Beast Cancer Data Analysis Part 2

January 15, 2020

1 Decision Tree

[14]:
```
import pandas as pd
import seaborn as sns # for data visualization
import matplotlib.pyplot as plt # for data visualization
%matplotlib inline
import numpy as np
import os
```

[2]:
```
df = pd.read_csv('Breast_cancer_data.csv')
```

[4]:
```
df.tail()
x = df.iloc[:,0:5]
y = df.iloc[:,5:]
```

[9]:
```
from sklearn.tree import DecisionTreeClassifier
model3 = DecisionTreeClassifier(max_features=3,criterion="entropy")
```

[10]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test=train_test_split(x,y,test_size=0.
 ↪20,random_state=1999)
```

[11]:
```
model3.fit(x_train,y_train)
```

[11]:
```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=None,
                       max_features=3, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')
```

[12]:
```
y_pred3 = model3.predict(x_test)
```

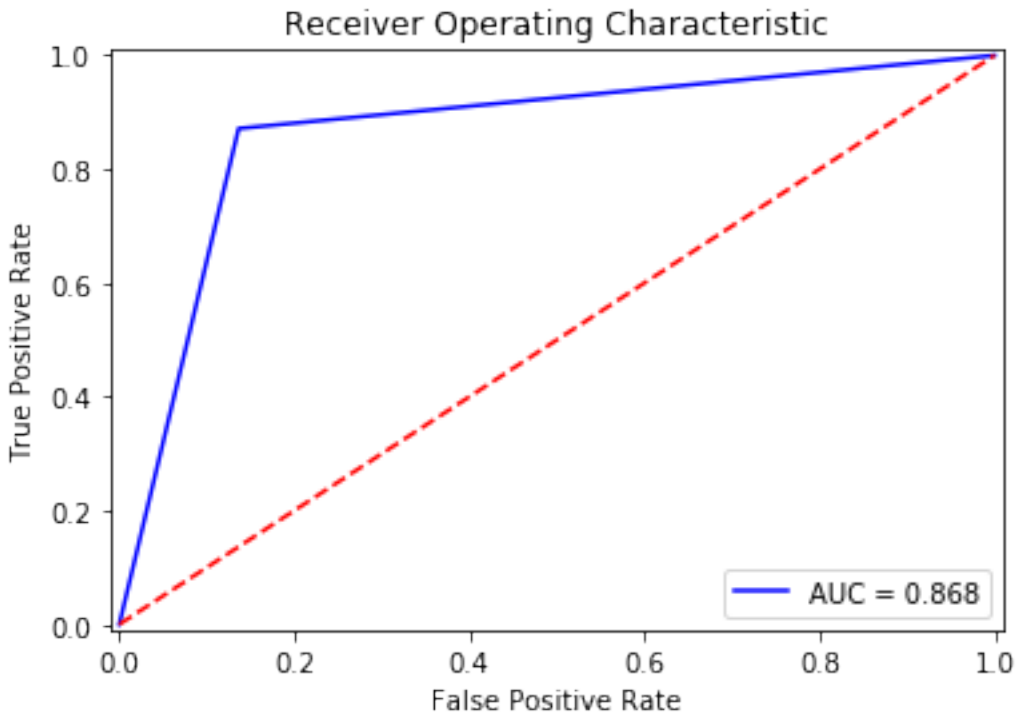
[15]:
```
import sklearn.metrics as metrics
# calculate the fpr and tpr for all thresholds of the classification
probs = model3.predict_proba(x_test) # probabilities for class 0,1
preds = probs[:,1] # probabilities for class 1
```

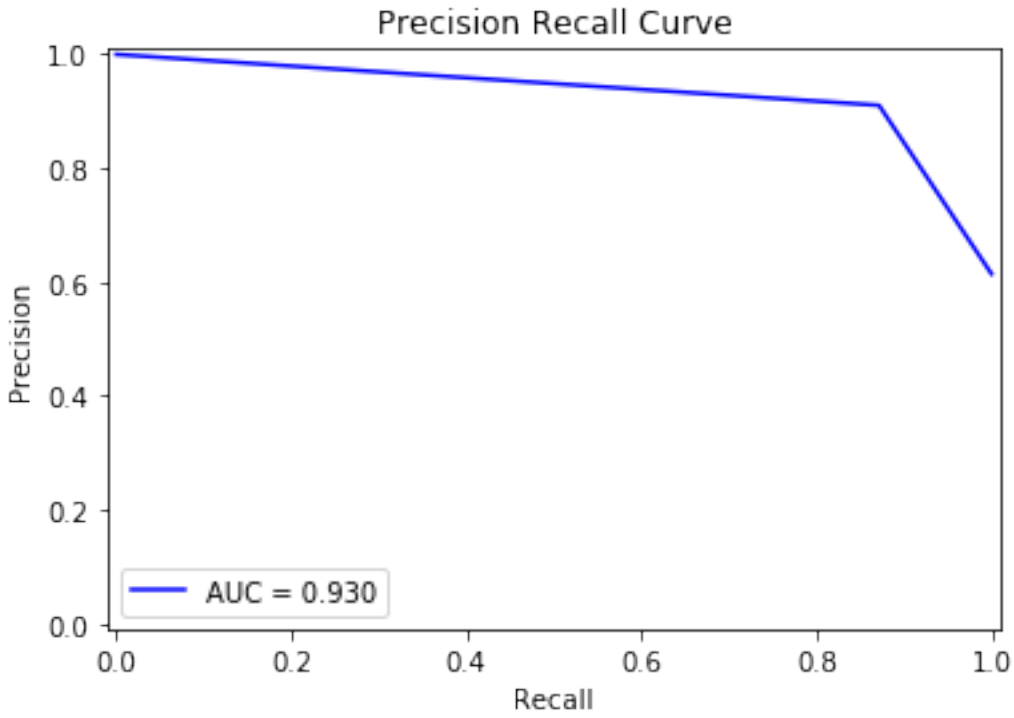
```
fpr, tpr, threshold = metrics.roc_curve(y_test, preds)
roc_auc = metrics.auc(fpr, tpr)

plt.title('Receiver Operating Characteristic')
plt.plot(fpr, tpr, 'b', label = 'AUC = %0.3f' % roc_auc)
plt.legend(loc = 'lower right')
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([-0.01, 1.01])
plt.ylim([-0.01, 1.01])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()

from sklearn.metrics import precision_recall_curve
from sklearn.metrics import auc
precision, recall, thresholds = precision_recall_curve(y_test, preds)

plt.title('Precision Recall Curve')
plt.plot(recall, precision, 'b', label = 'AUC = %0.3f' % auc(recall, precision))
plt.legend(loc = 'lower left')
plt.xlim([-0.01, 1.01])
plt.ylim([-0.01, 1.01])
plt.ylabel('Precision')
plt.xlabel('Recall')
plt.show()
```





[17]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
acc = accuracy_score(y_test, y_pred3)
print("Accuracy score using Logistic Regression:", acc)
```

```
Accuracy score using Logistic Regression: 0.868421052631579
```

[19]:
```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred3))
```

```
              precision    recall  f1-score   support

           0       0.81      0.86      0.84        44
           1       0.91      0.87      0.89        70

    accuracy                           0.87       114
   macro avg       0.86      0.87      0.86       114
weighted avg       0.87      0.87      0.87       114
```

[20]:
```
from sklearn.metrics import log_loss
from sklearn.metrics import f1_score
print("log_loss is", '%.03f' %log_loss(y_test, probs))
print("F1 is", '%.03f' %f1_score(y_test, y_pred3, average='weighted'))
```

```
log_loss is 4.545
F1 is 0.869
```