10}{12}\)

**Rule \{C\} \Rightarrow \{B,G\}**

- Support: \(\frac{2}{5}\)
- Confidence: \(\frac{2}{3}\)
- Expected confidence: \(\frac{3}{5}\)
- Lift: \(\frac{10}{12}\)

**Rule \{B,C\} \Rightarrow \{G\}**

- Support: \(\frac{2}{5}\)
- Confidence: \(\frac{2}{3}\)
- Expected confidence: \(\frac{4}{5}\)
- Lift: \(\frac{5}{4}\)