Assign 3

Q1. Consider a supervised learning setting where we want to minimize the **expected risk**, also known as the **true risk**, given by:

$$R(f) = \mathbb{E}[(Y - f(X))^2] = \int (y - f(x))^2 p(x, y)\, dx\, dy.$$

where $p(x, y)$ is the joint probability distribution of the input X and output Y.

1. **Find the optimal function** $f^*(x)$ that minimizes the true risk $R(f)$. [1]
2. **Interpretation**: What does the optimal function $f^*(x)$ represent in terms of the conditional distribution $p(y \mid x)$?

Q2. Consider the true model:

$$f(x) = 2x + 3$$

where x is sampled from a uniform distribution $x \sim U[0, 5]$, and the observed response is:

$$y = f(x) + \epsilon, \quad \text{where } \epsilon \sim \mathcal{N}(0, .1).$$

Training Data

Three different training datasets D_1, D_2, D_3 each contain 5 points:

$$D_1 = \{(0.5, 4.2), (1.5, 6.0), (2.5, 7.8), (3.5, 9.1), (4.5, 12.3)\}$$

$$D_2 = \{(0.6, 4.5), (1.6, 5.8), (2.6, 8.0), (3.6, 9.5), (4.6, 11.7)\}$$

$$D_3 = \{(0.4, 4.1), (1.4, 5.9), (2.4, 7.5), (3.4, 9.3), (4.4, 12.0)\}$$

A linear regression model $\hat{f}(x) = ax + b$ is trained separately on each dataset, yielding three different models:

$$\hat{f}_1(x) = 1.9x + 3.5, \quad \hat{f}_2(x) = 2.1x + 3.2, \quad \hat{f}_3(x) = 2.0x + 3.4$$

Bias and Variance Computation

1. Compute the expected prediction function. [.5]
2. Compute the bias at $x = 2$.[.5]
3. Compute the variance at $x = 2$.[.5]
4. Compute the expected squared error. Is it similar to what we get from bias-variance decomposition.[.5]

Decision tree implementation

Q3. Implement a Decision Tree Classifier in Python without using sklearn.tree. [3]

You are allowed to use the following libraries:

- numpy for numerical computations
- pandas for data handling

Your implementation should support:

1. **Binary Splitting**: The tree should split data based on the feature and threshold that minimize impurity.
2. **Impurity Metrics**: Implement Gini Impurity.
3. **Recursive Tree Construction**: Implement a recursive function to build the tree.
4. **Prediction**: Implement a function to classify new data points using the trained tree.
5. **Stopping Conditions**: Include stopping criteria based on:
   - Maximum depth of the tree.
   - Minimum number of samples per leaf node.

1 Example Dataset

Train the decision tree on the following dataset:
Tasks:

- Use Gini Impurity to train the Decision Tree on this dataset.
- Predict whether a new person (Age = 42, Income = Low, Student = No, Credit = Excellent) will buy a computer.

Q4 For the data given in Q3,

- Improve the performance by bagging 10 different trees. Compute the OOB error. [2]

Age	Income	Student	Credit Rating	Buy Computer
25	High	No	Fair	No
30	High	No	Excellent	No
35	Medium	No	Fair	Yes
40	Low	No	Fair	Yes
45	Low	Yes	Fair	Yes
50	Low	Yes	Excellent	No
55	Medium	Yes	Excellent	Yes
60	High	No	Fair	No

Table 1: Training Dataset for Decision Tree

- Improve the performance by bagging 10 different trees but using only two random predictors while building the trees. Compute the OOB error. [2]

Q5. You are tasked with evaluating the performance of a regression model using 5-fold cross-validation on synthetic data generated from a known function. [3]

Tasks

1. **Generate Data:**
   - Sample 100 points x uniformly from the interval $[0, 2\pi]$.
   - Compute target values using the function:

$$y = \sin(x) + \epsilon, \quad \text{where } \epsilon \sim \mathcal{N}(0, 0.1^2)$$

2. Consider models upto degree 4.
3. **Perform 5-Fold Cross-Validation:**
   - Split the dataset into 5 folds.
   - For each fold, train the model on 4 folds and test on the remaining fold.
   - Repeat the process to evaluate all 5 combinations.
   - Use the above process to find the degree of the polynomial to be used.
4. **Visualization:**
   - For the obtained degree
     - Plot the true function $y = \sin(x)$.
     - Plot the noisy training points.
     - Plot the regression model's prediction.