PA1

May 21, 2019

1 Programming Homework 1

1.1 Instructions

- Do not import other libraries. You are only allowed to use Math, Numpy, Scipy packages which are already imported in the file.
- Please follow the type annotations. There are some type annotations of the parameters of function calls and return values. Please use Python 3.5 or 3.6 (for full support of typing annotations). You can use Numpy/Scipy inside the function. You have to make the functions' return values match the required type.
- In this programming assignment you will to implement **k-Nearest Neighbours and Decision Tree**. We provide the bootstrap code and you are expected to complete the **classes** and **functions**.
- Download all files of PA1 from Vocareum and save in the same folder.
- Only modifications in files {`hw1_knn.py`, `hw1_dt.py`, `utils.py`} will be accepted and graded. All other modifications will be ignored. Submit those three files on Vocareum once you have finished. Which means you need to delete unnecessary files before you submit your work on Vocareum.

1.2 Office Hour:

```
Week 2
Jan. 14th Monday    LVL 2nd Floor-201B  1:00pm to 3:00pm    Cheng-Ju Lin chengjul@usc.edu
Jan. 15th Tuesday   LVL 2nd Floor-201B  1:00pm to 3:00pm    Yang Fang yangfang@usc.edu
Jan. 17th Thursday  LVL 2nd Floor-202B  10:00am to 12:00pm  Yixian Di yixiandi@usc.edu
Week 3
Jan. 22th Tuesday   SAL 125             11:00am to 1:00pm   Ashir Alam ashirala@usc.edu
Jan. 23th Wednesday SAL 125             11:00am to 1:00pm   Ashir Alam ashirala@usc.edu
Jan. 24th Thursday  LVL 2nd Floor-202B  10:00am to 12:00pm  Yixian Di yixiandi@usc.edu
```

1.3 Problem 1: K-nearest neighbor (KNN) for binary classification (50 points)

Some notes In this task, we will use four distance functions: (we removed the vector symbol for simplicity)

- Euclidean distance:

$$d(x, y) = \sqrt{\langle x - y, x - y \rangle}$$

- Inner product distance:
$$d(x,y) = \langle x, y \rangle$$
- Gaussian kernel distance:
$$d(x,y) = -\exp(\frac{1}{2}\langle x-y, x-y \rangle)$$
- Cosine Distance:
$$d(x,y) = \cos(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$$

F1-score is a important metric for binary classification, as sometimes the accuracy metric has the false positive (a good example is in MLAPP book 2.2.3.1 "Example: medical diagnosis", Page 29). We have provided a basic definition. For more you can read 5.7.2.3 from MLAPP book.

1.3.1 Part 1.1.1 Distance Functions

Implement the class in file hw1_knn.py the functions in utils.py
- f1_score - euclidean_distance - inner_product_distance - gaussian_kernel_distance - cosine distance

Simply follow the formula above and finish all these function. You are not allowed to call any package that we did not import for you.

1.3.2 Part 1.1.2 KNN Class

There are following functions you need to implement in KNN class:

1.def train(self, features: List[List[float]], labels: List[int])

In this function, features are simply all training data which is a 2D list with float. For example, if the data looks like the following: Student 1 with features age 25, grade 3.8 and labeled as 0, Student 2 with features age 22, grade 3.0 and labeled as 1, then the feature data should be [[25.0, 3.8], [22.0,3.0]], thus the coresponding label is [0,1]

For KNN, the training process is just loading all the training data. Thus, all you need to do in this function is create some local variable in KNN class to store these data so you can use the data in later process.

2.def get_k_neighbors(self, point: List[float]) -> List[int]:

This function takes one single data point and ask you to find the nearest k neighbours in the training set. You already have your k value, distance function and you just stored all training data in KNN class with the train function.

This function need to return a list of labels of all k neighours.

3.def predict(self, features: List[List[float]]) -> List[int]

The predict function take another 2D list which is all testing data point, Similar to those from train function. In this function, you need process every testing data point, reuse the get_k_neighbours function to find the nearest k neighbours for each testing data point, find the majority of labels for these neighbours as the predict label for that testing data point. Thus, you will get N predicted label for N testing data point.

This function need to return a list of predicted labels for all testing data points.

Once you finished everything above, you can run the next cell to continue.

```
In [1]: # for auto-reloading external modules
        # see http://stackoverflow.com/questions/1907993/autoreload-of-modules-in-ipython
        %load_ext autoreload
        %autoreload 2
        import numpy as np
        from hw1_knn import KNN
        from utils import euclidean_distance, gaussian_kernel_distance, inner_product_distance
        from utils import f1_score, model_selection_without_normalization, model_selection_wit
        distance_funcs = {
            'euclidean': euclidean_distance,
            'gaussian': gaussian_kernel_distance,
            'inner_prod': inner_product_distance,
            'cosine_dist': cosine_sim_distance,
        }
```

```
In [2]: from data import data_processing
        Xtrain, ytrain, Xval, yval, Xtest, ytest = data_processing()

        Xtrain, ytrain
```

```
Out[2]: (array([[ 1., 46.,  1., ...,  1.,  0.,  7.],
                [ 1., 68.,  1., ...,  2.,  0.,  7.],
                [ 1., 35.,  1., ...,  1.,  0.,  7.],
                ...,
                [ 1., 41.,  0., ...,  1.,  0.,  3.],
                [ 1., 61.,  1., ...,  2.,  1.,  7.],
                [ 1., 58.,  0., ...,  1.,  0.,  3.]]),
         array([1., 1., 1., 0., 1., 0., 0., 1., 1., 0., 1., 0., 1., 1., 1., 1., 1.,
                1., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 1., 1., 0., 0., 0.,
                0., 0., 0., 0., 0., 0., 1., 1., 0., 0., 0., 0., 0., 1., 1., 1., 1.,
                0., 1., 0., 0., 0., 1., 0., 0., 1., 1., 1., 1., 1., 0., 0., 1., 0.,
                0., 1., 1., 1., 1., 1., 1., 1., 0., 1., 1., 0., 1., 1., 0., 1., 1.,
                1., 0., 0., 0., 0., 1., 0., 0., 1., 0., 0., 1., 1., 1., 0., 0., 1.,
                0., 1., 0., 1., 0., 0., 0., 0., 1., 0., 1., 0., 0., 0., 1., 1., 0.,
                0., 1., 0., 0., 0., 0., 0., 0., 1., 1., 0., 0., 1., 0., 1., 1., 0.,
                0., 1., 0., 1., 1., 0., 0., 0., 0., 1., 1., 0., 0., 0., 0., 0., 0.,
                0., 0., 0., 0., 1., 0., 1., 0., 0., 1., 1., 0., 0., 1., 1., 1., 1.,
                0., 0., 0., 1., 0., 0., 1., 1., 0., 0., 1., 0., 0., 0., 0., 1., 0.,
                1., 0., 0., 0., 1., 0., 1., 1., 1., 0., 1., 0., 1., 1., 1., 0., 0.,
                1., 0., 1., 1., 1., 1., 1., 1., 0., 1., 1., 1., 0., 0., 0., 1., 0.,
                1., 1., 0., 1., 0., 1., 1., 0., 0., 0., 1., 0., 0., 1., 1., 0., 1.,
                1., 0., 1., 0.]))
```

