```
## HW 6 JG6615 Due 4/22/22

## Decision Tree Implementation

In this problem we’ll implement decision trees for both
classification and regression. The strategy will be to implement a
generic class, called Decision Tree, which we’ll supply with the loss
function we want to use to make node splitting decisions, as well as
the estimator we’ll use to come up with the prediction associated
with each leaf node. For classification, this prediction could be a
vector of probabilities, but for simplicity we’ll just consider hard
classifications here. We’ll work with the classification and
regression data sets from previous assignments.

# Problem 1
Complete the compute_entropy and compute_gini functions

## Problem 1 Answer:
Done Below
```

0.1 HW 6 JG6615 Due 4/22/22

0.2 Decision Tree Implementation

In this problem we’ll implement decision trees for both classification and regression. The strategy will be to implement a generic class, called Decision Tree, which we’ll supply with the loss function we want to use to make node splitting decisions, as well as the estimator we’ll use to come up with the prediction associated with each leaf node. For classification, this prediction could be a vector of probabilities, but for simplicity we’ll just consider hard classifications here. We’ll work with the classification and regression data sets from previous assignments.

1 Problem 1

Complete the compute_entropy and compute_gini functions

1.1 Problem 1 Answer:

Done Below

```
# Problem 2

Complete the class Decision Tree, given in the skeleton code. The
intended implementation is as follows: Each object of type Decision
Tree represents a single node of the tree. The depth of that node is
represented by the variable self.depth, with the root node having
depth 0. The main job of the fit function is to decide, given the
data provided, how to split the node or whether it should remain a
leaf node. If the node will split, then the splitting feature and
splitting value are recorded, and the left and right subtrees are fit
on the relevant portions of the data. Thus tree-building is a
recursive procedure. We should have as many Decision Tree objects as
there are nodes in the tree. We will not implement pruning here. Some
additional details are given in the skeleton code.

## Problem 2  Answer

Done Below
```

2 Problem 2

Complete the class Decision Tree, given in the skeleton code. The intended implementation is as follows: Each object of type Decision Tree represents a single node of the tree. The depth of that node is represented by the variable self.depth, with the root node having depth 0. The main job of the fit function is to decide, given the data provided, how to split the node or whether it should remain a leaf node. If the node will split, then the splitting feature and splitting value are recorded, and the left and right subtrees are fit on the relevant portions of the data. Thus tree-building is a recursive procedure. We should have as many Decision Tree objects as there are nodes in the tree. We will not implement pruning here. Some additional details are given in the skeleton code.

2.1 Problem 2 Answer

Done Below

```
# Problem 3:

Run the code provided that builds trees for the two-dimensional
classification data. In- clude the results. For debugging, you may
want to compare results with sklearn’s decision tree (code provided
in the skeleton code). For visualization, you’ll need to install
graphviz.

## Problem 3 Answer:

Done below
```

3 Problem 3:

Run the code provided that builds trees for the two-dimensional classification data. In- clude the results. For debugging, you may want to compare results with sklearn’s decision tree (code provided in the skeleton code). For visualization, you’ll need to install graphviz.

3.1 Problem 3 Answer:

Done below

```
# Problem 4

Complete the function mean absolute deviation around median (MAE).
Use the code provided to fit the Regression Tree to the krr dataset
using both the MAE loss and median predictions. Include the plots for
the 6 fits.

## Problem 4 Answer

Done below
```

4 Problem 4

Complete the function mean absolute deviation around median (MAE). Use the code provided to fit the Regression Tree to the krr dataset using both the MAE loss and median predictions. Include the plots for the 6 fits.

4.1 Problem 4 Answer

Done below

```
<div style="page-break-after: always;"></div>
```

4.2 Ensembling

Recall the general gradient boosting algorithm, for a given loss function ℓ and a hypothesis space \cf of regression functions (i.e. functions mapping from the input space to \reals):

0. Initialize $f_0(x)$= 0
1. For m = 1 to M:

```
    a. compute:
```

$$\frac{\delta}{\delta f_{m-1} * (x_j)} \sum_{i}^{n} l(y_i, f_{m-1}(x_i)) \tag{1}$$

In [ ]:

```

```