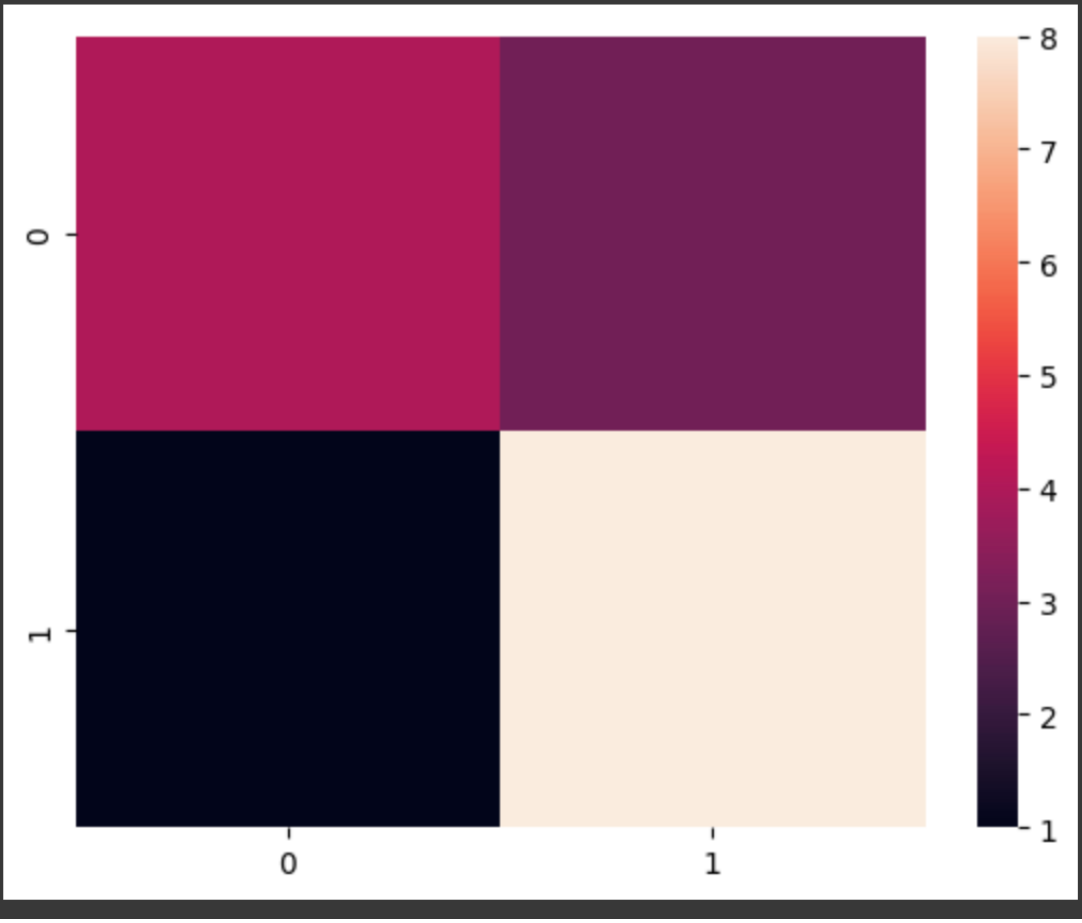
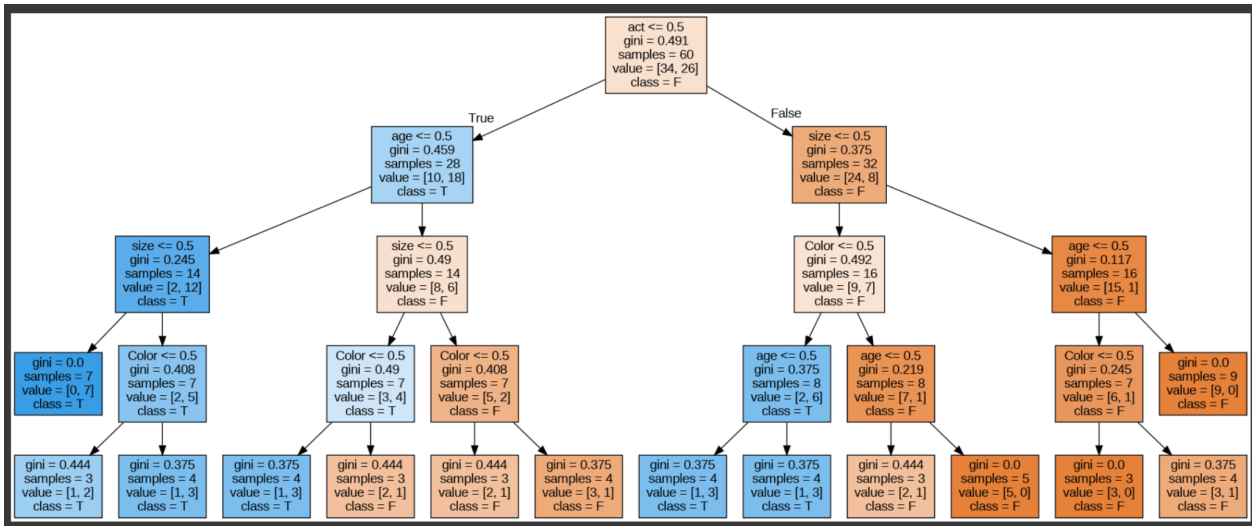
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.tree import export_graphviz
import graphviz
import sklearn.tree as tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix
from matplotlib import pyplot as plt
import seaborn as sns

df = pd.read_csv('/content/balloons.csv')
for i in range(len(df)):
  if df['Color'][i]=='YELLOW':
    df['Color'][i]=0
  else:
    df['Color'][i]=1
  if df['size'][i]=='SMALL':
    df['size'][i]=0
  else:
    df['size'][i]=1
  if df['act'][i]=='STRETCH':
    df['act'][i]=0
  else:
    df['act'][i]=1
  if df['age'][i]=='ADULT':
    df['age'][i]=0
  else:
    df['age'][i]=1
X = np.array(df.iloc[:, 0:4])
y = np.array(df.iloc[:, 4])
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
DT = DecisionTreeClassifier()
DT.fit(X_train, y_train)
pred = DT.predict(X_test)
print ('Accuracy Score: ', accuracy_score(y_test, pred))
print('\nConfusion Matrix\n', confusion_matrix(y_test, pred))
print(f'Feature Importance: [Color, size, age, act]{DT.feature_importances_}')
export_graphviz(DT, out_file='DecisionTree.dot')
with open('DecisionTree.dot') as f:
  dot_graph = f.read()
g = graphviz.Source(dot_graph)
g.render()
text_representation = tree.export_text(DT)
dot_data=tree.export_graphviz(DT, out_file=None, feature_names=['Color', 'size', 'age', 'act'], class_names=['F', 'T'],
graph = graphviz.Source(dot_data, format='png')
graph.render('balloons_dt', view=True)
text_representation = tree.export_text(DT)
print(text_representation)
sns.heatmap(confusion_matrix(y_test, pred))
plt.show()
```

```
Accuracy Score:  0.75

Confusion Matrix
 [[4 3]
 [1 8]]
Feature Importance: [Color, size, age, act][0.26313719 0.22419761 0.20807111 0.30459408]
