CS446: Machine Learning **Fall 2015**

Problem Set 2

Jifu Zhao *Handed In: 09/29/2015*

1. **Answer to problem 1**

(a) Determine the root attribute

To find the root attribute, we need to calculate the information gain for Holiday and Exam Tomorrow separately.

For **Holiday attribute**,

$p = 35/50$ and $n = 15/50$. So,entropy is: $H(Holoday) = -(35/50)\log_2(35/50) - (15/50)\log_2(15/50) = 0.88129$.

When $Holiday = yes$, the fraction is 15/50, and $p = 5/15$, $n = 10/15$. So,entropy is: $H(Holoday = yes) = -(5/15)\log_2(5/15) - (10/15)\log_2(10/15) = 0.91830$.

When $Holiday = no$, the fraction is 35/50, and $p = 30/35$, $n = 5/35$. So,entropy is: $H(Holoday = no) = -(30/35)\log_2(30/35) - (5/35)\log_2(5/35) = 0.59167$.

So, the **Information Gain** for Holiday attribute is:

$$Gain(Holiday) = H(Holoday) - (15/50)H(yes) - (35/50)H(no) = 0.19163$$

For **Exam Tomorrow attribute**,

$p = 35/50$ and $n = 15/50$. So,entropy is: $H(Exam) = -(35/50)\log_2(35/50) - (15/50)\log_2(15/50) = 0.88129$.

When $Exam = yes$, the fraction is 16/50, and $p = 15/16$, $n = 1/16$. So,entropy is: $H(Exam = yes) = -(15/16)\log_2(15/16) - (1/16)\log_2(1/16) = 0.33729$.

When $Exam = no$, the fraction is 34/50, and $p = 20/34$, $n = 14/34$. So,entropy is: $H(Exam = no) = -(20/34)\log_2(20/34) - (14/34)\log_2(14/34) = 0.97742$.

So, the **Information Gain** for Exam Tomorrow attribute is:

$$Gain(Exam) = H(Exam) - (16/50)H(yes) - (34/50)H(no) = 0.10871$$

Comparing $Gain(Holiday)$ and $Gain(Exam)$, it's easy to conclude that we should choose **Holiday** as the root attribute.

(b) Build the decision tree

1. Calculate the root attribute

Since we choose $MajorityError = min(p, 1 - p)$ as the the measure of entropy (H), then the information gain (Gain) will be:

$$Gain = H(S) - \sum_k \frac{|S_k|}{|S|} H(S_k)$$

According to this rule, let's first calculate the root attribute from Color, Size, Act and Age.

$$Gain(Color) = 0.125$$
$$Gain(Size) = 0.125$$
$$Gain(Act) = 0.125$$
$$Gain(Age) = 0.125$$

Since these 4 attributes have the same information gain, we can choose one of them randomly. Now, let's choose **Color** as the **root attribute**.

2. When Color = Yellow

When Color is Yellow, we can get that:

$$Gain(Size) = 0.25$$
$$Gain(Act) = 0.0$$
$$Gain(Age) = 0.0$$

So, we choose Size.

When Size is small, the **Inflated** value will always be **T**.

When Size is large, we can get:

$$Gain(Act) = 0.0$$
$$Gain(Age) = 0.0$$

So, we choose **Act**. When Act is Dip, all Inflated will be F. When Act is Strecth, we will choose Age. When Age is Adult, Inflated is F and when Age is Child, Inflated is T.

3. When Color = Purple

Similarly, we first get that:

$$Gain(Size) = 0.0$$
$$Gain(Act) = 0.0$$
$$Gain(Age) = 0.0$$

So we choose Size. Then, when Size is Small, we can get that:

$$Gain(Act) = 0.0$$
$$Gain(Age) = 0.0$$

So, we choose Act. When Act is Dip, all will be F. When Act is Stretch, when Age is Adult, we get T. When Age is Child, we get F.

When Size is Large, we can get that:

$$Gain(Act) = 0.0$$
$$Gain(Age) = 0.0$$

So, we choose Act. When Act is Dip, all will be F. When Act is Strecth, when Age is Adult, we get T. When Age is Child, we get F.

