In [11]:

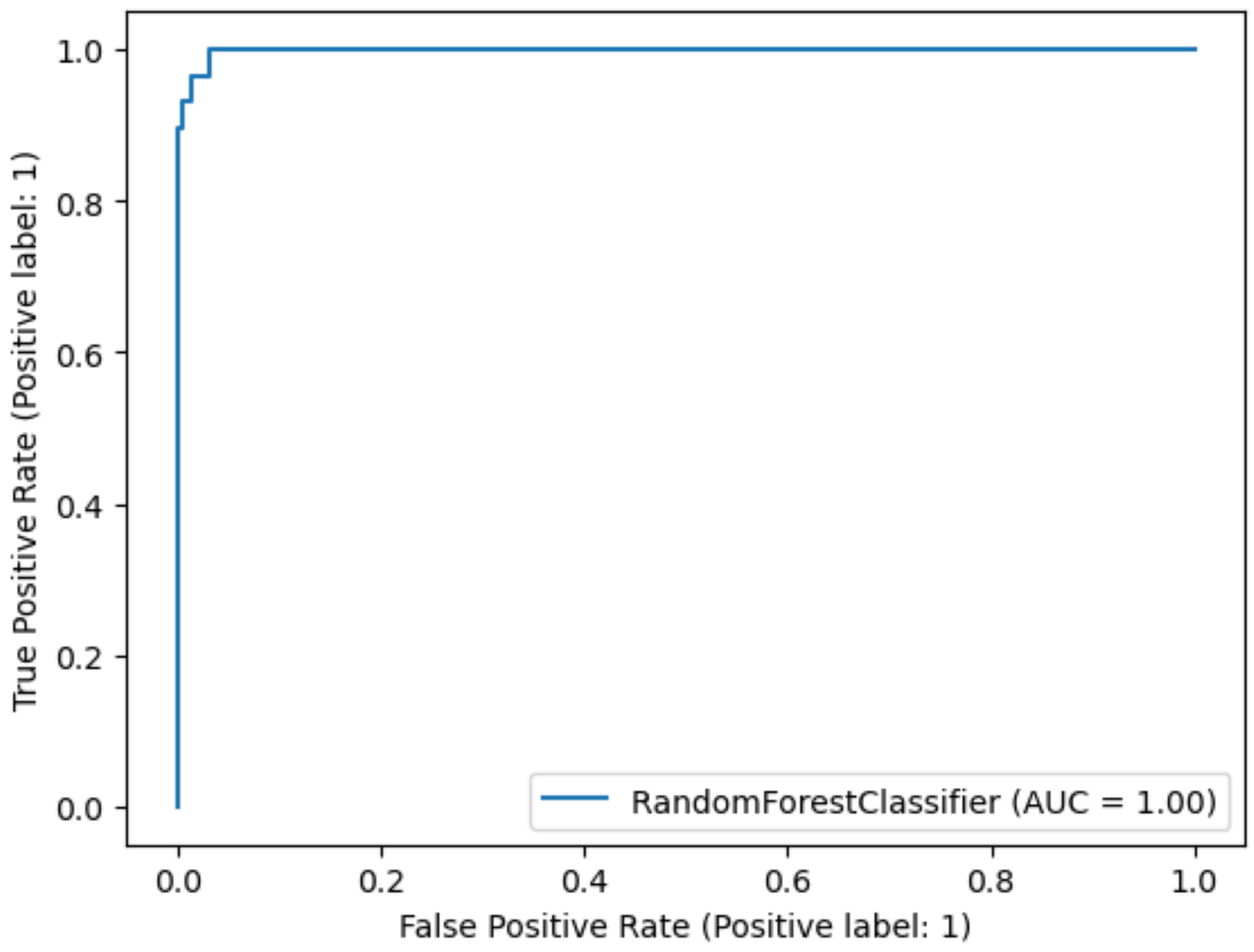
```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import f1_score, confusion_matrix, plot_confusion_matrix, plot_roc_curve, accuracy_score, recall_score, precision_score, balanced_accuracy_score, r
import pandas as pd
from sklearn.metrics import classification_report
import matplotlib.pyplot as plt
import seaborn as sns
#load data from CSV file
data = pd.read_csv("/Users/catherinebetancourt-lee/BMEN 415/fetal_health.csv.csv")

#Merging classification classes into binary
data['fetal_health'] = data['fetal_health'].replace(1.0,0)
data['fetal_health'] = data['fetal_health'].replace(2.0,0)
data['fetal_health'] = data['fetal_health'].replace(3.0,1)

#Separate data and target variables
X = data.drop('fetal_health', axis = 1)
y = data['fetal_health']

#split model into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=42)

#create random forest classifier
rfc = RandomForestClassifier(random_state=42)

#training of the classifier on training data
rfc.fit(X_train, y_train)

#Predictions on test data
y_pred = rfc.predict(X_test)

#metrics
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
f1 = f1_score(y_test, y_pred)
print("F1 Score", f1)
balanced_acc = balanced_accuracy_score(y_test, y_pred)
print("Balanced Accuracy:", balanced_acc)
recall = recall_score(y_test, y_pred)
print("Recall Score:", recall)
precision = precision_score(y_test, y_pred)
print("Precision Score:", precision)
auc = roc_auc_score(y_test, y_pred)
print("AUC Score:", auc)
cm = confusion_matrix(y_test, y_pred)
print("Confusion matrix:")
print(cm)
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.9882629107981221
F1 Score 0.9152542372881356
Balanced Accuracy: 0.9617389038478241
Recall Score: 0.9310344827586207
Precision Score: 0.9
AUC Score: 0.9617389038478241
Confusion matrix:
[[394   3]
 [  2  27]]
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       397
           1       0.90      0.93      0.92        29

    accuracy                           0.99       426
   macro avg       0.95      0.96      0.95       426
weighted avg       0.99      0.99      0.99       426
```

