Com S 573 Spring Semester 2021
Machine Learning

Homework 3

Instructor: Hongyang Gao Submitted by: Ibne Farabi Shihab

1 No. Question:

In the given training set examples, there are four features and they are Outlook,Temperature,Humidity and wind. "Play Tennis" is the class label.
To split the the data, we need to calculate Information Gain (IG) from entropy of the features. The formula is(from slide) :
IG = entropy Before (S) - entropy After (A)
Calculating entropy before :
$-\frac{9}{14}log\frac{9}{14}$-$\frac{5}{14}log\frac{5}{14} = 0.940$
Calculating entropy after :
For feature 1 (Outlook):

Outlook	positive	negative	Entropy	Weights
Sunny	2	3	0.97	5/14
rain	3	2	0.97	5/14
Overcast	4	0	0	4/14

Entropy(Outlook)=$0.97 * \frac{5}{14}+ 0.97 * \frac{5}{14} = 0.69$
For feature 2 (Temperature):

Temp	positive	negative	Entropy	Weights
Hot	2	2	1	4/14
Mild	4	2	0.918	6/14
Cold	3	1	0.811	4/14

Entropy(Temp)=$1 * \frac{4}{14}+0.92 * \frac{6}{14}+ 0.97 * \frac{4}{14} = 0.91$
For feature 3 (Humidity):

Humidity	positive	negative	Entropy	Weights
High	3	4	0.985	7/14
Normal	6	1	0.591	7/14

Entropy(Humidity)=$0.985 * \frac{7}{14}+0.59 * \frac{7}{14} = 0.79$
For feature 4 (Wind):

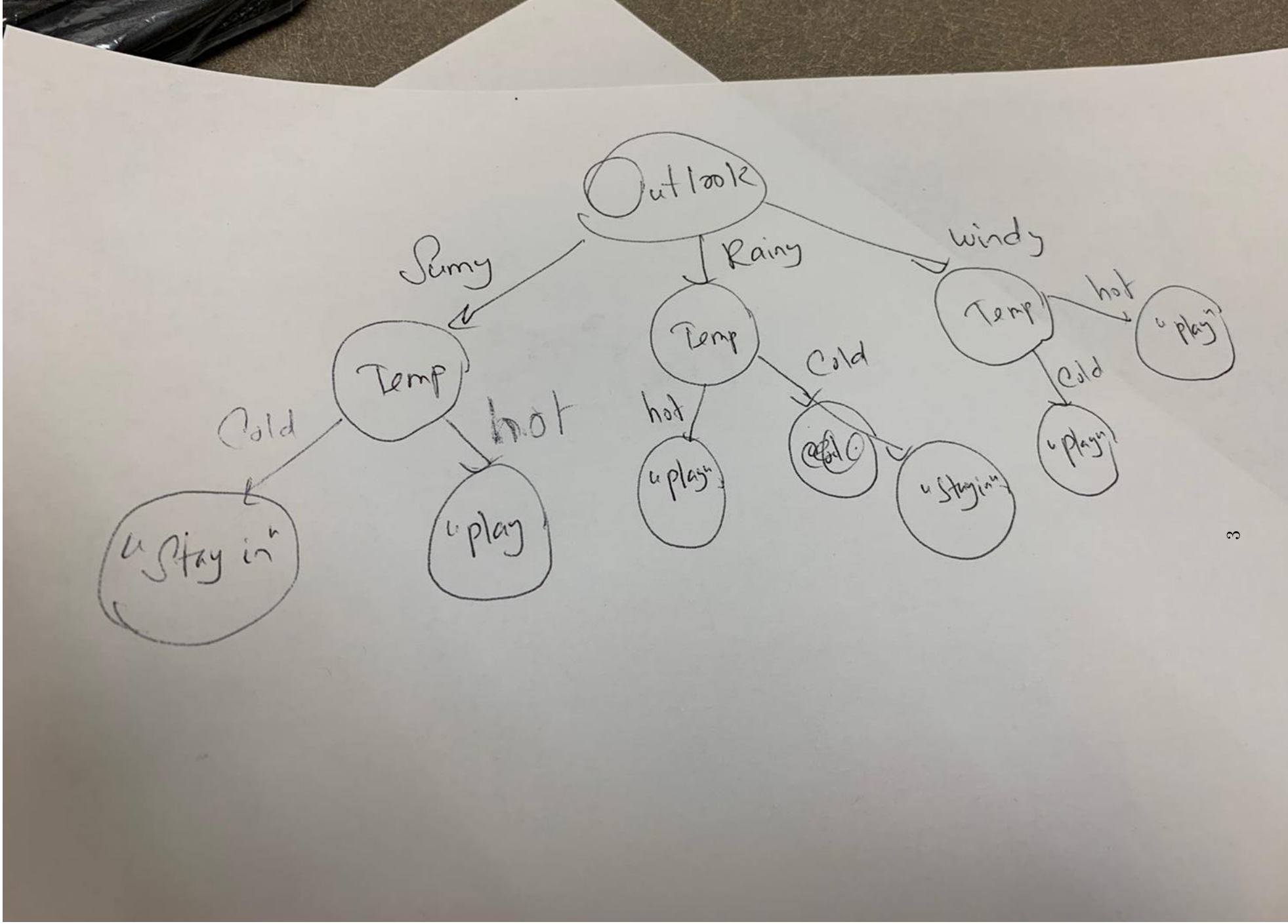
Wind	positive	negative	Entropy	Weights
Strong	3	3	1	6/14
Weak	6	2	0.81	8/14

