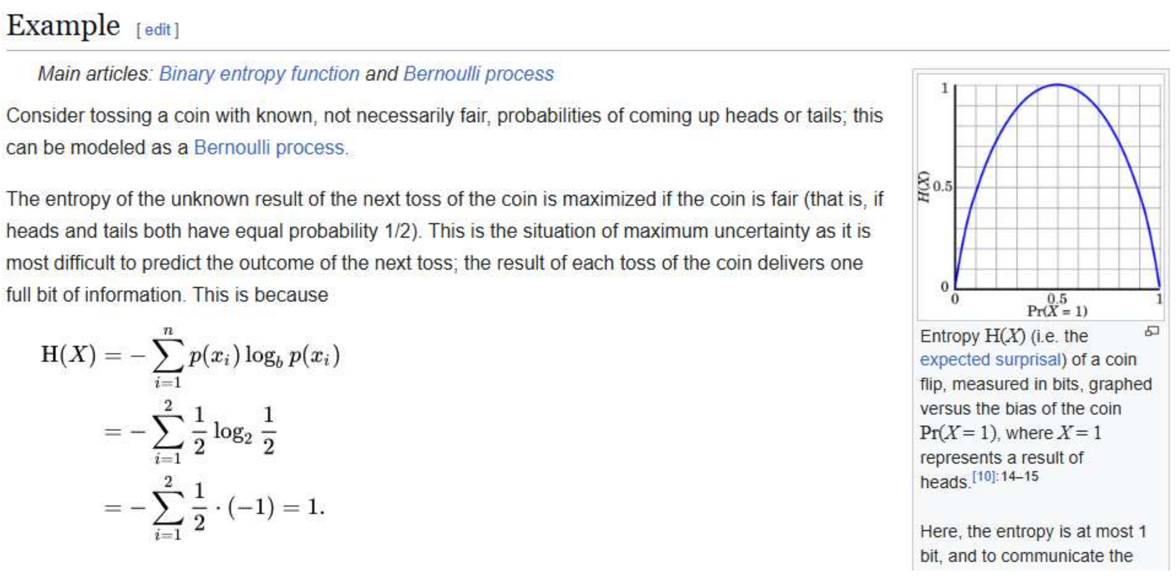
Lab: AI & ML Lab (23A31403)

College: Andhra Engineering College

10. Demonstrate decision tree algorithm for a classification problem and perform parameter tuning for better results

Math

https://en.wikipedia.org/wiki/Entropy_(information_theory)



Example [edit]

Main articles: Binary entropy function and Bernoulli process

Consider tossing a coin with known, not necessarily fair, probabilities of coming up heads or tails; this can be modeled as a Bernoulli process.

The entropy of the unknown result of the next toss of the coin is maximized if the coin is fair (that is, if heads and tails both have equal probability 1/2). This is the situation of maximum uncertainty as it is most difficult to predict the outcome of the next toss; the result of each toss of the coin delivers one full bit of information. This is because

$$\begin{aligned}\mathrm{H}(X) &= -\sum_{i=1}^{n} p(x_i)\log_b p(x_i)\\ &= -\sum_{i=1}^{2}\frac{1}{2}\log_2\frac{1}{2}\\ &= -\sum_{i=1}^{2}\frac{1}{2}\cdot(-1) = 1.\end{aligned}$$

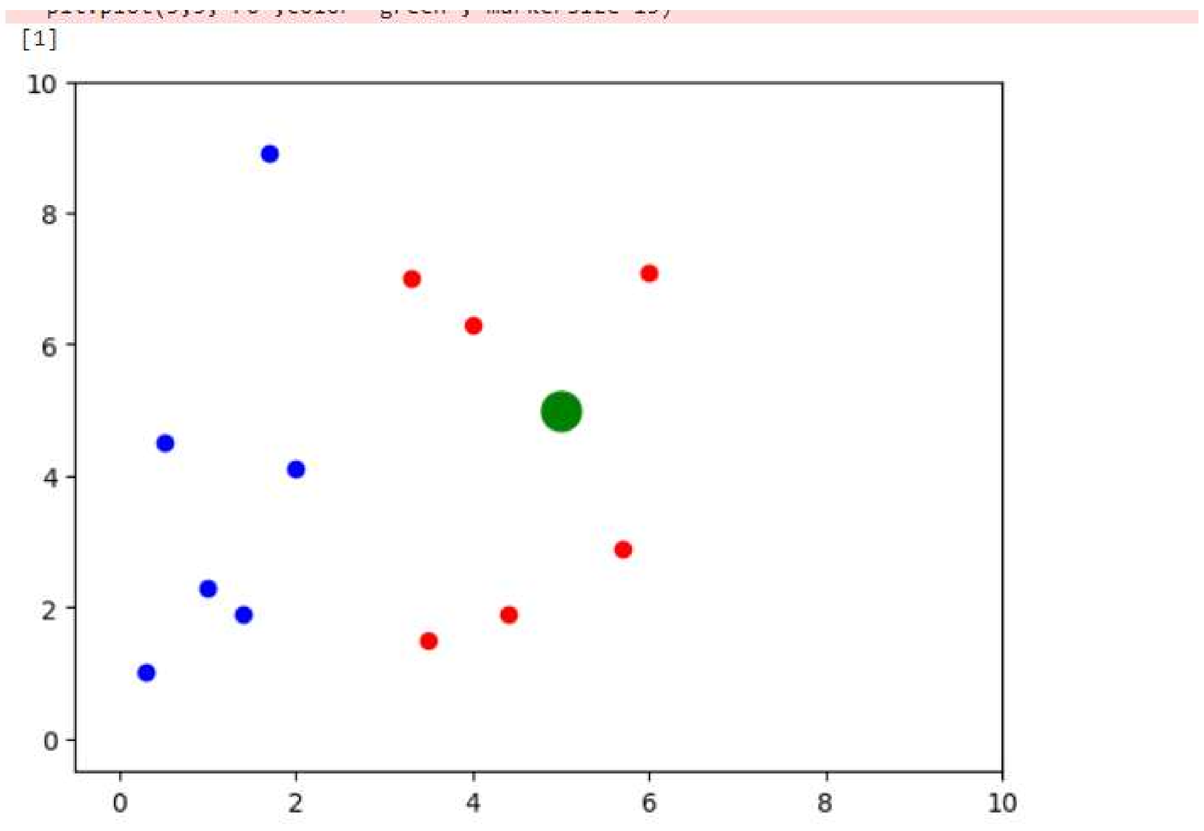
Entropy $\mathrm{H}(X)$ (i.e. the expected surprisal) of a coin flip, measured in bits, graphed versus the bias of the coin $\Pr(X=1)$, where $X=1$ represents a result of heads.[10]: 14–15

