hw3

1

A. I would choose Weather because it has the greatest information gain.

$$E(\text{Running}) = -\sum_{x \in X} p(x) \cdot \log_2 p(x)$$

$$E(\text{Running}) = -\tfrac{4}{7}\log_2\left(\tfrac{4}{7}\right) - \tfrac{3}{7}\log_2\left(\tfrac{3}{7}\right) = 0.985$$

Weather

$$E(X \mid \text{weather} = \text{sunny}) = -\tfrac{1}{3}\log_2\left(\tfrac{1}{3}\right) - \tfrac{2}{3}\log_2\left(\tfrac{2}{3}\right) = 0.918$$

$$E(X \mid \text{weather} = \text{cloudy}) = -1\log_2(1) = 0$$

$$E(X \mid \text{weather} = \text{Rainy}) = -1\log_2(1) = 0$$

$$E(X \mid \text{weather}) = \tfrac{3}{7}(0.918) + \tfrac{2}{7}(0) + \tfrac{2}{7}(0) = 0.394$$

$$E(\text{weather}) = E(X) - E(X \mid \text{weather}) = 0.592$$

Temperature

$$E(X \mid \text{temp} = \text{low}) = 0$$

$$E(X \mid \text{temp} = \text{med}) = 0$$

$$E(X \mid \text{temp} = \text{high}) = -\tfrac{1}{4}\log_2\left(\tfrac{1}{4}\right) - \tfrac{3}{4}\log_2\left(\tfrac{3}{4}\right) = 0.811$$

$$E(X \mid \text{temp}) = \tfrac{2}{7}(0) + \tfrac{2}{7}(0) + \tfrac{3}{7}(0.811) = 0.464$$

$$E(\text{temp}) = E(X) - E(X \mid \text{temp}) = 0.522$$

Wind Level

$$E(X \mid WL = \text{low}) = 0$$

$$E(X \mid WL = \text{med}) = -\tfrac{1}{3}\log_2\left(\tfrac{1}{3}\right) - \tfrac{2}{3}\log_2\left(\tfrac{2}{3}\right) = 0.918$$

$$E(X \mid WL = \text{high}) = -\tfrac{1}{2}\log_2\left(\tfrac{1}{2}\right) - \tfrac{1}{2}\log_2\left(\tfrac{1}{2}\right) = 1$$

$$E(X \mid WL) = \tfrac{2}{7}(0) + \tfrac{3}{7}(0.918) + \tfrac{2}{7}(1) = 0.679$$

$$E(WL) = E(X) - E(X \mid WL) = 0.306$$

B. Split on Weather -> Temperature -> Wind Level

C. Stop criterion for this process will be
   a. If all datapoints of one split belong to the same class.
   b. If the number of datapoints on one side of a split is below a threshold.

D. We don’t need to standardize features when using a Decision Tree because Decision Trees are not sensitive to the magnitude of the variables. This is because we only look at one dimension at a time.

E. **They are robust to outliers** because they split based on decision boundaries and outliers are contained within one of the sides.

2 True or False, Simple Explanations

1. (T or F) Bagging uses strong learners.
   False
   a. Bagging uses weak learners to create a strong learner.
2. (T or F) The number of predictors to select from at each split in boosting always equal to the number of predictors to select at each split in Random Forest.
   False
   a. Boosting uses all the predictors in a dataset while Random Forests uses a random subset of predictors for each tree.
3. Describe the advantages and disadvantages to taking either a Bagging or Boosting approach to ensemble learning methods?
   a. **Bagging** will **reduce variance** but **increase bias.** Bagging can also help **reduce overfitting** because it is an aggregate of multiple learners, but it **cannot reduce underfitting**.
   b. **Boosting** will **increase accuracy** while **increasing variance**, and boosting needs more **computer power**. It will also **reduce bias**.
4. Explain how a Rectified Linear Unit (ReLU) activation function can potentially address the vanishing gradient issue in training Neural Networks?
   a. Vanishing gradient in Neural Networks is where the derivative of certain nodes goes to 0 as we backpropogate. This occurs when the gradient is very small, which doesn't allow the Neural Network to update. With ReLU, we now have constant and large derivatives which will not lead to a vanishing gradient.

3 Overfitting Mitigation Strategies

For each of the following strategies state whether or not it might help mitigate **overfitting** and why:

1. Using a smaller dataset
    a. **No**
        i. Small datasets lead to overfitting.
2. Allowing your model to train for fewer iterations
    a. **Yes**
        i. Having smaller numbers of iterations can allow the model to settle at a more general solution.
3. Increasing the number of parameters in your model
    a. **No**
        i. Having more parameters will lead to a complex model which leads to overfitting of the model to the training data.
4. Randomly zeroing out half the nodes in a neural network
    a. **Yes**
        i. Yes this is “dropout” and it will reduce the co-dependency of layers in a Neural Network and it will also reduce the proclivity of our model to overfit.
5. Training your model on a GPU or specialized chip instead of a CPU
    a. **No**
        i. This just reduces the training time.
6. Changing the initialization values for your models
    a. **Yes**
        i. This can help our model not stop at a local minima.

4 Principal Component Analysis

A. What are some of the advantages and drawbacks of undertaking dimensionality reduction?
    a. advantages
        i. Reduces the issues related to high dimensionality, and makes the problem contain only the “useful” dimensions.
        ii. Reduces the chance of overfitting.
        iii. Increases interpretability of the model.
    b. drawbacks
        i. Removes some data.
        ii. Information Loss.
        iii. Relies on mean and covariance a lot.
        iv. Less interpretable models.

B. For each of the below situations, state whether or not PCA would work well, and briefly explain why.
    a. Data that has a linear distribution (i.e. linear across different feature dimensions)
        i. Yes
            1. PCA is used to transform data into a linear format, which can be done if the data is already linear.
    b. Data with a non-linear distribution (e.g., data lying on a hyperbolic plane)
        i. No
            1. PCA can be used on non-linear data but it will not have any value because non-linear data can’t be mapped onto a linear principle component.
    c. Data that has been scaled
        i. Yes
            1. The covariance matrix is not affected by scaling because the values are normalized.
    d. Data where each feature is statistically independent of all others
        i. No
            1. PCA uses the covariance matrix which needs correlations between features to work.

5 Perceptron

- Can a perceptron correctly classify this dataset with the proper set of parameters?
  - The perceptron **cannot** classify this dataset.
- If yes, provide an example that would satisfy the model, if not, explain.
  - It cannot because (X1,X2) = (1,0) => 0 and 1.
  - Because the same input gives different outputs a linear separator (i.e. perceptron) cannot separate this data.