4. The final Decision Tree

```
if Color = Yellow:
    if Size = Small:
        Class = T
    if Size != Small:
        if Act = Dip:
            Class = F
        if Act != Dip:
            if Age = Adult:
                Class = T
            if Age != Adult:
                Class = F
if Color != Yellow:
    if Size = Small:
        if Act = Dip:
            Class = F
        if Act != Dip:
            if Age = Adult:
                Class = T
            if Age != Adult:
                Class = F
    if Size != Small:
        if Act = Dip:
            Class = F
        if Act != Dip:
            if Age = Adult:
```

```
        Class = T
    if Age != Adult:
        Class = F
```

(c) Optimal decision tree analysis

Generally, ID3 algorithm is a forward search process. This means that it will search and build the decision along one branch of the tree. So, it will not have the process of backtracking. In other words, once it choose one attribute, it will not go back and reconsider its choice again. So, the final result will be the locally optimal choice along the single path. However, this locally optimal choice doesn't means the globally optimal choice. In order to have a globally optimal decision tree, we need to go back and compare our choice not only in the single path, but also with other path.

However, in this question, under this special cases, there is no optimal decision trees. All decisions have the same effect. But this is only for this special case.

2. **Answer to problem 2**

(a) Feature Extraction and Instance Generation

In this part, through change the jave file named FeatureGenerator.java, choose the first 5 character of firstName and lastName, the features are generated correctly. One thing to be mentioned is that, some names have less than 5 characters. In this case, I think that all feature will be 0. (Another method is to think them as Special Characters, such as "None", there will be more descriptions below)

(b) Build the decision tree

Case A

In this part, 5 algorithms are evaluated. SGD, Decision tree, Decision stumps of depth 4, Decision stumps of depth 8, and Decision Stumps as features. The result are shown below:

Table 1: Cross Validation Result

Algorithm	**CV Accuracy(%)**	**99% Interval(%)**	**Parameters**
Full Decision tree	68.3	[54.62, 82.07]	
SGD	66.31	[48.39, 84.23]	Rate=0.005, Threshold=0.001
SGD over 100 stumps	63.24	[44.92, 81.57]	Rate=0.005, Threshold=0.001
Decision Stump of 8	62.92	[40.04, 85.79]	tree depth = 8
Decision Stump of 4	59.86	[33.47, 86.26]	tree depth = 4

In this table, I choose the five characters from first name and last name. When there is less than 5 characters in first or last name, the corresponding feature value will be all 0. In this way, through calculation, I can get the following result.

After trying sometimes, I finally choose **Learning Rate $\alpha = 0.005$** and **Threshold = 0.001**. Of course we can choose smaller learning rate and smaller threshold, but that will strongly increase the calculation time. And the final result will not change too much.

Now, calculate the difference for each pair of consecutive algorithms. This procedure is done through calculate the sample standard deviation and sample mean. The results are shown in the table 2:

Table 2: Significance Analysis

SGD VS. Full tree	SGD stumps VS. SGD	Decision 8 VS. SGD strmps	Decision 4 VS. Decision 4
t = 1.399	t = 1.072	t = 0.059	t = 2.25

Having got these result, we can do the significance test now.

There, Hypothesis is: H_0**: The difference between two algorithms is 0**

Through Students' t table, we can find that, when degree of freedom is 4 (in this case), when P = 0.05, we got t = 2.776. When P = 0.01, t = 4.604. When P = 0.001, t = 8.610. So, within the confidence interval of 99.9% and 99%, we cannot reject the original hypothesis. However, within the confidence interval of 90%, we can reject this hypothesis for the Decision tree of depth 4 and Decision tree of depth 8.

Conclusion:

From the above analysis, we can conclude that Full Decision Tree have better performance over the other method. However, even for the Full decision tree, its accuracy is below 70%. This is because that the feature extraction is not very good. It is obvious that if we change the selection of features, we can have much better results.

For the SGD algorithm, since our problem may be not completely linearly separable, and also, due to our selection of learning rate α and threshold, it is possible that our result didn't look very good.

