Problem 1:

a)

There is one unique $\boldsymbol{H_D}$ exists.

Assume that $\boldsymbol{X}$, $\boldsymbol{L}$ and $\boldsymbol{D}$ are all finite and given, the one unique $\boldsymbol{H_D}$ exists. As we know the set of unique examples $\boldsymbol{X}$, called the *instance space*, $\boldsymbol{X}$ is consisting of a lot of attributes. $\boldsymbol{D}$ contains examples from $\boldsymbol{X}$ with known labels. When giving the $\boldsymbol{D}$, the number of the sample is certain. If we want to build a set that contains all the pairs of hypotheses that are distinguishable given $\boldsymbol{D}$, we need to get a lot of hypotheses, also the number of hypotheses is certain as well, so does its result set. So the $\boldsymbol{H_D}$ is unique.

Giving an example that $\boldsymbol{D}$ contains 2 factors{1, 2,}, each label is {0,1}, there must be 8 distinguishable hypotheses so that it can consider all the possibilities. It can be showed as follow:

	1	2
h(1)	L=0	L=0
h(2)	L=0	L=1
h(3)	L=1	L=0
h(4)	L=1	L=1

The number of hypotheses may change sometimes, but when we see it as a whole set, each of h(x) can contribute to the unique set, which is $\boldsymbol{H_D}$

b)

Let the size of the data and the label set be drawn from the counting numbers (i.e. | *D* |,| *L* |∈{1,2,3...}). The $\boldsymbol{H_D}$ can be expressed as:

$$\boldsymbol{H_D = L^D}$$

As showed in the example above, when $\boldsymbol{D}$ is certain, the number of giving samples is certain, so considering the possible result of $\boldsymbol{L}$ that each sample could possibly get, we can find the expression above.

c)

When giving the label set $\boldsymbol{L}$ is {0,1}. The size of the data set |$\boldsymbol{D}$| = 100. The size of the instance space |X| = 200. Assume a learner able to consider a maximal set of distinguishable hypotheses $\boldsymbol{H_D}$. As we conclude in the problem a and b, we know the total number of $\boldsymbol{H_D}$ can be

$$\boldsymbol{H_D = 2^{100}}$$

Assuming the learner is able to consider 10^9 (one billion) hypotheses per second. The worst case happens when the last hypothesis ***h*** is the one that being indistinguishable from the target function ***c***, given ***D***. So the total time can be expressed as:

$$\mathrm{t}=\frac{\boldsymbol{H}_D}{\mathrm{v}}=\frac{2^{100}}{10^9}\approx 10^{21}s$$

As we see, the time is too long for a learner, which means it is not a reasonable waiting time at all.

d)

The probability of hypothesis ***h*** is also indistinguishable from the target concept ***c*** , given ***X*** is $1/2\hat{}100$

We consider this question into the Bayes formula, assuming:

A= {Hypothesis ***h*** is indistinguishable from ***c*** given ***X***}

B= {Hypothesis ***h*** is indistinguishable from ***c*** given ***D***}

B|A= {Hypothesis ***h*** is indistinguishable from ***c*** given ***D***, when giving the condition that Hypothesis ***h*** is indistinguishable from ***c*** given ***X***}

A|B= {Hypothesis ***h*** is indistinguishable from ***c*** given ***X***, when giving the condition that Hypothesis ***h*** is indistinguishable from ***c*** given ***D***}

We know that:

$\mathrm{P(A)} = 1/2\hat{}(200-100) = 1/2\hat{}100$

$\mathrm{P(B)} = 1$ （giving the condition that ***h*** is applicable for ***D***）

$\mathrm{P(B|A)} = 1$ (If ***h*** is applied to ***X,*** it surely can be used in ***D***)

So based on the Bayes formula:

$$\mathrm{P(A|B)} = \mathrm{P(B|A)}\cdot\frac{\mathrm{P(A)}}{\mathrm{P(B)}}=\frac{1}{2^{100}}$$

Problem 2:

The giving data is as follow:

Origin	**Manufacturer**	**Color**	**Decade**	**Type**	**Classification**
Japan	Honda	Blue	1980	Economy	**1**
Japan	Toyota	Green	1970	Sports	**0**
Japan	Toyota	Blue	1990	Economy	**1**
USA	Chrysler	Red	1980	Economy	**0**
Japan	Honda	White	1980	Economy	**1**

a)