1.3.3 Part 1.1.3 Model selection

In this section, you need to implement the following function:

1.def model_selection_without_normalization(distance_funcs, Xtrain, ytrain, Xval, yval)

In this part, you should try different distance function you implemented in part 1.1, and find the best k. Use k range from 1 to 30 and increment by 2. We will use f1-score to compare different models.

THis function take the following parameter:

distance_funcs: dictionary of distance funtion you will use to calculate the distance. Make sure you loop over all distance function for each data point and each k value.

Xtrain: List[List[int]] training data set to train your KNN model

ytrain: List[int] train labels to train your KNN model

Xval: List[List[int]] validation data set you will use on your KNN predict function to produce predicted labels and tune k and distance function.

yval: List[int] validation labels

This function need to return the following:

best_model: an instance of KNN

best_k: best k choosed for best_model

best_func: name of best function choosed for best_model

Thus, the function only return one set of model, k and function.

Once you finished everything above, you can run the next cell to continue.

Note: When there is a tie, chose model based on the following priorities: Then check distance function [euclidean > gaussian > inner_prod > cosine_dist]; If they have same distance fuction, choose model who have a less k.

```
In [4]: best_model, best_k, best_function = model_selection_without_normalization(distance_func
```

1.3.4 Part 1.2 Data transformation

We are going to add one more step (data transformation) in the data processing part and see how it works. Sometimes, normalization plays an important role to make a machine learning model work (check term "Feature scaling" in wiki).

Here, we take two different data transformation approaches.

Normalizing the feature vector This one is simple but some times may work well. Given a feature vector x, the normalized feature vector is given by

$$x' = \frac{x}{\sqrt{\langle x, x \rangle}}$$

If a vector is a all-zero vector, we let the normalized vector also be a all-zero vector.

Min-max scaling the feature matrix The above normalization is data independent, that is to say, the output of the normalization function doesn't depend on the rest training data. However, sometimes it would be helpful to do data dependent normalization. One thing to note is that, when doing data dependent normalization, we can only use training data, as the test data is assumed to be unknown during training (at least for most classification tasks).

The min-max scaling works as follows: after min-max scaling, all values of training data's feature vectors are in the given range. Note that this doesn't mean the values of the validation/test data's fea- tures are all in that range, because the validation/test data may have dif- ferent distribution as the training data.

Implement the functions in utils.py

1.normalize

normalize the feature vector for each sample . For example, if the input features = [[3, 4], [1, -1], [0, 0]], the output should be [[0.6, 0.8], [0.707107, -0.707107], [0, 0]]

2.min_max_scale

normalize the feature vector for each sample . For example, if the input features = [[2, -1], [-1, 5], [0, 0]], the output should be [[1, 0], [0, 1], [0.333333, 0.16667]]

```
In [5]: from utils import NormalizationScaler, MinMaxScaler

        scaling_classes = {
            'min_max_scale': MinMaxScaler,
            'normalize': NormalizationScaler,
        }
```

Model selection This part will be similar to Part 1.1.3 except before pass your traing and validation data to KNN model, you need to create the normalized data using these two scaller to transform your data, both training and validation. Again, we will use f1-score to compare different models.

1.def model_selection_with_transformation(distance_funcs, scaling_classes, Xtrain, ytrain, Xval, yval)

The function take the following input:

distance_funcs: dictionary of distance funtion you will use to calculate the distance. Make sure you loop over all distance function for each data point and each k value.

scaling_classes: diction of scalers you will use to normalized your data

Xtrain: List[List[int]] training data set to train your KNN model

ytrain: List[int] train labels to train your KNN model

Xval: List[List[int]] validation data set you will use on your KNN predict function to produce predicted labels and tune your k, distance function and scaler.

yval: List[int] validation labels

This function need to return the following:

best_model: an instance of KNN

best_k: best k choosed for best_model

best_func: name of best function choosed for best_model

best_scaler: name of the scaler choosed for best_model

Thus, the function only return one set of model, k, function name and scaler name.

Once you finished everything above, you can run the next cell to continue.