Addition For **Full decision tree**, the best result is that:

```
Correctly Classified Instances    45    76.2712 %
```

```
Incorrectly Classified Instances      14      23.7288 %
```

Decision Tree:

ID3

```
firstName1=a = 1
|  lastName1=o = 1: +
|  lastName1=o = 0
|  |  firstName3=y = 1: -
|  |  firstName3=y = 0
|  |  |  lastName1=n = 1: -
|  |  |  lastName1=n = 0
|  |  |  |  lastName2=m = 1: -
|  |  |  |  lastName2=m = 0
|  |  |  |  |  lastName2=l = 1
|  |  |  |  |  |  firstName2=n = 1
|  |  |  |  |  |  |  firstName3=i = 1: -
|  |  |  |  |  |  |  firstName3=i = 0: +
|  |  |  |  |  |  firstName2=n = 0: -
|  |  |  |  |  lastName2=l = 0
|  |  |  |  |  |  firstName0=m = 1: +
|  |  |  |  |  |  firstName0=m = 0
|  |  |  |  |  |  |  firstName3=k = 1: -
|  |  |  |  |  |  |  firstName3=k = 0
|  |  |  |  |  |  |  |  lastName1=a = 1
|  |  |  |  |  |  |  |  |  firstName0=s = 1: -
|  |  |  |  |  |  |  |  |  firstName0=s = 0
|  |  |  |  |  |  |  |  |  |  firstName0=p = 1: -
|  |  |  |  |  |  |  |  |  |  firstName0=p = 0: +
|  |  |  |  |  |  |  |  lastName1=a = 0
|  |  |  |  |  |  |  |  |  firstName2=m = 1: -
|  |  |  |  |  |  |  |  |  firstName2=m = 0
|  |  |  |  |  |  |  |  |  |  lastName1=k = 1: -
|  |  |  |  |  |  |  |  |  |  lastName1=k = 0: +
firstName1=a = 0
|  firstName4=a = 1
|  |  lastName4=e = 1
|  |  |  firstName0=m = 1: +
|  |  |  firstName0=m = 0
|  |  |  |  firstName1=t = 1: +
|  |  |  |  firstName1=t = 0: -
|  |  lastName4=e = 0
|  |  |  lastName0=z = 1: -
|  |  |  lastName0=z = 0
|   |   |   |   lastName2=z = 1: -
|   |   |   |   lastName2=z = 0: +
|   firstName4=a = 0
|   |   firstName3=a = 1
|   |   |   lastName0=s = 1
|   |   |   |   firstName0=m = 1: +
|   |   |   |   firstName0=m = 0: -
|   |   |   lastName0=s = 0
|   |   |   |   firstName0=b = 1
|   |   |   |   |   lastName0=d = 1: -
|   |   |   |   |   lastName0=d = 0: +
|   |   |   |   firstName0=b = 0: +
|   |   firstName3=a = 0
|   |   |   firstName3=d = 1: +
|   |   |   firstName3=d = 0
|   |   |   |   firstName2=a = 1
|   |   |   |   |   lastName3=t = 1: +
|   |   |   |   |   lastName3=t = 0
|   |   |   |   |   |   lastName0=c = 1: +
|   |   |   |   |   |   lastName0=c = 0
|   |   |   |   |   |   |   lastName0=m = 1: +