To illustrate, suppose S is a collection of 5 examples of Boolean concept, including 3 positive (whose classification is 1) and 2 negative (whose classification is 0) examples. Then the entropy of S can be:

$$\text{Entropy}(S) = \sum_{i=1}^{c} -p_i log_2 p_i$$

$$\text{Entrophy}[3+,2-] = -\frac{2}{5} log_2\left(\frac{2}{5}\right) - \frac{3}{5} log_2\left(\frac{3}{5}\right) = 0.971$$

(1) Considering the first attribute origin:

Values(origin) = Japan, USA

$S_{Japan} = [3+, 1-]$

$S_{USA} = [0+, 1-]$

$$\text{Gain}(\text{S}, \text{origin}) \quad = \text{Entropy}(\text{S}) - \sum_{v \in \{Japan, USA\}} \frac{|S_V|}{|S|} \text{Entrophy}(S_V)$$

$$= 0.971 - \frac{4}{5} \times 0.811 = 0.322$$

(2) Considering the second attribute Manufacturer:

Values(Manufacturer) = Toyota, Chrysler, Honda

$S_{Toyota} = [1+, 1-]$
$S_{Chrysler} = [0+, 1-]$
$S_{Honda} = [2+, 0-]$

$$\text{Gain(S, manufacturer)} \quad = \text{Entrophy(S)} - \sum_{v \in \{T,C,H\}} \frac{|S_V|}{|S|} \text{Entrophy}(S_V)$$

$$= 0.971 - 0.4 = 0.571$$

(3) Considering the third attribute Color:
Values(Manufacturer) = blue, red, white, green
$S_{blue} = [2+, 0-]$
$S_{red} = [0+, 1-]$
$S_{white} = [1+, 0-]$
$S_{green} = [0+, 1-]$

$$\text{Gain(S, colour)} \quad = \text{Entrophy(S)} - \sum_{v \in \{b,r,w,g\}} \frac{|S_V|}{|S|} \text{Entrophy}(S_V)$$

$$= 0.971 - 0 = 0.971$$

(4) Considering the fourth attribute Decade:
Values(Decade) = 1970,1980,1990
$S_{1970} = [0+, 1-]$
$S_{1980} = [2+, 1-]$
$S_{1990} = [1+, 0-]$

$$\text{Gain(S, decade)} \quad = \text{Entrophy(S)} - \sum_{v \in \{70,80,90\}} \frac{|S_V|}{|S|} \text{Entrophy}(S_V)$$

$$= 0.971 - \frac{3}{5} \times 0.918 = 0.420$$

(5) Considering the fifth attribute Decade:
Values(Type) = Economy, Sports
$S_{Economy} = [3+, 1-]$
$S_{Sports} = [0+, 1-]$

$$\text{Gain(S, decade)} \quad = \text{Entrophy(S)} - \sum_{v \in \{E,S\}} \frac{|S_V|}{|S|} \text{Entrophy}(S_V)$$

$$= 0.971 - \frac{4}{5} \times 0.811 = 0.322$$

Compared those four Information gains, we can find the largest value come from the term of “color”.

b)

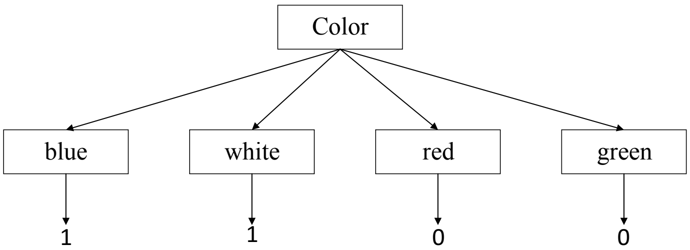
The logic function can be expressed by the tree above.

$\text{If Color} = \text{blue} \cup \text{Color} = \text{white}$

$\text{Then classification} = 1$

$\text{If Color} = \text{red} \cup \text{Color} = \text{green}$

$\text{Then classification} = 0$



(1) This function does not use the data very well, in fact, it just finds the connection between color and the classification. However, based on our background knowledge, color should not be a determined factor.

(2) Also, it does badly to explain the concept of "Japan Economy car", this decision tree has nothing to do with either Japan or Economy.

(3) The limitation of the giving samples causes this severe problem when running ID3, 5 cars is not enough for a sample to explain the concept of "Japan Economy car", instead, it accidently finds the connection between classification and color, which should have not existed.

Problem 3

a)

I have two thoughts to pick the split point:

(1) The first method is to put the split point in the middle of the value area. For example, the value area of attribute ***a*** is [0, 6], then I the split point should be 3.

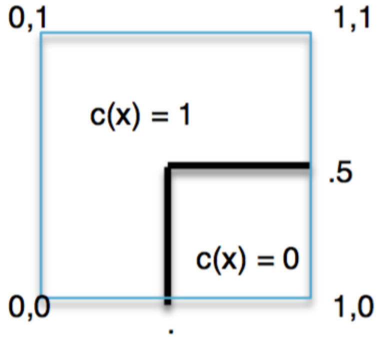
This is directly based on the rule of dichotomy in the term of value area, the performance can be great when the value distributed more averagely through the giving area.

(2) The second method is to put the split point in the estimated median of all giving examples. The first step is to give a estimated median of all attribute ***a*** of giving examples, then put the split point into the estimated median value.

Comparing 2 methods above, the first one is easier to achieve, it performs well when the value of examples are distributed averagely, however, it can be useless when the value of examples overdisperse, for example, giving the value area [0, 6], 90% of attribute ***a*** in the ***X*** is less than 3, only 10% is greater than 3. In this case, the split point cannot perform well.

On the contrary, the second method can be more accurate when the giving sample ***D*** is representative for the examples ***X*** and also the number of samples is sufficient. In this case can it solve the binary problem.

b)



I think the second concept can be represented.

For the 2-dimensional real-valued points, it can be expressed as in coordinate (x,y). To divide the feature space, there shall be 2 split points, one for value of x, another for value of y. In this circumstance, the 2-dimensional area can be divide into axis-parallel rectangle as the second method presents.

However, for the first concept, it's criterial equation is y=x, it is not a feasible way because two variable x and y are presented in the same equation. It is not the way that decision tree uses.

Problem 4:

a)

The term of True positive, True negative, False positive, False negative are used to measure the performance of a binary classification test.

(1) True positive: Cases that positives that are correctly identified as such (e.g., Cases that sick people who are correctly identified as having the condition).

(2) True negative: Cases that negatives that are correctly identified as such (e.g., Cases that healthy people who are correctly identified as not having the condition).

(3) False positive: Cases that positives that are incorrectly identified as such (e.g., Cases that unhealthy people who are incorrectly identified as not having the condition).

(4) False negative: Cases that negatives that are incorrectly identified as such (e.g., Cases that healthy people who are correctly identified as having the condition).

Cited from: https://en.wikipedia.org/wiki/Sensitivity_and_specificity

b)

In pattern recognition and information retrieval with binary classification, precision (also called positive predictive value) is the fraction of retrieved instances that are relevant, while recall (also known as sensitivity) is the fraction of relevant instances that are retrieved. Both precision and recall are therefore based on an understanding and measure of relevance.

(1) Precision: In the field of information retrieval, precision is the fraction of retrieved documents that are relevant to the query

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retreieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

(2) Recall: Recall in information retrieval is the fraction of the documents that are relevant to the query that are successfully retrieved.

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retreieved documents}\}|}{|\{\text{relevant documents}\}|}$$

Cited from https://en.wikipedia.org/wiki/Precision_and_recall

c)

F1 score:

In statistical analysis of binary classification, the F1 score (also F-score or F-measure) is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results, and r is the number of correct positive results divided by the number of positive results that should have been returned. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst at 0.

The traditional F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Cited from : https://en.wikipedia.org/wiki/F1_score

d)

(1) Confusion matrix:

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix,[4] is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class (or vice-versa). The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabelling one as another).

Cited from "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation"

(2) Reason in being useful for analysis of machine learning:

The confusion matrix is useful to machine learning because it can compare the predicate values with actual values, by this means can people find if the original hypotheses successfully achieve. It’s easy to visually inspect the table for errors, as they will be represented by values outside the diagonal.

(3) Example when a classifier has more than 2 classes:

To consider a classifier that has more than 2 classes, we can bring 3 kinds of fruits: Apples, Bananas, Peach. Assuming the total number of them is 20 (8 apples, 5 bananas, 7 peaches).

A system is supposed to distinguish among those 3 kinds. A confusion matrix can summarize the performance of the system, as follow:

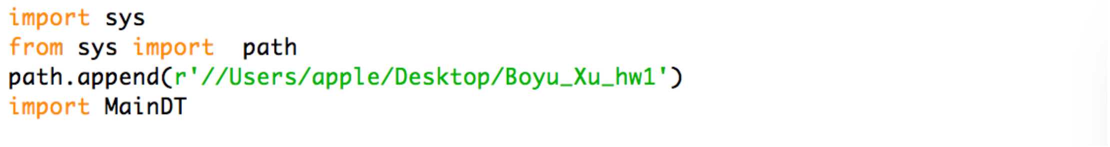
		Predicted		
Actual Class	Class	Apple	Banana	Peach
	Apple	4	0	2
	Banana	0	4	1
	Peach	2	0	5

In this confusion matrix, of the 8 actual apples, the system predicted that 1 belongs to peach. And for the 5 bananas, it predicted that 1 was peach. As for peach, 2 of them are categorized as apple. We can see from the matrix that the system in question has trouble distinguishing between apples and peaches, comparably, this system can make the distinction between bananas and other types of fruits pretty well. As it appears, all correct guesses are located in the diagonal of the table, so it's easy to visually inspect the table for errors, as they will be represented by values outside the diagonal. It's help for a system when it contains more than 2 classes, it's can greatly find the wrongdoing classes among all elements.

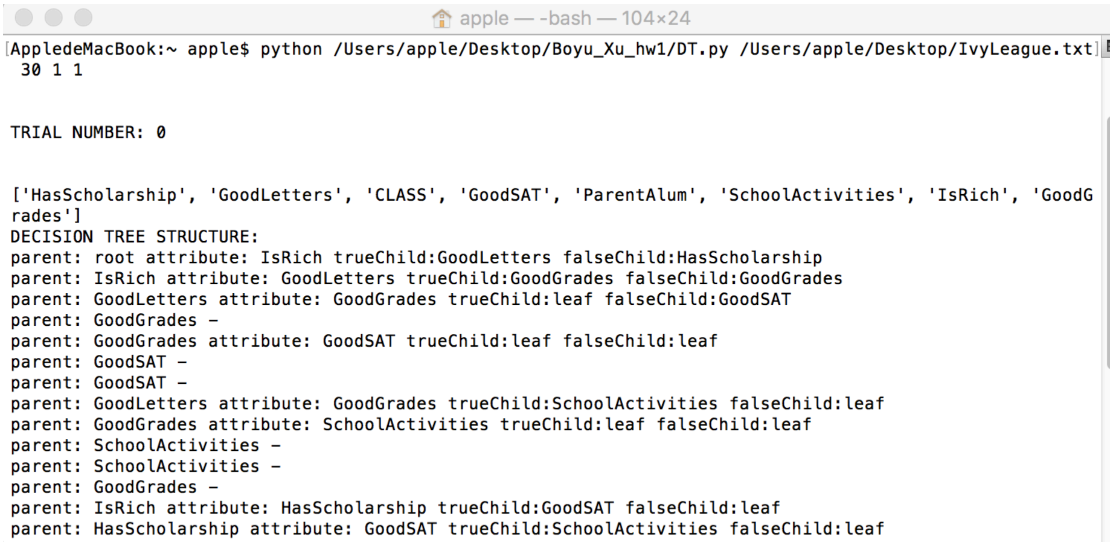
Problem 5:

NOTE:

As attached 2 .py files. Also, to test my program, here needs a little bit change in the *input path for the "DT.py"*



```
import sys
from sys import  path
path.append(r'//Users/apple/Desktop/Boyu_Xu_hw1')
import MainDT
```



```
AppledeMacBook:~ apple$ python /Users/apple/Desktop/Boyu_Xu_hw1/DT.py /Users/apple/Desktop/IvyLeague.txt
 30 1 1


TRIAL NUMBER: 0


['HasScholarship', 'GoodLetters', 'CLASS', 'GoodSAT', 'ParentAlum', 'SchoolActivities', 'IsRich', 'GoodG
rades']
DECISION TREE STRUCTURE:
parent: root attribute: IsRich trueChild:GoodLetters falseChild:HasScholarship
parent: IsRich attribute: GoodLetters trueChild:GoodGrades falseChild:GoodGrades
parent: GoodLetters attribute: GoodGrades trueChild:leaf falseChild:GoodSAT
parent: GoodGrades -
parent: GoodGrades attribute: GoodSAT trueChild:leaf falseChild:leaf
parent: GoodSAT -
parent: GoodSAT -
parent: GoodLetters attribute: GoodGrades trueChild:SchoolActivities falseChild:leaf
parent: GoodGrades attribute: SchoolActivities trueChild:leaf falseChild:leaf
parent: SchoolActivities -
parent: SchoolActivities -
parent: GoodGrades -
parent: IsRich attribute: HasScholarship trueChild:GoodSAT falseChild:leaf
parent: HasScholarship attribute: GoodSAT trueChild:SchoolActivities falseChild:leaf
```