|--- feature_3 <= 0.50
|   |--- feature_2 <= 0.50
|   |   |--- feature_1 <= 0.50
|   |   |   |--- class: T
|   |   |--- feature_1 >  0.50
|   |   |   |--- feature_0 <= 0.50
|   |   |   |   |--- class: T
|   |   |   |--- feature_0 >  0.50
|   |   |   |   |--- class: T
|   |--- feature_2 >  0.50
|   |   |--- feature_1 <= 0.50
|   |   |   |--- feature_0 <= 0.50
|   |   |   |   |--- class: T
|   |   |   |--- feature_0 >  0.50
|   |   |   |   |--- class: F
|   |   |--- feature_1 >  0.50
|   |   |   |--- feature_0 <= 0.50
|   |   |   |   |--- class: F
|   |   |   |--- feature_0 >  0.50
|   |   |   |   |--- class: F
|--- feature_3 >  0.50
|   |--- feature_1 <= 0.50
|   |   |--- feature_0 <= 0.50
|   |   |   |--- feature_2 <= 0.50
|   |   |   |   |--- class: T
|   |   |   |--- feature_2 >  0.50
|   |   |   |   |--- class: T
|   |   |--- feature_0 >  0.50
|   |   |   |--- feature_2 <= 0.50
|   |   |   |   |--- class: F
|   |   |   |--- feature_2 >  0.50
|   |   |   |   |--- class: F
|   |--- feature_1 >  0.50
```

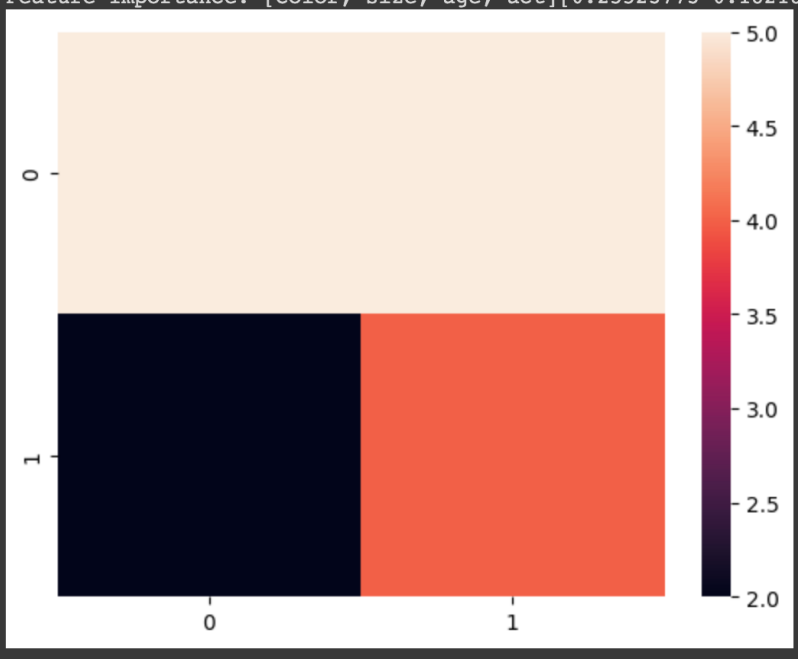
```
|   |   |--- feature_2 <= 0.50
|   |   |   |--- feature_0 <= 0.50
|   |   |   |   |--- class: F
|   |   |   |--- feature_0 >  0.50
|   |   |   |   |--- class: F
|   |   |--- feature_2 >  0.50
|   |   |   |--- class: F
```




act <= 0.5
gini = 0.491
samples = 60
value = [34, 26]
class = F
True
False
age <= 0.5
gini = 0.459
samples = 28
value = [10, 18]
class = T
size <= 0.5
gini = 0.375
samples = 32
value = [24, 8]
class = F
size <= 0.5
gini = 0.245
samples = 14
value = [2, 12]
class = T
size <= 0.5
gini = 0.49
samples = 14
value = [8, 6]
class = F
Color <= 0.5
gini = 0.492
samples = 16
value = [9, 7]
class = F