Note: When there is a tie, chose model based on the following priorities: For normalization, [min_max_scale > normalize]; Then check distance function [euclidean > gaussian > inner_prod > cosine_dist]; If they have same distance fuction, choose model who have a less k.

```
In [6]: best_model, best_k, best_function, best_scaler = model_selection_with_transformation(di

/Users/yangfang/Documents/csci567-ML/Design/PA1_finial/utils.py:187: RuntimeWarning: invalid va
  norm[norm < 1e-6] = 1
```

1.4 Grading Guideline for KNN

1. UTILS function: 15 points

2. 2 functions in hw1_Knn (10 points- 5 each)
3. Finding best K before scaling - 10 points
4. Finding best K after scaling - 10 points
5. Doing classification of the data - 5 points

2 Problem 2: Decision Tree (50 points)

- Remember from lecture, we learned that we can use decision tree to solve classification and regression problem. Mostly we focus on classification.
- In problem 1 we used KNN to do classification. We could use decision tree algorithm to do the same job.
- For Decision Tree, we will implement ID3 algorithm. It's garanteed that all features are discrete. ## Part 2.1 Implementation ### 2.1.1
- In ID3 algorithm, we use Entropy to measure the uncertainty in the data set. We use Information Gain to measure the quality of a split.
- Entropy: H(S)=$\sum_{xX} -p(x)log_2p(x)$
- Information_Gain: IG(S,A) = H(S)-$\sum_{tT} p(t)H(T)$ = H(S) - H(S | A)
- see more detail on ID3 Algorithm In this section, you need to implement Information_Gain function on utils.py.

```
def Information_Gain(S, branches):
# calculate information_gain according to branches seperated by one feature
# input:
    -S: float Entropy of current state
    -branches: List[List[int]] for a specific attribute, number of cases belongs to each attril
# return: float
```

2.0.1 2.1.2

- In ID3 algorithm, we use the largest information gain to split the set S. Please consult the Lecture 2 notes page 23.
- Implement TreeNode split function and TreeNode predict function in hw1_dt.py:
    - TreeNode.split

  In the TreeNode class, the features variable means all the points in current TreeNode, and the labels variable means the corresponding labels for all data. The children variable is a list of TreeNode after split the current node based on the best attributs. This should be a recursive process that once we call the split function, the TreeNode will keep spliting untill we get the whole tree structure.

  Note: when there is a tie of information gain when comparing the attributes, always choose the attribute which has more attribute values. If they are all same, use the one with small index. Also build your child list with increasing order of attribute value.

    - TreeNode.predict

This function will be called once we create the tree structure by the split function. It will take one single data point as a parameter, your code should process that data point and go through your tree to a leaf and make prediction. Thus, this function need to return a predicted lable.

- You no longer need to implement Decision Tree predict and train function in hw1_dt.py:
  - DecisionTree.train
  - DecisionTree.predict

  We will provide these two function in the statercode. Reading the train and predict function should help you understanding funcitons that you need to implement.

2.1 Part 2.2 Sanity Test

Do the following steps, as a simple test to check your algorithm works well - Load training data (features and values) from function data.sample_decision_tree_data. - Create a Decision Tree based on training data. - Load test data from function data.sample_decision_tree_test. - Test the prediction function of your algorithm.

```
In [7]: import data
        import hw1_dt as decision_tree
        import utils as Utils
        from sklearn.metrics import accuracy_score

        features, labels = data.sample_decision_tree_data()

        # build the tree
        dTree = decision_tree.DecisionTree()
        dTree.train(features, labels)

        # print
        Utils.print_tree(dTree)
```

```
branch 0{
        deep: 0
        num of samples for each class: 2 : 2
        split by dim 0
        branch 0->0{
                deep: 1
                num of samples for each class: 1
                class: 0
        }
        branch 0->1{
                deep: 1
                num of samples for each class: 1 : 1
                split by dim 0
                branch 0->1->0{
                        deep: 2
                        num of samples for each class: 1
                        class: 0
                }
                branch 0->1->1{
                        deep: 2
                        num of samples for each class: 1
                        class: 1
                }
        }
        branch 0->2{
                deep: 1
                num of samples for each class: 1
                class: 1
        }
}
```

```
In [8]: # data
        X_test, y_test = data.sample_decision_tree_test()

        # testing
        y_est_test = dTree.predict(X_test)
        test_accu = accuracy_score(y_est_test, y_test)
        print('test_accu', test_accu)
```

```
test_accu 1.0
```

2.2 Part 2.3 Train and Predict

2.2.1 2.3.1

- Load data (features and values) from function data.load_decision_tree_data.
- Train your decision tree

```
In [9]: #load data
        X_train, y_train, X_test, y_test = data.load_decision_tree_data()

        # set classifier
        dTree = decision_tree.DecisionTree()

        # training
        dTree.train(X_train.tolist(), y_train.tolist())
```

