

```
In [1]: import numpy as np # Importing NumPy library
import pandas as pd # Importing Pandas library
import matplotlib.pyplot as plt # Importing Matplotlib library's "pyplot" module
import seaborn as sns # Importing Seaborn library

import os
```

```
In [2]: data = pd.read_csv("Classification.csv")
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)
data
```

```
Out[2]:
```

	baseline value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe
0	120	0.000	0.000	0.000	0.000	
1	132	0.006	0.000	0.006	0.003	
2	133	0.003	0.000	0.008	0.003	
3	134	0.003	0.000	0.008	0.003	
4	132	0.007	0.000	0.008	0.000	
...
2121	140	0.000	0.000	0.007	0.000	
2122	140	0.001	0.000	0.007	0.000	
2123	140	0.001	0.000	0.007	0.000	
2124	140	0.001	0.000	0.006	0.000	
2125	142	0.002	0.002	0.008	0.000	

2126 rows x 22 columns

```
In [3]: x = data.drop(["fetal_health"], axis=1)
y = data['fetal_health']
```

```
In [4]: from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random
```

```
In [5]: from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
lda = LinearDiscriminantAnalysis(n_components=1)
lda.fit(x_train, y_train)

# Make predictions on the test set
y_pred=lda.predict(x_test)
```

```
In [6]: #Training Score
lda.score(x_train, y_train)*100
```

```
Out[6]: 96.94117647058823
```

```
In [7]: #Testing Score
lda.score(x_test, y_test)*100
```

```
Out[7]: 95.77464788732394
```

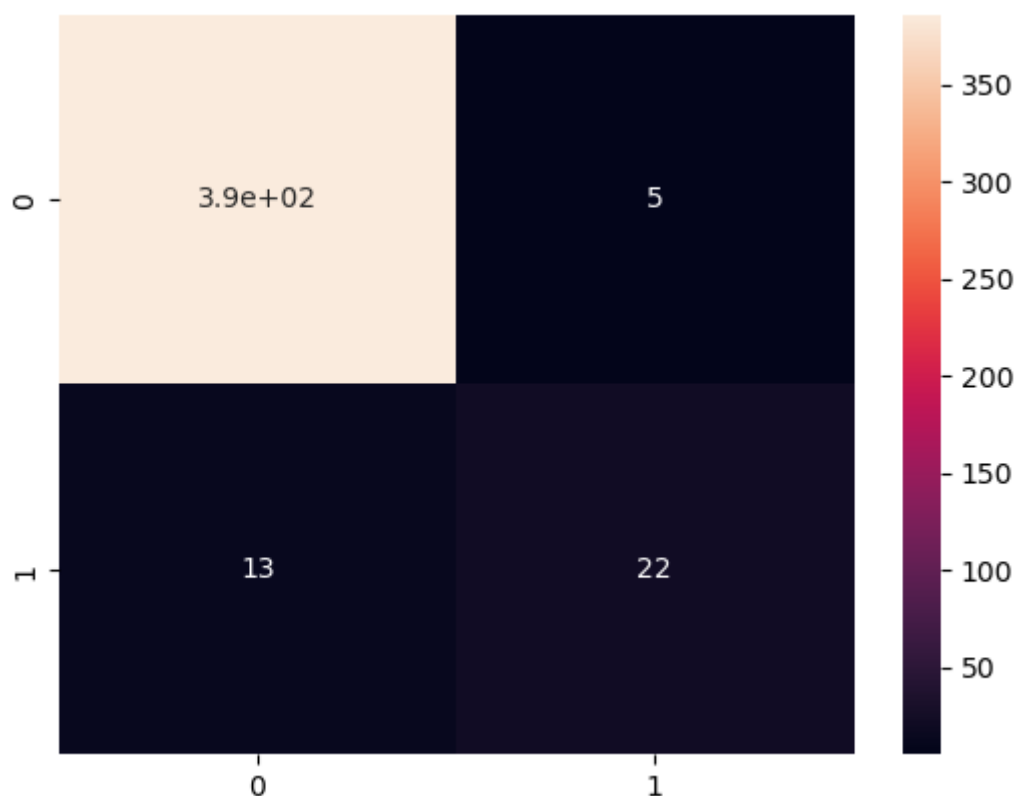
```
In [8]: from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score

cm = confusion_matrix(y_test, y_pred)

accuracy_score(y_test, y_pred)

sns.heatmap(cm, annot=True)
```

```
Out[8]: <AxesSubplot:>
```