age <= 0.5
gini = 0.117
samples = 16
value = [15, 1]
class = F
gini = 0.0
samples = 7
value = [0, 7]
class = T
Color <= 0.5
gini = 0.408
samples = 7
value = [2, 5]
class = T
Color <= 0.5
gini = 0.49
samples = 7
value = [3, 4]
class = T
Color <= 0.5
gini = 0.408
samples = 7
value = [5, 2]
class = F
age <= 0.5
gini = 0.375
samples = 8
value = [2, 6]
class = T
age <= 0.5
gini = 0.219
samples = 8
value = [7, 1]
class = F
Color <= 0.5
gini = 0.245
samples = 7
value = [6, 1]
class = F
gini = 0.0
samples = 9
value = [9, 0]
class = F
gini = 0.444
samples = 3
value = [1, 2]
class = T
gini = 0.375
samples = 4
value = [1, 3]
class = T
gini = 0.375
samples = 4
value = [1, 3]
class = T
gini = 0.444
samples = 3
value = [2, 1]
class = F
gini = 0.444
samples = 3
value = [2, 1]
class = F
gini = 0.375
samples = 4
value = [3, 1]
class = F
gini = 0.375
samples = 4
value = [1, 3]
class = T
gini = 0.375
samples = 4
value = [1, 3]
class = T
gini = 0.444
samples = 3
value = [2, 1]
class = F
gini = 0.0
samples = 5
value = [5, 0]
class = F
gini = 0.0
samples = 3
value = [3, 0]
class = F
gini = 0.375
samples = 4
value = [3, 1]
class = F

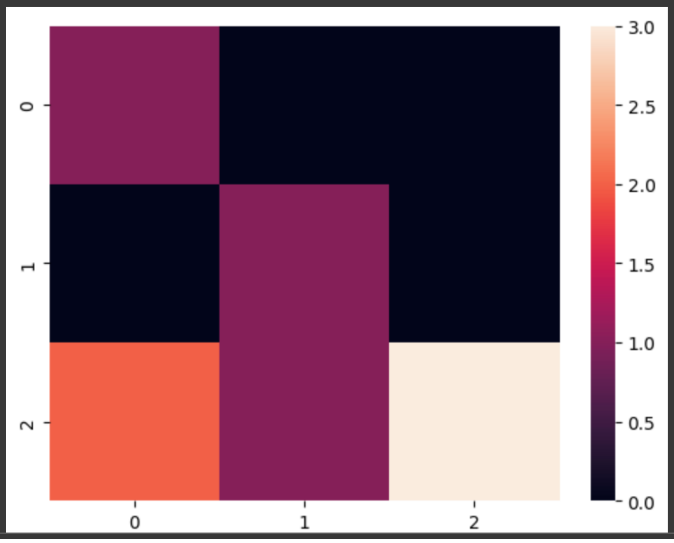
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
from matplotlib import pyplot as plt
import seaborn as sns
df = pd.read_csv('/content/balloons.csv')
for i in range(len(df)):
  if df['Color'][i]=='YELLOW':
    df['Color'][i]=0
  else:
    df['Color'][i]=1
  if df['size'][i]=='SMALL':
    df['size'][i]=0
  else:
    df['size'][i]=1
  if df['act'][i]=='STRETCH':
    df['act'][i]=0
  else:
    df['act'][i]=1
  if df['age'][i]=='ADULT':
    df['age'][i]=0
  else:
    df['age'][i]=1
X = np.array(df.iloc[:, 0:4])