2.2.2 2.3.2

- Print your decision tree.

```
In [10]: # print
         Utils.print_tree(dTree)
```

```
branch 0{
        deep: 0
        num of samples for each class: 845 : 260 : 2
        split by dim 5
        branch 0->0{
                deep: 1
                num of samples for each class: 369
                class: 0
        }
        branch 0->1{
                deep: 1
                num of samples for each class: 273 : 96
                split by dim 3
                branch 0->1->0{
                        deep: 2
                        num of samples for each class: 123
                        class: 0
                }
                branch 0->1->1{
                        deep: 2
                        num of samples for each class: 78 : 45
                        split by dim 3
                        branch 0->1->1->0{
                                deep: 3
                                num of samples for each class: 40 : 1
                                split by dim 0
                                branch 0->1->1->0->0{
                                        deep: 4
                                        num of samples for each class: 16
                                        class: 0
                                }
                                branch 0->1->1->0->1{
                                        deep: 4
                                        num of samples for each class: 8 : 1
                                        split by dim 0
                                        branch 0->1->1->0->1->0{
                                                deep: 5
                                                num of samples for each class: 4
                                                class: 0
                                        }
                                        branch 0->1->1->0->1->1{
                                                deep: 5
                                                num of samples for each class: 1
                                                class: 1
                                        }
                                        branch 0->1->1->0->1->2{
                                                deep: 5
                                                num of samples for each class: 4
                                                class: 0
                                        }
                                }
                                branch 0->1->1->0->2{
                                        deep: 4
                                        num of samples for each class: 16
                                        class: 0
                                }
                        }
                        branch 0->1->1->1{
                                deep: 3
                                num of samples for each class: 26 : 15
                                split by dim 2
                                branch 0->1->1->1->0{
                                        deep: 4
                                        num of samples for each class: 10 : 1
                                        split by dim 0
                                        branch 0->1->1->1->0->0{
                                                deep: 5
                                                num of samples for each class: 4
                                                class: 0
                                        }
                                        branch 0->1->1->1->0->1{
                                                deep: 5
                                                num of samples for each class: 2 : 1
                                                split by dim 0
                                                branch 0->1->1->1->0->1->0{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 0
                                                }
                                                branch 0->1->1->1->0->1->1{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 1
                                                }
                                                branch 0->1->1->1->0->1->2{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 0
                                                }
                                        }
                                        branch 0->1->1->1->0->2{
                                                deep: 5
                                                num of samples for each class: 4
                                                class: 0
                                        }
                                }
```

```
branch 0->1->1->1->1{
        deep: 4
        num of samples for each class: 10
        class: 0
}
branch 0->1->1->1->2{
        deep: 4
        num of samples for each class: 3 : 7
        split by dim 0
        branch 0->1->1->1->2->0{
                deep: 5
                num of samples for each class: 2 : 2
                split by dim 0
                branch 0->1->1->1->2->0->0{
                        deep: 6
                        num of samples for each class: 1 : 1
                        class: 0
                }
                branch 0->1->1->1->2->0->1{
                        deep: 6
                        num of samples for each class: 1
                        class: 1
                }
                branch 0->1->1->1->2->0->2{
                        deep: 6
                        num of samples for each class: 1
                        class: 0
                }
        }
        branch 0->1->1->1->2->1{
                deep: 5
                num of samples for each class: 2
                class: 1
        }
        branch 0->1->1->1->2->2{
                deep: 5
                num of samples for each class: 1 : 3
                split by dim 0
                branch 0->1->1->1->2->2->0{
                        deep: 6
                        num of samples for each class: 1 : 1
                        class: 0
                }
                branch 0->1->1->1->2->2->1{
                        deep: 6
                        num of samples for each class: 1
                        class: 1
                }
```

```
                                                branch 0->1->1->1->2->2->2{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 1
                                                }
                                        }
                                }
                                branch 0->1->1->1->3{
                                        deep: 4
                                        num of samples for each class: 3 : 7
                                        split by dim 0
                                        branch 0->1->1->1->3->0{
                                                deep: 5
                                                num of samples for each class: 2 : 2
                                                split by dim 0
                                                branch 0->1->1->1->3->0->0{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                                branch 0->1->1->1->3->0->1{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 1
                                                }
                                                branch 0->1->1->1->3->0->2{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 0
                                                }
                                        }
                                        branch 0->1->1->1->3->1{
                                                deep: 5
                                                num of samples for each class: 2
                                                class: 1
                                        }
                                        branch 0->1->1->1->3->2{
                                                deep: 5
                                                num of samples for each class: 1 : 3
                                                split by dim 0
                                                branch 0->1->1->1->3->2->0{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                                branch 0->1->1->1->3->2->1{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 1
                                                }
                                                branch 0->1->1->1->3->2->2{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 1
                                                }
                                        }
                                }
                        }
                        branch 0->1->1->2{
                                deep: 3
                                num of samples for each class: 12 : 29
                                split by dim 0
                                branch 0->1->1->2->0{
                                        deep: 4
                                        num of samples for each class: 8 : 8
                                        split by dim 0
                                        branch 0->1->1->2->0->0{
                                                deep: 5
                                                num of samples for each class: 4 : 4
                                                split by dim 0
                                                branch 0->1->1->2->0->0->0{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                                branch 0->1->1->2->0->0->1{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                                branch 0->1->1->2->0->0->2{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                                branch 0->1->1->2->0->0->3{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                        }
                                        branch 0->1->1->2->0->1{
                                                deep: 5
                                                num of samples for each class: 4
                                                class: 1
                                        }
                                branch 0->1->1->2->0->2{
                                        deep: 5
                                        num of samples for each class: 4
                                        class: 0
                                }
                        }
                        branch 0->1->1->2->1{
                                deep: 4
                                num of samples for each class: 9
                                class: 1
                        }
                        branch 0->1->1->2->2{
                                deep: 4
                                num of samples for each class: 4 : 12
                                split by dim 0
                                branch 0->1->1->2->2->0{
                                        deep: 5
                                        num of samples for each class: 4 : 4
                                        split by dim 0
                                        branch 0->1->1->2->2->0->0{
                                                deep: 6
                                                num of samples for each class: 1 : 1
                                                class: 0
                                        }
                                        branch 0->1->1->2->2->0->1{
                                                deep: 6
                                                num of samples for each class: 1 : 1
                                                class: 0
                                        }
                                        branch 0->1->1->2->2->0->2{
                                                deep: 6
                                                num of samples for each class: 1 : 1
                                                class: 0
                                        }
                                        branch 0->1->1->2->2->0->3{
                                                deep: 6
                                                num of samples for each class: 1 : 1
                                                class: 0
                                        }
                                }
                                branch 0->1->1->2->2->1{
                                        deep: 5