|   |   |   |   |   |   |   lastName0=m = 0
|   |   |   |   |   |   |   |   lastName0=k = 1
|   |   |   |   |   |   |   |   |   firstName1=m = 1: -
|   |   |   |   |   |   |   |   |   firstName1=m = 0: +
|   |   |   |   |   |   |   |   lastName0=k = 0
|   |   |   |   |   |   |   |   |   firstName0=f = 1: +
|   |   |   |   |   |   |   |   |   firstName0=f = 0: -
|   |   |   |   firstName2=a = 0
|   |   |   |   |   lastName3=r = 1: +
|   |   |   |   |   lastName3=r = 0
|   |   |   |   |   |   firstName4=n = 1
|   |   |   |   |   |   |   firstName0=g = 1: -
|   |   |   |   |   |   |   firstName0=g = 0: +
|   |   |   |   |   |   firstName4=n = 0
|   |   |   |   |   |   |   firstName3=n = 1
|   |   |   |   |   |   |   |   lastName0=r = 1: +
|   |   |   |   |   |   |   |   lastName0=r = 0
|   |   |   |   |   |   |   |   |   lastName2=a = 1
|   |   |   |   |   |   |   |   |   |   firstName0=r = 1: -
|   |   |   |   |   |   |   |   |   |   firstName0=r = 0: +
|   |   |   |   |   |   |   |   |   lastName2=a = 0: -
|   |   |   |   |   |   |   firstName3=n = 0
|   |   |   |   |   |   |   |   firstName2=d = 1
|   |   |   |   |   |   |   |   |   firstName0=a = 1
|   |   |   |   |   |   |   |   |   |   lastName0=b = 1: +
|   |   |   |   |   |   |   |   |   |   lastName0=b = 0: −
|   |   |   |   |   |   |   |   |   firstName0=a = 0: +
|   |   |   |   |   |   |   |   firstName2=d = 0
|   |   |   |   |   |   |   |   |   firstName2=n = 1
|   |   |   |   |   |   |   |   |   |   firstName1=o = 1
|   |   |   |   |   |   |   |   |   |   |   firstName0=l = 1: −
|   |   |   |   |   |   |   |   |   |   |   firstName0=l = 0: +
|   |   |   |   |   |   |   |   |   |   firstName1=o = 0: −
|   |   |   |   |   |   |   |   |   firstName2=n = 0
|   |   |   |   |   |   |   |   |   |   firstName3=o = 1
|   |   |   |   |   |   |   |   |   |   |   firstName0=a = 1: +
|   |   |   |   |   |   |   |   |   |   |   firstName0=a = 0
|   |   |   |   |   |   |   |   |   |   |   |   firstName2=k = 1: +
|   |   |   |   |   |   |   |   |   |   |   |   firstName2=k = 0: −
|   |   |   |   |   |   |   |   |   |   firstName3=o = 0
|   |   |   |   |   |   |   |   |   |   |   firstName0=n = 1: +
|   |   |   |   |   |   |   |   |   |   |   firstName0=n = 0: −
```