y = np.array(df.iloc[:, 4])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
RF = RandomForestClassifier(n_estimators=500)
RF.fit(X_train, y_train)
pred = RF.predict(X_test)
print ('Accuracy Score: ', accuracy_score(y_test, pred))
print('\nConfusion Matrix\n', confusion_matrix(y_test, pred))
print(f'Feature Importance: [Color, size, age, act]{RF.feature_importances_}')
sns.heatmap(confusion_matrix(y_test, pred))
plt.show()
```

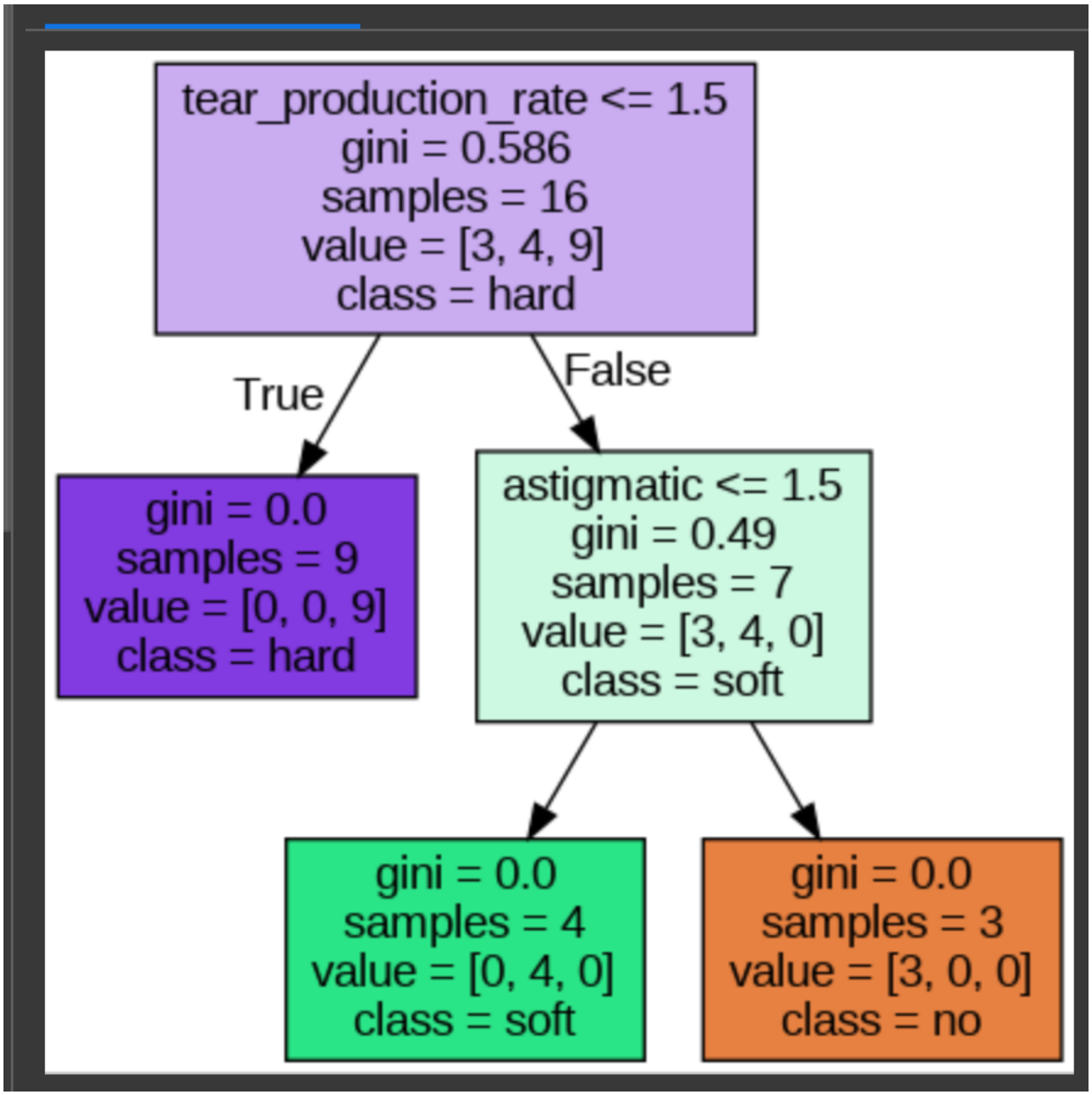
```
Accuracy Score:  0.5625

Confusion Matrix
 [[5 5]
 [2 4]]
Feature Importance: [Color, size, age, act][0.25523773 0.18218883 0.199912   0.36266144]
```




```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.tree import export_graphviz
import graphviz
import sklearn.tree as tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix
from matplotlib import pyplot as plt
import seaborn as sns
names=['n/a','age','spectacle','astigmatic','tear production rate','Class']
df = pd.read_csv('/content/lenses.csv',names=names)
df.drop(df.columns[[0]], axis=1, inplace=True)
X = np.array(df.iloc[:, 0:4])
y = np.array(df.iloc[:, 4])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)
DT = DecisionTreeClassifier()
DT.fit(X_train, y_train)
pred = DT.predict(X_test)
print ('Accuracy Score: ', accuracy_score(y_test, pred))
print('\nConfusion Matrix\n', confusion_matrix(y_test, pred))
print(f'Feature Importance: [age,spectacle,astigmatic,tear production rate]{DT.feature_importances_}')
dot_data=tree.export_graphviz(DT, out_file=None, feature_names=['age','spectacle','astigmatic','tear_production_rate'], class_names=['no','soft','hard'], fi
graph = graphviz.Source(dot_data, format='png')
graph.render('lenses_dt', view=True)
text_representation = tree.export_text(DT)