                                        num of samples for each class: 4
                                        class: 1
                                }
                                branch 0->1->1->2->2->2{
                                        deep: 5
                                        num of samples for each class: 4
```

```
                                class: 1
                        }
                }
        }
}
branch 0->1->2{
        deep: 2
        num of samples for each class: 72 : 51
        split by dim 3
        branch 0->1->2->0{
                deep: 3
                num of samples for each class: 41
                class: 0
        }
        branch 0->1->2->1{
                deep: 3
                num of samples for each class: 19 : 22
                split by dim 2
                branch 0->1->2->1->0{
                        deep: 4
                        num of samples for each class: 10 : 1
                        split by dim 0
                        branch 0->1->2->1->0->0{
                                deep: 5
                                num of samples for each class: 4
                                class: 0
                        }
                        branch 0->1->2->1->0->1{
                                deep: 5
                                num of samples for each class: 2 : 1
                                split by dim 0
                                branch 0->1->2->1->0->1->0{
                                        deep: 6
                                        num of samples for each class: 1
                                        class: 0
                                }
                                branch 0->1->2->1->0->1->1{
                                        deep: 6
                                        num of samples for each class: 1
                                        class: 1
                                }
                                branch 0->1->2->1->0->1->2{
                                        deep: 6
                                        num of samples for each class: 1
                                        class: 0
                                }
                        }
                        branch 0->1->2->1->0->2{
                                deep: 5
                                num of samples for each class: 4
                                class: 0
                        }
                }
                branch 0->1->2->1->1{
                        deep: 4
                        num of samples for each class: 3 : 7
                        split by dim 0
                        branch 0->1->2->1->1->0{
                                deep: 5
                                num of samples for each class: 2 : 2
                                split by dim 0
                                branch 0->1->2->1->1->0->0{
                                        deep: 6
                                        num of samples for each class: 1 : 1
                                        class: 0
                                }
                                branch 0->1->2->1->1->0->1{
                                        deep: 6
                                        num of samples for each class: 1
                                        class: 1
                                }
                                branch 0->1->2->1->1->0->2{
                                        deep: 6
                                        num of samples for each class: 1
                                        class: 0
                                }
                        }
                        branch 0->1->2->1->1->1{
                                deep: 5
                                num of samples for each class: 2
                                class: 1
                        }
                        branch 0->1->2->1->1->2{
                                deep: 5
                                num of samples for each class: 1 : 3
                                split by dim 0
                                branch 0->1->2->1->1->2->0{
                                        deep: 6
                                        num of samples for each class: 1 : 1
                                        class: 0
                                }
                                branch 0->1->2->1->1->2->1{
                                        deep: 6
                                        num of samples for each class: 1
                                        class: 1
                                }
```

```
                                                branch 0->1->2->1->1->2->2{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 1
                                                }
                                        }
                                }
                                branch 0->1->2->1->2{
                                        deep: 4
                                        num of samples for each class: 3 : 7
                                        split by dim 0
                                        branch 0->1->2->1->2->0{
                                                deep: 5
                                                num of samples for each class: 2 : 2
                                                split by dim 0
                                                branch 0->1->2->1->2->0->0{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                                branch 0->1->2->1->2->0->1{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 1
                                                }
                                                branch 0->1->2->1->2->0->2{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 0
                                                }
                                        }
                                        branch 0->1->2->1->2->1{
                                                deep: 5
                                                num of samples for each class: 2
                                                class: 1
                                        }
                                        branch 0->1->2->1->2->2{
                                                deep: 5
                                                num of samples for each class: 1 : 3
                                                split by dim 0
                                                branch 0->1->2->1->2->2->0{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                                branch 0->1->2->1->2->2->1{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 1
                                                }
                                                branch 0->1->2->1->2->2->2{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 1
                                                }
                                        }
                                }
                                branch 0->1->2->1->3{
                                        deep: 4
                                        num of samples for each class: 3 : 7
                                        split by dim 0
                                        branch 0->1->2->1->3->0{
                                                deep: 5
                                                num of samples for each class: 2 : 2
                                                split by dim 0
                                                branch 0->1->2->1->3->0->0{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                                branch 0->1->2->1->3->0->1{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 1
                                                }
                                                branch 0->1->2->1->3->0->2{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 0
                                                }
                                        }
                                        branch 0->1->2->1->3->1{
                                                deep: 5
                                                num of samples for each class: 2
                                                class: 1
                                        }
                                        branch 0->1->2->1->3->2{
                                                deep: 5
                                                num of samples for each class: 1 : 3
                                                split by dim 0
                                                branch 0->1->2->1->3->2->0{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                                branch 0->1->2->1->3->2->1{
```

```
                                deep: 6
                                num of samples for each class: 1
                                class: 1
                        }
                        branch 0->1->2->1->3->2->2{
                                deep: 6
                                num of samples for each class: 1
                                class: 1
                        }
                }
        }
}
branch 0->1->2->2{
        deep: 3
        num of samples for each class: 12 : 29
        split by dim 0
        branch 0->1->2->2->0{
                deep: 4
                num of samples for each class: 8 : 8
                split by dim 0
                branch 0->1->2->2->0->0{
                        deep: 5
                        num of samples for each class: 4 : 4
                        split by dim 0
                        branch 0->1->2->2->0->0->0{
                                deep: 6
                                num of samples for each class: 1 : 1
                                class: 0
                        }
                        branch 0->1->2->2->0->0->1{
                                deep: 6
                                num of samples for each class: 1 : 1
                                class: 0
                        }
                        branch 0->1->2->2->0->0->2{
                                deep: 6
                                num of samples for each class: 1 : 1
                                class: 0
                        }
                        branch 0->1->2->2->0->0->3{
                                deep: 6
                                num of samples for each class: 1 : 1
                                class: 0
                        }
                }
                branch 0->1->2->2->0->1{
                        deep: 5
                        num of samples for each class: 4
```

```
                                                class: 1
                                        }
                                        branch 0->1->2->2->0->2{
                                                deep: 5
                                                num of samples for each class: 4
                                                class: 0
                                        }
                                }
                                branch 0->1->2->2->1{
                                        deep: 4
                                        num of samples for each class: 9
                                        class: 1
                                }
                                branch 0->1->2->2->2{
                                        deep: 4
                                        num of samples for each class: 4 : 12
                                        split by dim 0
                                        branch 0->1->2->2->2->0{
                                                deep: 5
                                                num of samples for each class: 4 : 4
                                                split by dim 0
                                                branch 0->1->2->2->2->0->0{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                                branch 0->1->2->2->2->0->1{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                                branch 0->1->2->2->2->0->2{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                                branch 0->1->2->2->2->0->3{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                        }