In [8]:

```python
from sklearn import metrics
metrics.plot_roc_curve(rfc, X_test, y_test)
plt.show()
```

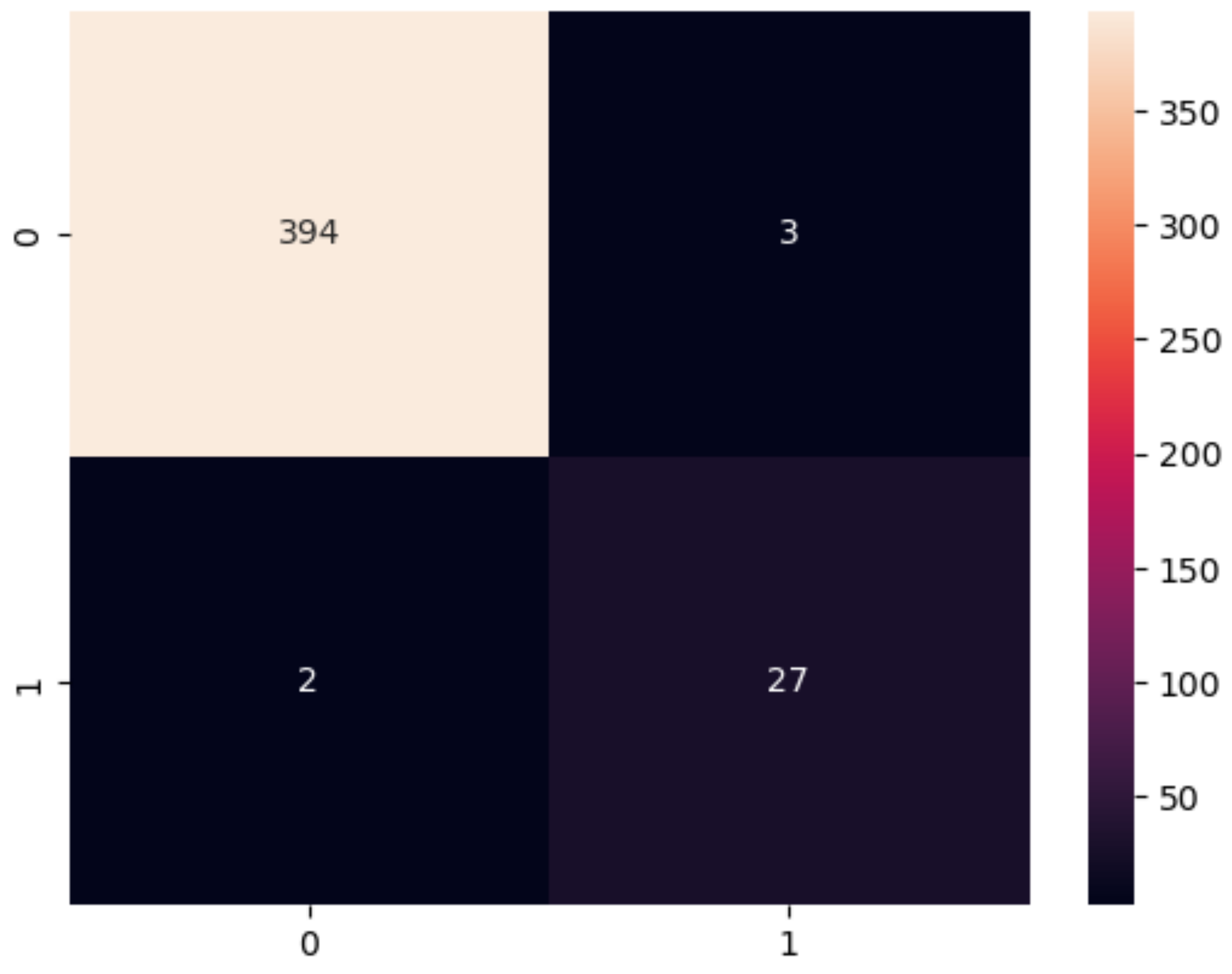
```
/Users/catherinebetancourt-lee/opt/anaconda3/lib/python3.9/site-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function plot_roc_curve is deprecated; Function
:func:`plot_roc_curve` is deprecated in 1.0 and will be removed in 1.2. Use one of the class methods: :meth:`sklearn.metric.RocCurveDisplay.from_predictions` or :meth:`
sklearn.metric.RocCurveDisplay.from_estimator`.
  warnings.warn(msg, category=FutureWarning)
```



In [14]:

```python
sns.heatmap(cm,annot=True, annot_kws={'size':10}, fmt='d')
print(accuracy_score(y_test, y_pred))
```

```
0.9882629107981221
```



394
3
2
27
0
1
0
1
350
300
250
200
150
100
50