print(text_representation)
sns.heatmap(confusion_matrix(y_test, pred))
plt.show()
```

```
Accuracy Score:  0.625

Confusion Matrix
 [[1 0 0]
 [0 1 0]
 [2 1 3]]
Feature Importance: [age,spectacle,astigmatic,tear production rate][0.         0.         0.36571429 0.63428571]
|--- feature_3 <= 1.50
|   |--- class: 3
|--- feature_3 >  1.50
|   |--- feature_2 <= 1.50
|   |   |--- class: 2
|   |--- feature_2 >  1.50
|   |   |--- class: 1
```




tear_production_rate <= 1.5
gini = 0.586
samples = 16
value = [3, 4, 9]
class = hard
True
False
gini = 0.0
samples = 9
value = [0, 0, 9]
class = hard
astigmatic <= 1.5
gini = 0.49
samples = 7
value = [3, 4, 0]
class = soft
gini = 0.0
samples = 4
value = [0, 4, 0]
class = soft
gini = 0.0
samples = 3
value = [3, 0, 0]
class = no

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
from matplotlib import pyplot as plt
import seaborn as sns
names=['n/a','age','spectacle','astigmatic','tear production rate','Class']
df = pd.read_csv('/content/lenses.csv',names=names)
df.drop(df.columns[[0]], axis=1, inplace=True)
X = np.array(df.iloc[:, 0:4])
y = np.array(df.iloc[:, 4])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)
RF = RandomForestClassifier()
RF.fit(X_train, y_train)
pred = RF.predict(X_test)
print ('Accuracy Score: ', accuracy_score(y_test, pred))
print('\nConfusion Matrix\n', confusion_matrix(y_test, pred))
print(f'Feature Importance: [age,spectacle,astigmatic,tear production rate]{RF.feature_importances_}')
sns.heatmap(confusion_matrix(y_test, pred))
plt.show()
```

```
Accuracy Score:  0.875

Confusion Matrix
 [[1 0 0]
 [0 0 1]
 [0 0 6]]
Feature Importance: [age,spectacle,astigmatic,tear production rate][0.23775969 0.10569694 0.24667946 0.40986391]
```