                                        branch 0->1->2->2->2->1{
                                                deep: 5
                                                num of samples for each class: 4
                                                class: 1
                                        }
                                        branch 0->1->2->2->2->2{
                                                deep: 5
                                                num of samples for each class: 4
                                                class: 1
                                        }
                                }
                        }
                }
        }
        branch 0->2{
                deep: 1
                num of samples for each class: 203 : 164 : 2
                split by dim 3
                branch 0->2->0{
                        deep: 2
                        num of samples for each class: 123
                        class: 0
                }
                branch 0->2->1{
                        deep: 2
                        num of samples for each class: 36 : 86 : 1
                        split by dim 0
                        branch 0->2->1->0{
                                deep: 3
                                num of samples for each class: 24 : 24
                                split by dim 0
                                branch 0->2->1->0->0{
                                        deep: 4
                                        num of samples for each class: 12 : 12
                                        split by dim 0
                                        branch 0->2->1->0->0->0{
                                                deep: 5
                                                num of samples for each class: 3 : 3
                                                split by dim 0
                                                branch 0->2->1->0->0->0->0{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                                branch 0->2->1->0->0->0->1{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                                branch 0->2->1->0->0->0->2{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
```

```
}
branch 0->2->1->0->0->1{
        deep: 5
        num of samples for each class: 3 : 3
        split by dim 0
        branch 0->2->1->0->0->1->0{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
        branch 0->2->1->0->0->1->1{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
        branch 0->2->1->0->0->1->2{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
}
branch 0->2->1->0->0->2{
        deep: 5
        num of samples for each class: 3 : 3
        split by dim 0
        branch 0->2->1->0->0->2->0{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
        branch 0->2->1->0->0->2->1{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
        branch 0->2->1->0->0->2->2{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
}
branch 0->2->1->0->0->3{
        deep: 5
        num of samples for each class: 3 : 3
        split by dim 0
        branch 0->2->1->0->0->3->0{
                deep: 6
                num of samples for each class: 1 : 1
```

```
                                                class: 0
                                        }
                                        branch 0->2->1->0->0->3->1{
                                                deep: 6
                                                num of samples for each class: 1 : 1
                                                class: 0
                                        }
                                        branch 0->2->1->0->0->3->2{
                                                deep: 6
                                                num of samples for each class: 1 : 1
                                                class: 0
                                        }
                                }
                        }
                        branch 0->2->1->0->1{
                                deep: 4
                                num of samples for each class: 12
                                class: 1
                        }
                        branch 0->2->1->0->2{
                                deep: 4
                                num of samples for each class: 12
                                class: 0
                        }
                }
                branch 0->2->1->1{
                        deep: 3
                        num of samples for each class: 26 : 1
                        split by dim 0
                        branch 0->2->1->1->0{
                                deep: 4
                                num of samples for each class: 12
                                class: 1
                        }
                        branch 0->2->1->1->1{
                                deep: 4
                                num of samples for each class: 2 : 1
                                split by dim 1
                                branch 0->2->1->1->1->0{
                                        deep: 5
                                        num of samples for each class: 1
                                        class: 1
                                }
                                branch 0->2->1->1->1->1{
                                        deep: 5
                                        num of samples for each class: 1
                                        class: 1
                                }
                                branch 0->2->1->1->1->2{
                                        deep: 5
                                        num of samples for each class: 1
                                        class: 3
                                }
                        }
                        branch 0->2->1->1->2{
                                deep: 4
                                num of samples for each class: 12
                                class: 1
                        }
                }
                branch 0->2->1->2{
                        deep: 3
                        num of samples for each class: 12 : 36
                        split by dim 0
                        branch 0->2->1->2->0{
                                deep: 4
                                num of samples for each class: 12 : 12
                                split by dim 0
                                branch 0->2->1->2->0->0{
                                        deep: 5
                                        num of samples for each class: 3 : 3
                                        split by dim 0
                                        branch 0->2->1->2->0->0->0{
                                                deep: 6
                                                num of samples for each class: 1 : 1
                                                class: 0
                                        }
                                        branch 0->2->1->2->0->0->1{
                                                deep: 6
                                                num of samples for each class: 1 : 1
                                                class: 0
                                        }
                                        branch 0->2->1->2->0->0->2{
                                                deep: 6
                                                num of samples for each class: 1 : 1
                                                class: 0
                                        }
                                }
                                branch 0->2->1->2->0->1{
                                        deep: 5
                                        num of samples for each class: 3 : 3
                                        split by dim 0
                                        branch 0->2->1->2->0->1->0{
                                                deep: 6
                                                num of samples for each class: 1 : 1
                                                class: 0
                                        }
                                        branch 0->2->1->2->0->1->1{
                                                deep: 6
                                                num of samples for each class: 1 : 1
                                                class: 0
                                        }
                                        branch 0->2->1->2->0->1->2{
                                                deep: 6
                                                num of samples for each class: 1 : 1
                                                class: 0
                                        }
                                }
                                branch 0->2->1->2->0->2{
                                        deep: 5
                                        num of samples for each class: 3 : 3
                                        split by dim 0
                                        branch 0->2->1->2->0->2->0{
                                                deep: 6
                                                num of samples for each class: 1 : 1
                                                class: 0
                                        }
                                        branch 0->2->1->2->0->2->1{
                                                deep: 6
                                                num of samples for each class: 1 : 1
                                                class: 0
                                        }
                                        branch 0->2->1->2->0->2->2{
                                                deep: 6
                                                num of samples for each class: 1 : 1
                                                class: 0
                                        }
                                }
                                branch 0->2->1->2->0->3{
                                        deep: 5
                                        num of samples for each class: 3 : 3
                                        split by dim 0
                                        branch 0->2->1->2->0->3->0{
                                                deep: 6
                                                num of samples for each class: 1 : 1
                                                class: 0
                                        }
                                        branch 0->2->1->2->0->3->1{
                                                deep: 6
                                                num of samples for each class: 1 : 1
                                                class: 0
                                        }
                                        branch 0->2->1->2->0->3->2{
                                                deep: 6
```

```
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                        }
                                }
                                branch 0->2->1->2->1{
                                        deep: 4
                                        num of samples for each class: 12
                                        class: 1
                                }
                                branch 0->2->1->2->2{
                                        deep: 4
                                        num of samples for each class: 12
                                        class: 1
                                }
                        }
                }
                branch 0->2->2{
                        deep: 2
                        num of samples for each class: 44 : 78 : 1
                        split by dim 0
                        branch 0->2->2->0{
                                deep: 3
                                num of samples for each class: 26 : 22
                                split by dim 0
                                branch 0->2->2->0->0{
                                        deep: 4
                                        num of samples for each class: 13 : 11
                                        split by dim 0
                                        branch 0->2->2->0->0->0{
                                                deep: 5
