Problem 1: Decision Trees

Consider this small dataset. We will be using decision trees in this problem to predict whether or not a suggested dish will make it onto the menu in our restaurant.

<table>
<thead>
<tr>
<th>Dish name</th>
<th>Prep Time</th>
<th>Num Ingredients</th>
<th>Num of cooks who liked it</th>
<th>Prep Cost</th>
<th>On the Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crispy Chicken</td>
<td>20</td>
<td>6</td>
<td>10</td>
<td>5</td>
<td>Yes</td>
</tr>
<tr>
<td>Loaded Salad</td>
<td>10</td>
<td>17</td>
<td>15</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>Triple Burger</td>
<td>15</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>Yes</td>
</tr>
<tr>
<td>Quad Burger</td>
<td>15</td>
<td>9</td>
<td>3</td>
<td>9</td>
<td>No</td>
</tr>
<tr>
<td>Banana split</td>
<td>2</td>
<td>10</td>
<td>18</td>
<td>4</td>
<td>Yes</td>
</tr>
</tbody>
</table>

1. How deep does a decision tree like the one we saw in class and on Coursera need to be to perform as a perfect classifier on this dataset?

2. Draw such a tree. (There is more than one right answer)

3. Does normalizing your inputs help with decision trees in the same way it would help with other algorithms? Could it hurt the performance?

4. In this assignment we introduce the idea of decision trees with multiple branches. This can be useful for categorical features where each category slits into a branch or can be used to set \( n - 1 \) thresholds for \( n \) branches for a continuous value.

   The measurement of information gain for such a tree is calculated by

   \[
   H(p^\text{root}_1) - \sum_{i} w^i H(p^i_1)
   \]

   instead of

   \[
   H(p^\text{root}_1) - w^\text{right} H(p^\text{right}_1) + w^\text{left} H(p^\text{left}_1)
   \]

   Draw a tree with three branches trained on our dataset. (There is only one possible tree this time.)

5. Now draw a decision tree with two branches that uses the same logic as the three branched one. (There are two right answers)
As you may have observed from the previous examples adding more branches and adding more depth are practically equivalent, and both can lead to over-fitting.

6. For this problem we revisit the botany example. This is a graph with a feature on the x axis and another feature on the y axis.

What would the decision boundary look like for a decision tree with a maximum depth of two trained on recognising Versicolour?

7. Generally describe what the decision boundary would look like if we used a maximum depth of 20.
Problem 2: Bias-Variance trade-off

1. (5 points) You have a neural network that predicts really well on the validation set. How does increasing
the number of hidden units affect the training error? The validation error? Why?

2. (1 point) Remember that the regularized form of a cost function is:

\[
\text{Regularized Cost} = \text{Cost} + \lambda \times \text{Penalty}, \quad \lambda > 0
\]

How does the regularization penalty change with the value of \( \lambda \)?

3. (4 points) You fit a logistic regression for a classification problem. The training error is low but the
validation error is high. How does regularizing affect the variance? The bias? What can you say about
the training/validation error?

4. (5 points) Adding regularization creates a new hyperparameter: \( \lambda \). Give one method to tune \( \lambda \). Note:
there are several.

Problem 3: Principal Component Analysis

1. Choose True/False. No justification needed.

   (a) (2 points) The goal of PCA is to interpret the underlying structure of the data in terms of the
   principal components that are best at predicting the output variable.

   (b) (2 points) The sum of the PCA eigenvalues is equal to the sum of the variances of the variables.
   (Hint: think about the trace of a covariance matrix.)

   (c) (2 points) Principal component analysis (PCA) can be used with variables of any mathematical
   types: quantitative, qualitative, or a mixture of these types.

2. Choose True/False. No justification needed.

   Remember that PCA is computed as follows:

   Step 1: Compute the covariance matrix: \( \Sigma = \frac{1}{m}X^TX \)

   Step 2: Compute the SVD of \( \Sigma \): \( [U, S, V] = \text{SVD}(\Sigma) \)

   Step 3: Compute \( U_{\text{reduce}} = U[:, 1 : k] \) with \( k \) the number of principal components chosen

   Step 4: Compute the projections: \( Z = XU_{\text{reduce}} \)

   (a) (2 points) Removing columns of \( U \) will still result in an approximation of \( X \), but this will never
   be better than \( X \)

   (b) (2 points) In the case where \( U_{\text{reduce}} = U \), then \( ZZ^T = XX^T \). Hint: think about the properties
   of \( U \).

3. Given a data set, explain how you would use PCA. Specifically, answer these five questions:

   (a) (1 point) Why would you like to use feature normalization?

   (b) (1 point) What do the first two matrices of SVD represent?

   (c) (1 point) What do the eigenvectors represent?

   (d) (1 point) What do the eigenvalues represent?

   (e) (1 point) How do you choose the number of principal components?

4. (2 points) Give one advantage and one disadvantage of using PCA.

5. (3 points) How can PCA be used to speed up supervised learning?
Problem 4: K-means algorithm

Suppose we have the following points in one dimension:

\[ x_1 = 0, \ x_2 = 2, \ x_3 = 3, \ x_4 = 8, \ x_5 = 10 \]

Run the 2-means clustering until convergence with the following initialization:

\[ \mu_1 = -1, \ \mu_2 = 5 \]

Note: in the case of a tie, assign the point to the class with a lower number (i.e. if one point is tied between class 1 and class 2, assign it to class 1).

1. (2 points) Draw a number line to help you visualize what is happening.
2. (2 points) How many iterations did you perform? Note: do not double count! Therefore if iterations \( n \) and \( n+1 \) give the same result, the algorithm converges in \( n \) iterations.
3. (5 points) What is the final assignment?
4. (5 points) What are the final centroids?
5. (5 points) Remember that the loss in the K-means algorithm is given by:

   \[ \text{Loss} = \sum_{i=1}^{m} ||x_i - \mu_{z_i}||^2 \text{ with } z_i \text{ the cluster of point } i \]

   Compute the final loss.
6. (1 point) In the general case, does the K-means algorithm converge to the global minimum?