Entropy(Humidity)=$1 * \frac{7}{14}+0.81 * \frac{8}{14} = 0.89$
Step 4: Calculating IG for all features:
IG (Outlook)= $0.940 - 0.693 = 0.247$
IG (Temperature)= $0.940 - 0.911 = 0.029$
IG (Humidity)= $0.940 - 0.79 = 0.15$
IG (Wind)= $0.940 - 0.89 = 0.048$
So, the highest value for Gain is Outlook 0.247 , for which the root node will be Outlook.

2 No Question:

It is possible to convert the rule set R into an equivalent decision tree. A example will be as follow : Lets say we have 2 attributes with some values like below:

1) Outlook : "Sunny", "Rain", "Windy"
2) Temperature: "Hot", "Cool"
Say the the decision example are as follow:
"Stay-in" or "Play"
Let the rules given be as follows:
1.Outlook = "Sunny", Temperature = "Cold" and Decision = "Stay in"
2.Outlook = "Sunny", Temperature = "Hot" and Decision = "Play"
3.Outlook = "Windy",Temperature = "Hot" and Decision = "Play"
4.Outlook = "Rainy", Temperature = "Cold" and Decision = "Stay In"
5.Outlook = "Rainy", Temperature = "Hot" and Decision = "Play"
The equivalent decision tree can be formed as follows:


Outlook
Sunny
Rainy
Windy
Temp
Temp
Temp
Cold
hot
hot
Cold
Cold
hot
"Stay in"
"play"
"play"
"Stay in"
"play"
"play"

3 No Question (a)

Here we can simply the given as follows:

$$\left(\frac{1}{\sqrt{d}}\sum_{i=1}^{d}x_i - \frac{1}{\sqrt{d}}\sum_{i=1}^{d}z_i\right)^2 = \frac{1}{d}\left(\sum_{i=1}^{d}x_i - \sum_{i=1}^{d}z_i\right)^2$$

$$= \frac{1}{d}\left(\sum_{i=1}^{d}(x_i - z_i)\right)^2$$

Now, $(X_i - Z_i)$ is convex for any value $i = 1, 2, ...d$. We know that the sum of any convex function is also convex.. It means $(\sum_{i=1}^{d}(x_i - z_i)$ is also convex.
Now, using Jensen's inequality $f(E[X]) \leq E[f(X)]$.

So,

$$= \left(\sum_{i=1}^{d}(x_i - z_i)\right)^2 \leq \sum_{i=1}^{d}(x_i - z_i)^2$$

As it is said that d is very large number. So we can add d here as it is very insignificant.

$$\frac{1}{d}\left(\sum_{i=1}^{d}(x_i - z_i)\right)^2 \leq \sum_{i=1}^{d}(x_i - z_i)^2$$

or,

$$\left(\frac{1}{\sqrt{d}}\sum_{i=1}^{d}x_i - \frac{1}{\sqrt{d}}\sum_{i=1}^{d}z_i\right)^2 \leq \sum_{i=1}^{d}(x_i - z_i)^2$$

3 NO Question (b)

We know that the squared distance between one-dimensional projections of points x and z is always less than or equals to the distance between them in d-dimensional space.
In case of the calculation of nearest neighbour in d-dimension, first complex distance computations have to be made in d-dimensional for each vector which are later sorted to find the nearest neighbour from them.
However, if use this inequality, we can see that a we can get a similar sorting order by calculating mean of feature values to project each point on to a real number line by doing a simple addition of d points and then a division by a scalar. After that we can sort those values, which is a less complex operation with compare to the former one. In this way, the computation of nearest neighbour will be less complex and easy to calculate.

4 No Question (c)

	DT (without pruning)	DT (with pruning)
Validation acc	0.57	0.58
Testing acc	0.67	0.55
Tree Depth	182	60

We started our with a general decision tree and later pruned it. We decided on pruning using the validation accuracy with compare to majority class accuracy. If the validation case accuracy is higher for majority class than the general tree validation accuracy we go for pruning. Below are the observation from the results:

1. In case of validation accuracy we see an improvement for the tree with pruning. In this case, we can say that, pruning has improved our accuracy in validation set.

2. In terms of testing accuracy, we see a significant improvement in our testing accuracy in decision tree without pruning. However the accuracy on testing set is pretty high for DT without pruning which is unlikely and not an normal case. One reason for this might be that the data are not drawn for the same distribution. Another can be for the random nature of tree structure in general. On the contrary, after doing the pruning the high accuracy disappeared which justify that there are not anything with the code rather something to do with the data and the random nature of tree. In this case, we can say that pruning did help us to get rid of the weirdness.

3. Pruning also helped to get almost closer accuracy without going to deep. Similarity between testing accuracy and validation accuracy of signifies that.

Acknowledgement: For the coding part and question 3 few help has been taken from various online resources. Question has been somewhat a replica of the example in question 1.