                                                num of samples for each class: 4 : 2
                                                split by dim 0
                                                branch 0->2->2->0->0->0->0{
                                                        deep: 6
                                                        num of samples for each class: 2
                                                        class: 0
                                                }
                                                branch 0->2->2->0->0->0->1{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                                branch 0->2->2->0->0->0->2{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
```

```
}
branch 0->2->2->0->0->1{
        deep: 5
        num of samples for each class: 3 : 3
        split by dim 0
        branch 0->2->2->0->0->1->0{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
        branch 0->2->2->0->0->1->1{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
        branch 0->2->2->0->0->1->2{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
}
branch 0->2->2->0->0->2{
        deep: 5
        num of samples for each class: 3 : 3
        split by dim 0
        branch 0->2->2->0->0->2->0{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
        branch 0->2->2->0->0->2->1{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
        branch 0->2->2->0->0->2->2{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
}
branch 0->2->2->0->0->3{
        deep: 5
        num of samples for each class: 3 : 3
        split by dim 0
        branch 0->2->2->0->0->3->0{
                deep: 6
                num of samples for each class: 1 : 1
```

```
                        class: 0
                }
                branch 0->2->2->0->0->3->1{
                        deep: 6
                        num of samples for each class: 1 : 1
                        class: 0
                }
                branch 0->2->2->0->0->3->2{
                        deep: 6
                        num of samples for each class: 1 : 1
                        class: 0
                }
        }
}
branch 0->2->2->0->1{
        deep: 4
        num of samples for each class: 1 : 11
        split by dim 0
        branch 0->2->2->0->1->0{
                deep: 5
                num of samples for each class: 1 : 2
                split by dim 0
                branch 0->2->2->0->1->0->0{
                        deep: 6
                        num of samples for each class: 1
                        class: 0
                }
                branch 0->2->2->0->1->0->1{
                        deep: 6
                        num of samples for each class: 1
                        class: 1
                }
                branch 0->2->2->0->1->0->2{
                        deep: 6
                        num of samples for each class: 1
                        class: 1
                }
        }
        branch 0->2->2->0->1->1{
                deep: 5
                num of samples for each class: 3
                class: 1
        }
        branch 0->2->2->0->1->2{
                deep: 5
                num of samples for each class: 3
                class: 1
        }
```

```
                branch 0->2->2->0->1->3{
                        deep: 5
                        num of samples for each class: 3
                        class: 1
                }
        }
        branch 0->2->2->0->2{
                deep: 4
                num of samples for each class: 12
                class: 0
        }
}
branch 0->2->2->1{
        deep: 3
        num of samples for each class: 3 : 23 : 1
        split by dim 1
        branch 0->2->2->1->0{
                deep: 4
                num of samples for each class: 3 : 5 : 1
                split by dim 1
                branch 0->2->2->1->0->0{
                        deep: 5
                        num of samples for each class: 3
                        class: 0
                }
                branch 0->2->2->1->0->1{
                        deep: 5
                        num of samples for each class: 3
                        class: 1
                }
                branch 0->2->2->1->0->2{
                        deep: 5
                        num of samples for each class: 2 : 1
                        split by dim 0
                        branch 0->2->2->1->0->2->0{
                                deep: 6
                                num of samples for each class: 1
                                class: 1
                        }
                        branch 0->2->2->1->0->2->1{
                                deep: 6
                                num of samples for each class: 1
                                class: 3
                        }
                        branch 0->2->2->1->0->2->2{
                                deep: 6
                                num of samples for each class: 1
                                class: 1
                        }
                }
        }
        branch 0->2->2->1->1{
                deep: 4
                num of samples for each class: 6
                class: 1
        }
        branch 0->2->2->1->2{
                deep: 4
                num of samples for each class: 6
                class: 1
        }
        branch 0->2->2->1->3{
                deep: 4
                num of samples for each class: 6
                class: 1
        }
}
branch 0->2->2->2{
        deep: 3
        num of samples for each class: 15 : 33
        split by dim 0
        branch 0->2->2->2->0{
                deep: 4
                num of samples for each class: 13 : 11
                split by dim 0
                branch 0->2->2->2->0->0{
                        deep: 5
                        num of samples for each class: 4 : 2
                        split by dim 0
                        branch 0->2->2->2->0->0->0{
                                deep: 6
                                num of samples for each class: 2
                                class: 0
                        }
                        branch 0->2->2->2->0->0->1{
                                deep: 6
                                num of samples for each class: 1 : 1
                                class: 0
                        }
                        branch 0->2->2->2->0->0->2{
                                deep: 6
                                num of samples for each class: 1 : 1
                                class: 0
                        }
                }
                branch 0->2->2->2->0->1{
```

```
        deep: 5
        num of samples for each class: 3 : 3
        split by dim 0
        branch 0->2->2->2->0->1->0{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
        branch 0->2->2->2->0->1->1{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
        branch 0->2->2->2->0->1->2{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
}
branch 0->2->2->2->0->2{
        deep: 5
        num of samples for each class: 3 : 3
        split by dim 0
        branch 0->2->2->2->0->2->0{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
        branch 0->2->2->2->0->2->1{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
        branch 0->2->2->2->0->2->2{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
}
branch 0->2->2->2->0->3{
        deep: 5
        num of samples for each class: 3 : 3
        split by dim 0
        branch 0->2->2->2->0->3->0{
                deep: 6
                num of samples for each class: 1 : 1
                class: 0
        }
```

```
                                                branch 0->2->2->2->0->3->1{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                                branch 0->2->2->2->0->3->2{
                                                        deep: 6
                                                        num of samples for each class: 1 : 1
                                                        class: 0
                                                }
                                        }
                                }
                                branch 0->2->2->2->1{
                                        deep: 4
                                        num of samples for each class: 1 : 11
                                        split by dim 0
                                        branch 0->2->2->2->1->0{
                                                deep: 5
                                                num of samples for each class: 1 : 2
                                                split by dim 0
                                                branch 0->2->2->2->1->0->0{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 0
                                                }
                                                branch 0->2->2->2->1->0->1{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 1
                                                }
                                                branch 0->2->2->2->1->0->2{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 1
                                                }
                                        }
                                        branch 0->2->2->2->1->1{
                                                deep: 5
                                                num of samples for each class: 3
                                                class: 1
                                        }
                                        branch 0->2->2->2->1->2{
                                                deep: 5
                                                num of samples for each class: 3
                                                class: 1
                                        }
                                        branch 0->2->2->2->1->3{
                                                deep: 5
                                                num of samples for each class: 3
                                                class: 1
                                        }
                                }
                                branch 0->2->2->2->2{
                                        deep: 4
                                        num of samples for each class: 1 : 11
                                        split by dim 0
                                        branch 0->2->2->2->2->0{
                                                deep: 5
                                                num of samples for each class: 1 : 2
                                                split by dim 0
                                                branch 0->2->2->2->2->0->0{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 0
                                                }
                                                branch 0->2->2->2->2->0->1{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 1
                                                }
                                                branch 0->2->2->2->2->0->2{
                                                        deep: 6
                                                        num of samples for each class: 1
                                                        class: 1
                                                }
                                        }
                                        branch 0->2->2->2->2->1{
                                                deep: 5
                                                num of samples for each class: 3
                                                class: 1
                                        }
                                        branch 0->2->2->2->2->2{
                                                deep: 5
                                                num of samples for each class: 3
                                                class: 1
                                        }
                                        branch 0->2->2->2->2->3{
                                                deep: 5
                                                num of samples for each class: 3
                                                class: 1
                                        }
                                }
                        }
                }
        }
}
```

