HW3

November 1, 2018

0.1 Problem 1

$$Cost(A) = -1 * (25) + 100 * (25) + 1 * (0) = 2475$$
$$Cost(B) = -1 * (30) + 100 * (20) + 1 * (20) = 1990$$
$$Cost(C) = -1 * (25) + 100 * (25) + 1 * (25) = 2500$$

B would be chosen

0.2 Problem 2

Accuracy = (98 + 143) / (98 + 143 + 37 + 20) = .81
Precision = 98 / (98 + 37) = .73
Recall = 98 / (98 + 20) = .83
Cost =
Sensitivity = Recall
Specificity = (147) / (37 + 143) = .82
False Positive Rate = (37) / (37 + 143) = .21

0.3 Problem 3

Recall = Retrieved and Relevant Documents / Relevant Documents = TP / 10 Precision = Retrieved and Relevant Documents / Retrieved Documents = TP / (TP + FP)

0.3.1 a)

If recall is 40%, then TP = 4.
Precision(A) = 4 / 10
Precision(B) = 4 / 12

0.3.2 b)

If only 15 documents are retrieved, then
Recall(A) = 6 / 10
Precision(A) = 6 / 15
F(A) = .48
Recal(B) = 5 / 10
Precision(B) = 5 / 15
F(B) = .4

0.4 Problem 4

```
In [57]: from sklearn import datasets
         from sklearn import tree
         from sklearn.model_selection import cross_val_score
         import numpy as np

         # import some data to play with
         iris = datasets.load_iris()
```

```
In [58]: print(iris.feature_names)
         print(iris.data[0:10]) # Show 10 exampes
```

```
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]]
```

```
In [59]: clf = tree.DecisionTreeClassifier()
```

```
In [60]: cross_val_score(clf, iris.data, iris.target, cv=3)
```

```
Out[60]: array([0.98039216, 0.92156863, 0.97916667])
```

```
In [61]: cross_val_score(clf, iris.data, iris.target, cv=5)
```

```
Out[61]: array([0.96666667, 0.96666667, 0.9       , 0.96666667, 1.        ])
```

```
In [62]: cross_val_score(clf, iris.data, iris.target, cv=10)
```

```
Out[62]: array([1.        , 0.93333333, 1.        , 0.93333333, 0.93333333,
                0.86666667, 0.93333333, 1.        , 1.        , 1.        ])
```

```
In [67]: clf2 = tree.DecisionTreeClassifier(max_leaf_nodes=5)
```

```
In [68]: cross_val_score(clf2, iris.data, iris.target, cv=3)
```

```
Out[68]: array([0.98039216, 0.92156863, 1.        ])
```

```
In [69]: cross_val_score(clf2, iris.data, iris.target, cv=5)
```

```
Out[69]: array([0.96666667, 0.96666667, 0.9       , 1.        , 1.        ])
```

```
In [70]: cross_val_score(clf2, iris.data, iris.target, cv=10)
```

```
Out[70]: array([1.        , 0.93333333, 1.        , 0.93333333, 0.93333333,
                0.86666667, 0.93333333, 1.        , 1.        , 1.        ])
```

0.5 Problem 5

0.5.1 a)

```
In [4]: import math

        Gain_r1 = 12 * (math.log(12. / 15, 2) - math.log(29. / 50, 2))
        Gain_r1
```

```
Out[4]: 5.567365197117486
```

```
In [5]: Gain_r2 = 7 * (math.log(7. / 10, 2) - math.log(29. / 50, 2))
        Gain_r2
```

```
Out[5]: 1.8991141527217605
```

```
In [ ]: Gain_r3 = 8 * (math.log(8. / 12, 2) - math.log(29. / 50, 2))
        Gain_r3
```

```
In [6]: # R1 is the best rule
```

```
Out[6]: 1.6073015514079714
```

0.5.2 b)

```
In [10]: laplace_r1 = (12 + 1) / (15 + 2)
         laplace_r1
```

```
Out[10]: 0.7647058823529411
```

```
In [11]: laplace_r2 = (7 + 1) / (10 + 2)
         laplace_r2
```

```
Out[11]: 0.6666666666666666
```

```
In [ ]: laplace_r3 = (8 + 1) / (12 + 2)
        laplace_r3
```

```
In [12]: # R1 is the best rule
```

```
Out[12]: 0.6428571428571429
```

0.5.3 c)

```
In [16]: m_r1 = (12 + 2 * 29 / 50) / (15 + 2)
         m_r1
```

```
Out[16]: 0.7741176470588236
```

```
In [17]: m_r2 = (7 + 2 * 29 / 50) / (10 + 2)
         m_r2
```

```
Out[17]: 0.68

In [18]: m_r3 = (8 + 2 * 29 / 50) / (12 + 2)
         m_r3

Out[18]: 0.6542857142857142

In [19]: # R1 is the best rule
```

0.5.4 d)

```
In [20]: accuracy_r1 = (12 / 15)
         accuracy_r1

Out[20]: 0.8

In [21]: accuracy_r2 = (7 / 10)
         accuracy_r2

Out[21]: 0.7

In [22]: accuracy_r3 = (8 / 12)
         accuracy_r3

Out[22]: 0.6666666666666666

In [23]: # R1 is the best rule
```

0.5.5 e)

```
In [27]: accuracy_r1 = (3/3)
         accuracy_r1

Out[27]: 1.0

In [28]: accuracy_r2 = (7/10)
         accuracy_r2

Out[28]: 0.7

In [31]: accuracy_r3 = (6/10)
         accuracy_r3

Out[31]: 0.6

In [32]: # R1 is the best rule
```

0.5.6 f)

```
In [35]: accuracy_r2 = (7/10)
         accuracy_r2

Out[35]: 0.7

In [36]: accuracy_r3 = (6/8)
         accuracy_r3

Out[36]: 0.75

In [37]: # R3 is the best rule
```

0.6 Problem 6

a) 3- : Square 5 : Circle

b)

Eucludean

$$Circles = 1$$
$$Squares = \frac{1}{1^2+1^2} + \frac{1}{2^2+1^2} = \frac{7}{10}$$

Circles win
Manhattan

$$Circles = 1$$
$$Squares = \frac{1}{(1+1)^2} + \frac{1}{(2+1)^2} = .36$$

Circles win again