University of San Diego
SHILEY-MARCOS SCHOOL OF ENGINEERING

Assignment 1.1 Exercises: Short Answer Template

1. **Terminology**:
    - What is the difference between classification and regression?
      Classification is the process of categorizing a data object/instance into one of several discrete, mutually exclusive nominal classes ("red","blue", etc.) while regression is the process of estimating the relationship between some set of independent variables and a response variable. More specifically, regression focuses on the correlation between X and y and optimizes for minimizing residuals.

    - What is the difference between ordinal and multiclass classification?
      Multiclass classification has a zero-sum probability constraint, which requires: Sum(P(Y=y)) = 1

      Ordinal classification, however, simply provides an output of likely classes in order of highest relevance. It is conceivable to receive, for instance, several netflix recommendations where some output function f(X) provides scores f1+f2+f3+f4 >= 1.

    - Explain the difference between a parameter and a hyperparameter.
      A parameter is estimated during the training process – coefficients for linear regression and weights in NN are one example. These are supplied by the model itself. A hyperparameter is supplied by the user and controls the nature of the modeling process itself. Tree depth and number of neighbors in KNN are examples of hyperparameters.

The next two questions refer to the ranking question, see the assignment for details:

- Is predicting the outcome of a match between two players, given their scores, a regression or classification problem? Explain your answer in one sentence.
  This is a classification problem – based on the context provided, it's clear that we're using a probabilistic classifier in a ranking problem, and thus it cannot be a regression problem.

- Are the scores,r_a, r_b, and so on examples of hyperparameters, parameters, or labels? Explain your answer in one sentence.
  Scores are labels/features – they are supplied in the dataset, are not estimated during the modeling process, and are not provided by the user to affect the modeling process.

2.1.1. **Short Answer (Animal Control)**

- What does pruning do to the decision tree?
  Pruning recursively eliminates nodes with the smallest alpha parameter in a decision tree – generally, this results in less overfitting during the training process, and thus reduces the test error rate.

2.2.1. **Short Answer (Text Data)**

- What accuracy do you get with the standard decision tree (no pruning, control of depth)?
  33%

- What does this say about the accuracy of decision trees using text data?
  Decision trees require significant modifications such as pruning, using high-frequency or low-frequency terms, and controlling depth.

- What accuracy do you get by varying the decision tree depth? Do you get higher accuracy with the entire vocabulary, or does choosing a smaller set of words increase accuracy? What does this suggest about text classification, in terms of how much "information" a particular word might be providing the algorithm to make its decision with?

  Without modifying the set of words to use, varying the decision tree depth doesn't seem to provide a stable convergence point for accuracy, and oscillates between 90% and 40%. Modifying the term matrix to use low frequency words only produces an accuracy of 77.5%. Modifying the term matrix to use high-frequency words only produces an expected accuracy of about 78% at a depth of 2-4 nodes. This suggests that many of the terms in a document matrix might be uninformative predictors that add noise to the decision tree.

2.3.1. **Short Answer (Audio Data)**

- In the jupyter notebook, you made plots of the "before" and "after" decision trees. Refer back to these plots for the following questions:
  - Based on how much the tree changes in these two datasets, what can you say about stability of decision trees?
    While the accuracy was not modified at all, the depth of the decision tree (despite keeping the maxdepth parameter constant) shrunk when we omitted a single row from the dataset.

  - If two students develop decision tree classifiers using the audio dataset, are their trees likely to be similar or not?

Given the previous answer, assuming the students both partition the dataset into training and testing randomly, it's highly unlikely the trees will be similar either in depth or in the order of feature splits.

2.4. **Decision Trees Summary:**

- Which dataset has the best accuracy, using decision trees? These aren't exactly apples-to-apples comparisons, but using both your quantitative measures of accuracy and your subjective impression from manipulating the data, which of the three datasets gave the best performance?

Dataset	**Best Accuracy Found**
Animal shelter	**~73.7%**
text	**~78%**
Audio	**~95%**

The audio dataset scored a 95% accuracy for a tree of depth = 5 both with and without a dropped data point, which puts it significantly over the other two datasets.

- What types of data seem to be best suited for decision tree classification? In this question, be more general than just the three data types - explain what might make any new dataset good or bad for decision tree classification. Answer by explaining in terms of inductive bias.

Highly deterministic, low-noise data is generally compatible for decision trees. While we extrapolate beyond the three datasets, it's irrefutable that the audio dataset performed extremely well due to the deterministic nature associated with frequency predictor spaces. Decision trees form the modeling process through information gain (entropy, gini) which seeks to maximize the purity of nodes – that is, trees have an inductive bias towards separating groups into highly imbalanced nodes. Predictor spaces with a highly deterministic nature allow this separation to occur more seamlessly and in fewer nodes relative to the size of P.

3.1. **Short Answer (Cross Validation)**

- You should have a plot with three lines, showing train accuracy, test accuracy, and random predictions. Does it seem that decision trees outperform random guessing on this dataset?

Initially, decision trees vastly outperform the random classifier when we partition n into a 50-50 split. When the test set is 1% of n, however, we find some

instances where, by chance, the random classifier performs just as well as our tree classifier.

- Plot accuracy vs depth again, using the smaller amount of training data. How has the plot changed? How different are the training and test set errors for the decision tree? How different are the accuracy predictions for random guessing, using this smaller test set? Keep in mind you are making fewer predictions, so random guessing might do well at some times.

  After plotting with a smaller test set, we find comparatively more overlap between the random classifier error and the tree error. This is in contrast to the 50-50 partition, wherein the random classifier had an expected error of about 50%, while the tree classifier, on average, had an error less than 10% at any given depth.

- Why would you use leave one out cross validation as opposed to a different method?
  LOOCV is useful in instances where the sample size is small enough (either relative to the size of the population or the predictor space) where partitioning into a test set is unadvisable.

- When would you use a third validation set in addition to using a training and test set?
  In instances where we want tune hyperparameters (and subsequent performance) without interacting directly with the test data.