2.2.3 2.3.3

- do prediction on test dataset.

```
In [11]: import json
         # testing
         y_est_test = dTree.predict(X_test)
         test_accu = accuracy_score(y_est_test, y_test)
         print('test_accu', test_accu)
```

```
test_accu 0.6357267950963222
```

2.3 Part 2.4 Pruning The Tree

Sometimes, in order to prevent overfitting. We need to pruning our Decition Tree. There are several approaches to avoiding overfitting in building decision trees.

- Pre-pruning that stop growing the tree earlier, before it perfectly classifies the training set.
- Post-pruning that allows the tree to perfectly classify the training set, and then post prune the tree.

Practically, the second approach of post-pruning overfit trees is more successful because it is not easy to precisely estimate when to stop growing the tree. We will use Reduced Error Pruning, as one of Post-pruning in this part.

```
Reduced Error Pruning
0. Split data into training and validation sets.
1. Do until further pruning is harmful:
2. Evaluate impact on validation set of pruning each possible node (plus those below it)
3. Greedily remove the one that most improves validation set accuracy
- Produces smallest version of most accurate subtree.
- Requires that a lot of data be available.
```

For Pruning of Decision Tree, you can refer Reduce Error Pruning and P69 of Textbook: Machine Learning -Tom Mitchell.

2.3.1 2.4.1

Hint: in this part, you can add another parameters or functions in TreeNode class and DecisionTree class for your convenience. But your changes should not influent results of previous parts. implement the reduced_error_pruning function on util.py.

```
def reduced_error_pruning(decitionTree):
# input:
    - decitionTree: decitionTree trained based on training data set.
    - X_test: List[List[any]] test data, num_cases*num_attributes
    - y_test: List[any] test labels, num_cases*1
```

Note: To prune from a node, simply set its children to an empty list.

```
In [13]: Utils.reduced_error_prunning(dTree, X_test, y_test)
```

2.3.2 2.4.2

Test your prediction accuracy on validation dataset after pruning.

```
In [14]: y_est_test = dTree.predict(X_test)
         test_accu = accuracy_score(y_est_test, y_test)
         print('test_accu', test_accu)
```

```
test_accu 0.7950963222416813
```

2.3.3 2.4.3

Print your decision tree after pruning.

```
In [15]: Utils.print_tree(dTree)
```

```
branch 0{
        deep: 0
        num of samples for each class: 845 : 260 : 2
        split by dim 5
        branch 0->0{
                deep: 1
                num of samples for each class: 369
                class: 0
        }
        branch 0->1{
                deep: 1
                num of samples for each class: 273 : 96
                split by dim 3
                branch 0->1->0{
                        deep: 2
                        num of samples for each class: 123
                        class: 0
                }
                branch 0->1->1{
                        deep: 2
                        num of samples for each class: 78 : 45
                        class: 1
                }
                branch 0->1->2{
                        deep: 2
                        num of samples for each class: 72 : 51
                        class: 1
                }
        }
        branch 0->2{
                deep: 1
                num of samples for each class: 203 : 164 : 2
                split by dim 3
                branch 0->2->0{
                        deep: 2
                        num of samples for each class: 123
                        class: 0
                }
                branch 0->2->1{
                        deep: 2
                        num of samples for each class: 36 : 86 : 1
                        class: 3
                }
                branch 0->2->2{
                        deep: 2
                        num of samples for each class: 44 : 78 : 1
                        class: 3
                }
        }
}
```

2.4 Grading Guidline

1. Information_Gain function - 10 points we will test your Infomation Gain function on another ten inputs. To receive full credits of this part, your function should be able to output right valus.
2. Train your decision tree - 15 points we will test your decision tree after training on training dataset. To receive full credit of this part, your algorithm will generate the identical decision tree as our answer.
3. Prediction of decision tree - 10 points we will use another dataset to test your prediction part of decision tree, you can assume that test dataset has identical attributs and values as traning dataset. To receive full credit of this part, your algorithm will generate the identical prediction of our answer.
4. Pruning of decision tree - 15 points we will test your decision tree after pruning. To receive full credit of this part, your algorithm will generate the identical decision tree as our answer.

Good Luck! :)