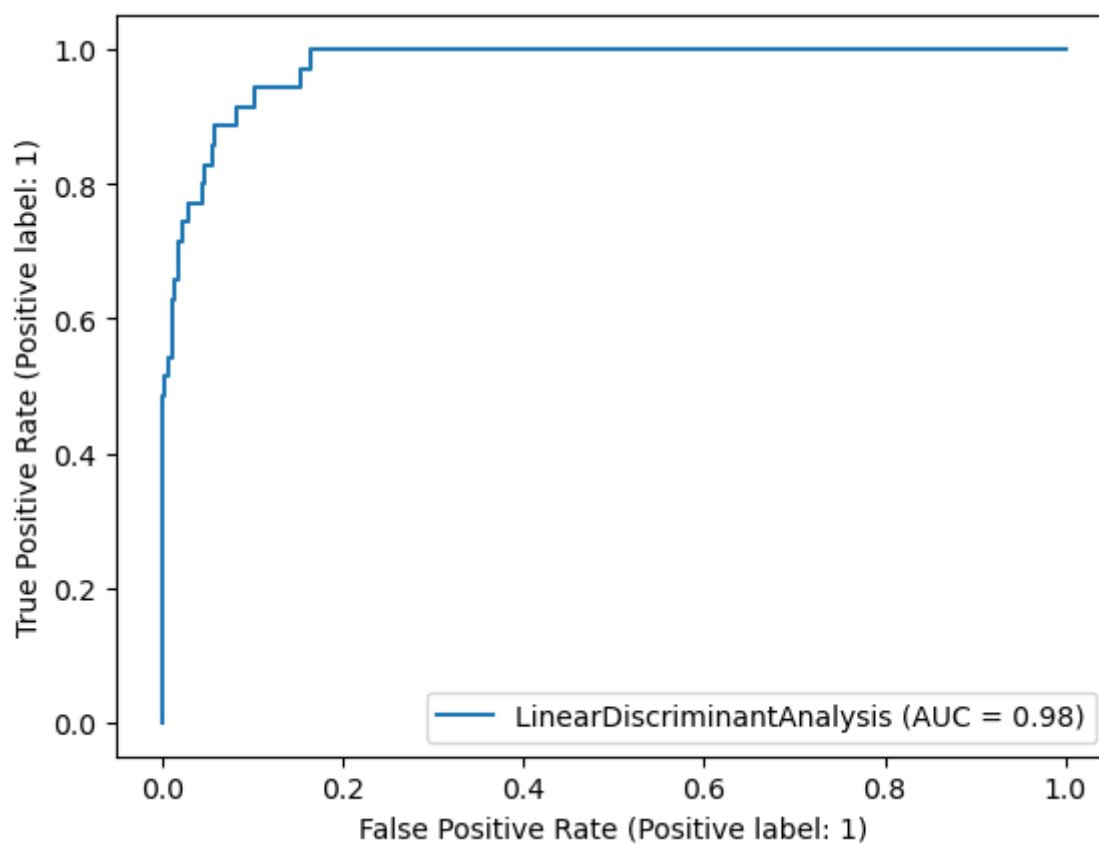


```
In [9]: from sklearn.metrics import classification_report
#evaluate model performance
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.97	0.99	0.98	391
1	0.81	0.63	0.71	35
accuracy			0.96	426
macro avg	0.89	0.81	0.84	426
weighted avg	0.95	0.96	0.96	426

```
In [10]: from sklearn.metrics import RocCurveDisplay
```

```
RocCurveDisplay.from_estimator(lda, x_test, y_test)  
plt.show()
```



In []: