CIS 419/519: Homework 1

{Zhuheng Jiang}

Although the solutions are entirely my own, I consulted with the following people and sources while working on this homework: {Yongxin Guo, Yihang Xu, Chang Liu, Yupeng Li,
Wikipedia: https://en.wikipedia.org/wiki/ID3$_algorithm$}

1 Decision Tree Learning

a. Show your work:

$$InfoGain(PainLocation) = Info(X) - Info(X|PainLocation)$$

$=(-\frac{5}{14} * log_2\frac{5}{14} - \frac{9}{14} * log_2\frac{9}{14}) - [\frac{5}{14}(-\frac{2}{5} * log_2\frac{2}{5} - \frac{3}{5} * log_2\frac{3}{5}) + \frac{5}{14}(-\frac{2}{5} * log_2\frac{2}{5} - \frac{3}{5} * log_2\frac{3}{5})]$
$= 0.9403 - 0.6935 = 0.2468$

$$InfoGain(Temperature) = Info(X) - Info(X|Tempurature)$$

$=(-\frac{5}{14} * log_2\frac{5}{14} - \frac{9}{14} * log_2\frac{9}{14}) - [\frac{10}{14}(-\frac{3}{14} * log_2\frac{3}{14} - \frac{7}{14} * log_2\frac{7}{14}) + \frac{4}{14}(-\frac{2}{4} * log_2\frac{2}{4} - \frac{1}{2} * log_2\frac{1}{2})]$
$= 0.9403 - 0.9151 = 0.0252$

b. Show your work:

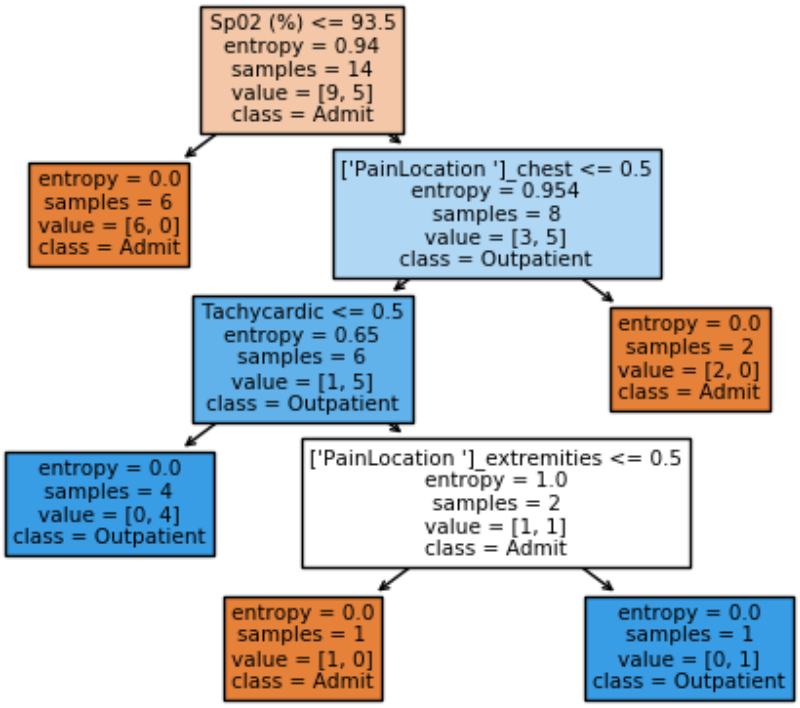
$$GainRatio(PainLocation) = \frac{InfoGain(PainLocation)}{SplitInfo(X, PainLocation)}$$

$$= \frac{0.2468}{-\frac{5}{14} * log_2\frac{5}{14} - \frac{5}{14} * log_2\frac{5}{14} - \frac{4}{14} * log_2\frac{4}{14}} = \frac{0.2468}{1.5774} = 0.1565$$

$$GainRatio(Temperature) = \frac{InfoGain(Temperature)}{SplitInfo(X, Temperature)}$$

$$= \frac{0.2468}{-\frac{10}{14} * log_2\frac{10}{14} - \frac{4}{14} * log_2\frac{4}{14}} = \frac{0.2468}{0.8631} = 0.0292$$

c.



d. No, because firstly, ID3 can converge to local optimal. In every iteration of selecting the local attribute, it will find the currently best choice due to its greedy strategy. On the other hand, ID3 can overfit the data, and usually it will return a relative small decision tree rather than the smallest tree.

2 Decision Trees & Linear Discriminants [CIS 519 ONLY]

A decision tree can include oblique splits by obtain the correlations between features. In classic decision tree algorithm, features are always linearly independent which is reflected in splits as splits parallel to axis. The first thing for obtaining oblique splits is connecting features. For example, when considering both temperature and humidity as features for predicting weather, these two features are definitely not separated since the temperature may cause the humidity to change. Then we need to find the function or the matrix of the features so that we can get the slope, the intercept and other information and plot the oblique splits accordingly.

3 Programming Exercises

Features: What features did you choose and how did you preprocess them?

I attempted three sets of features separately:

DT1. All the pre-processed features out of the entire data set.

DT2. 30 features which have the biggest correlation with the y label "DIABETIC"

DT3. The features introduced in class

In pre-processing section, firstly I removed the columns containing over 50% NaN values, since these features' value are relatively deficient. Then I did one-hot encoding to the data set(concatenating the

dummies code with Dataframe and removing the original string columns). Next, I filled all the NaN value left with the columns' mean value so the data set consists no empty value now. I did not remove the outlier using standard deviation, because if applying 3-δ clear the outlier, then a majority of the data will be removed. Only if $\pm 10\delta$ is applied can the data set be relatively complete.

Parameters: What parameters did you use to train your best decision tree

I only used the 'cpp_alpha' to control how much my decision tree will be pruned, and the Pearson's correlation for picking out 30 most relevant features with label "DIABETIC".

Performance Table:

Feature Set	Accuracy	Conf. Interval [519 ONLY]
DT 1	0.9927641277641278	$\pm$0.0007979423287013635
DT 2	0.9365356265356265	$\pm$0.0019809172502261684
DT 3	0.9148648648648648	$\pm$0.0021284558710378742

Conclusion: What can you conclude from your experience? I conclude that: 1. More features does not mean better. Some time getting rid of some useless data can increase the accuracy. 2. Pruning tree is a tricky thing. Keeping pruning can not increase the performance all the time. From this experience, I get a lot of strategies of processing data. I get to know that the data can be used in so many different way.