For **Decision tree of depth 8**, the best result is that:
Correctly Classified Instances   44   74.5763 %
Incorrectly Classified Instances   15   25.4237 %

Decision Tree:

ID3

```
firstName1=a = 1
|   lastName1=o = 1: +
|   lastName1=o = 0
|   |   firstName3=y = 1: −
|   |   firstName3=y = 0
|   |   |   lastName1=n = 1: −
|   |   |   lastName1=n = 0
|   |   |   |   lastName2=m = 1: −
|   |   |   |   lastName2=m = 0
|   |   |   |   |   lastName2=l = 1
|   |   |   |   |   |   firstName2=n = 1
|   |   |   |   |   |   |   firstName3=i = 1: −
|   |   |   |   |   |   |   firstName3=i = 0: +
|   |   |   |   |   |   firstName2=n = 0: −
|   |   |   |   |   lastName2=l = 0
|   |   |   |   |   |   firstName0=m = 1: +
|   |   |   |   |   |   firstName0=m = 0
|   |   |   |   |   |   |   firstName3=k = 1: −
|   |   |   |   |   |   |   firstName3=k = 0
|   |   |   |   |   |   |   |   lastName1=a = 1: +
|   |   |   |   |   |   |   |   lastName1=a = 0: +
firstName1=a = 0
|   firstName4=a = 1
|   |   lastName4=e = 1
|   |   |   firstName0=m = 1: +
|   |   |   firstName0=m = 0
|   |   |   |   firstName1=t = 1: +
|   |   |   |   firstName1=t = 0: -
|   |   lastName4=e = 0
|   |   |   lastName0=z = 1: -
|   |   |   lastName0=z = 0
|   |   |   |   lastName2=z = 1: -
|   |   |   |   lastName2=z = 0: +
|   firstName4=a = 0
|   |   firstName3=a = 1
|   |   |   lastName0=s = 1
|   |   |   |   firstName0=m = 1: +
|   |   |   |   firstName0=m = 0: -
|   |   |   lastName0=s = 0
|   |   |   |   firstName0=b = 1
|   |   |   |   |   lastName0=d = 1: -
|   |   |   |   |   lastName0=d = 0: +
|   |   |   |   firstName0=b = 0: +
|   |   firstName3=a = 0
|   |   |   firstName3=d = 1: +
|   |   |   firstName3=d = 0
|   |   |   |   firstName2=a = 1
|   |   |   |   |   lastName3=t = 1: +
|   |   |   |   |   lastName3=t = 0
|   |   |   |   |   |   lastName0=c = 1: +
|   |   |   |   |   |   lastName0=c = 0
|   |   |   |   |   |   |   lastName0=m = 1: +
|   |   |   |   |   |   |   lastName0=m = 0
|   |   |   |   |   |   |   |   lastName0=k = 1: +
|   |   |   |   |   |   |   |   lastName0=k = 0: -
|   |   |   |   firstName2=a = 0
|   |   |   |   |   lastName3=r = 1: +
|   |   |   |   |   lastName3=r = 0
|   |   |   |   |   |   firstName4=n = 1
|   |   |   |   |   |   |   firstName0=g = 1: -
|   |   |   |   |   |   |   firstName0=g = 0: +
|   |   |   |   |   |   firstName4=n = 0
|   |   |   |   |   |   |   firstName3=n = 1
|   |   |   |   |   |   |   |   lastName0=r = 1: +
|   |   |   |   |   |   |   |   lastName0=r = 0: -
|   |   |   |   |   |   |   firstName3=n = 0
|   |   |   |   |   |   |   |   firstName2=d = 1: +
|   |   |   |   |   |   |   |   firstName2=d = 0: -
```

For **Full decision tree**, the best result is that:
Correctly Classified Instances 43 72.8814 %
Incorrectly Classified Instances 16 27.1186 %

Decision Tree:

ID3

```
firstName1=a = 1
|   lastName1=n = 1: -
|   lastName1=n = 0
|   |   firstName0=g = 1: -
|   |   firstName0=g = 0
|   |   |   lastName1=k = 1: -
|   |   |   lastName1=k = 0
|   |   |   |   lastName3=g = 1: -
|   |   |   |   lastName3=g = 0: +
firstName1=a = 0
|   firstName4=a = 1
|   |   lastName3=g = 1: -
|   |   lastName3=g = 0
|   |   |   firstName2=n = 1
|   |   |   |   firstName3=n = 1: +
|   |   |   |   firstName3=n = 0: -
|   |   |   firstName2=n = 0
|   |   |   |   lastName4=a = 1: -
|   |   |   |   lastName4=a = 0: +
|   firstName4=a = 0
|   |   firstName3=a = 1
|   |   |   firstName0=y = 1: -
|   |   |   firstName0=y = 0
|   |   |   |   lastName1=a = 1: -
|   |   |   |   lastName1=a = 0: +
|   |   firstName3=a = 0
|   |   |   firstName0=a = 1
|   |   |   |   lastName3=m = 1: -
|   |   |   |   lastName3=m = 0: +
|   |   |   firstName0=a = 0
|   |   |   |   firstName2=a = 1: +
|   |   |   |   firstName2=a = 0: -
```

Case B

In case A, if there is no characters in first name or last name, I just simply let its feature to be 0. Another method is to think this as a special character. For example, we can think it as "None", "Null" or " ". In this way, we can get another group of the data.

Table 3: Cross Validation

Algorithm	**CV Accuracy (%)**	**Parameters**
Full Decision tree	66.31	
SGD	65.63	LearningRate=0.005, Threshold=0.001
Decision Stump of 8	62.98	tree depth = 8
Decision Stump of 4	59.87	tree depth = 4
SGD over 100 stumps	54.09	LearningRate=0.005, Threshold=0.001