Here, the entropy is at most 1 bit, and to communicate the

Code

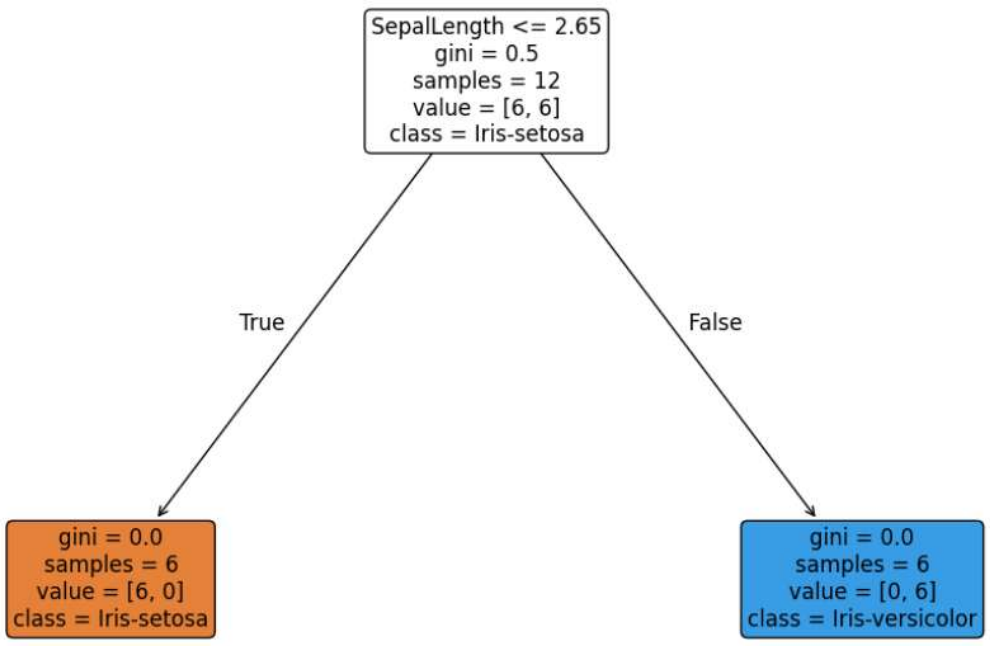
```
import numpy as np
from matplotlib import pyplot as plt
from sklearn import tree

xBlue = np.array([0.3,0.5,1,1.4,1.7,2])
yBlue = np.array([1,4.5,2.3,1.9,8.9,4.1])

xRed = np.array([3.3,3.5,4,4.4,5.7,6])
yRed = np.array([7,1.5,6.3,1.9,2.9,7.1])

X =
np.array([[0.3,1],[0.5,4.5],[1,2.3],[1.4,1.9],[1.7,8.9],[2,4.1],[3.3,7],[3.5,1.5],[4,6.3],[4.4,1.9],[5
.7,2.9],[6,7.1]])
y = np.array([0,0,0,0,0,0,1,1,1,1,1,1]) # 0: blue class, 1: red class
```

```
plt.plot(xBlue, yBlue, 'ro', color = 'blue')
plt.plot(xRed, yRed, 'ro', color='red')

plt.plot(5,5,'ro',color='green', markersize=15)

plt.axis([-0.5,10,-0.5,10])

classifier = tree.DecisionTreeClassifier()
classifier.fit(X,y)

# pred = classifier.predict([5,5])
pred = classifier.predict([[5, 5]])

print(pred)

plt.show()
```



Program

```
from sklearn import tree
import matplotlib.pyplot as plt

plt.figure(figsize=(15,10))
tree.plot_tree(
    classifier,
    feature_names=["SepalLength","SepalWidth","PetalLength","PetalWidth"],
    class_names=irisData['Class'].unique().astype(str),
    filled=True,
    rounded=True,
    fontsize=12
)
plt.show()
```


SepalLength <= 2.65
gini = 0.5
samples = 12
value = [6, 6]
class = Iris-setosa
True
False
gini = 0.0
samples = 6
value = [6, 0]
class = Iris-setosa
gini = 0.0
samples = 6
value = [0, 6]
class = Iris-versicolor