```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df = pd.read_csv('/content/kyphosis.csv')
```

```
df.head()
```

	Kyphosis	Age	Number	Start
0	absent	71	3	5
1	absent	158	3	14
2	present	128	4	5
3	absent	2	5	1
4	absent	1	4	15

Next steps: Generate code with df View recommended plots

```
df.info()
```

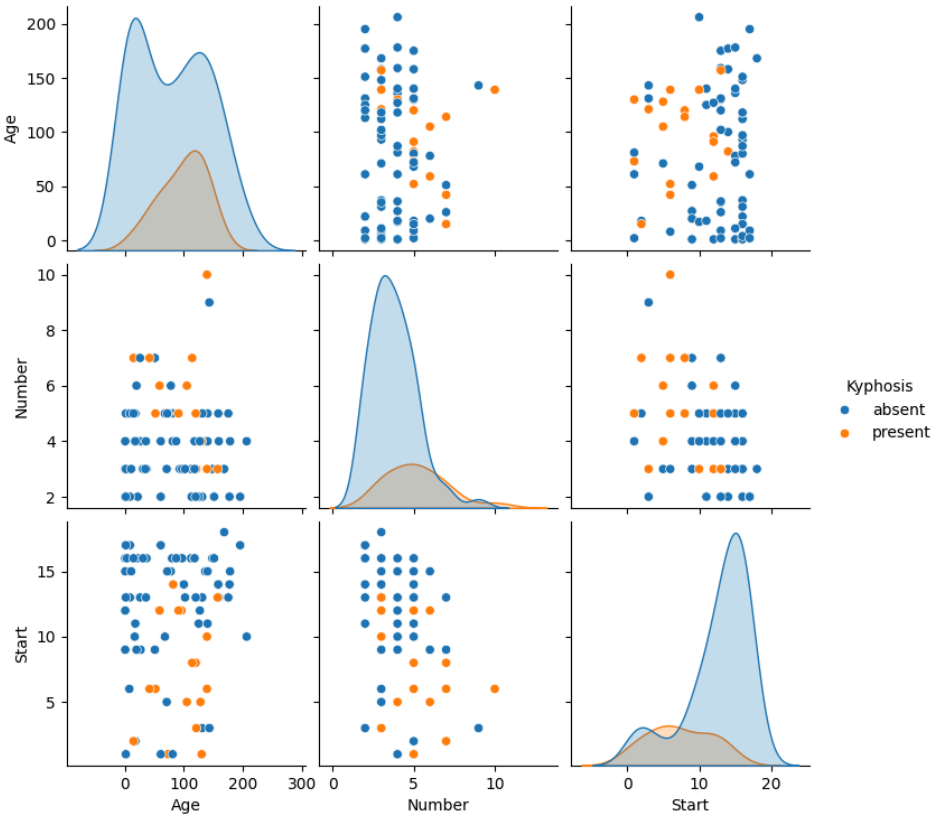
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 81 entries, 0 to 80
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Kyphosis  81 non-null     object
 1   Age       81 non-null     int64
 2   Number    81 non-null     int64
 3   Start     81 non-null     int64
dtypes: int64(3), object(1)
memory usage: 2.7+ KB
```

```
df.describe()
```

	Age	Number	Start
count	81.000000	81.000000	81.000000
mean	83.654321	4.049383	11.493827
std	58.104251	1.619423	4.883962
min	1.000000	2.000000	1.000000
25%	26.000000	3.000000	9.000000
50%	87.000000	4.000000	13.000000
75%	130.000000	5.000000	16.000000
max	206.000000	10.000000	18.000000

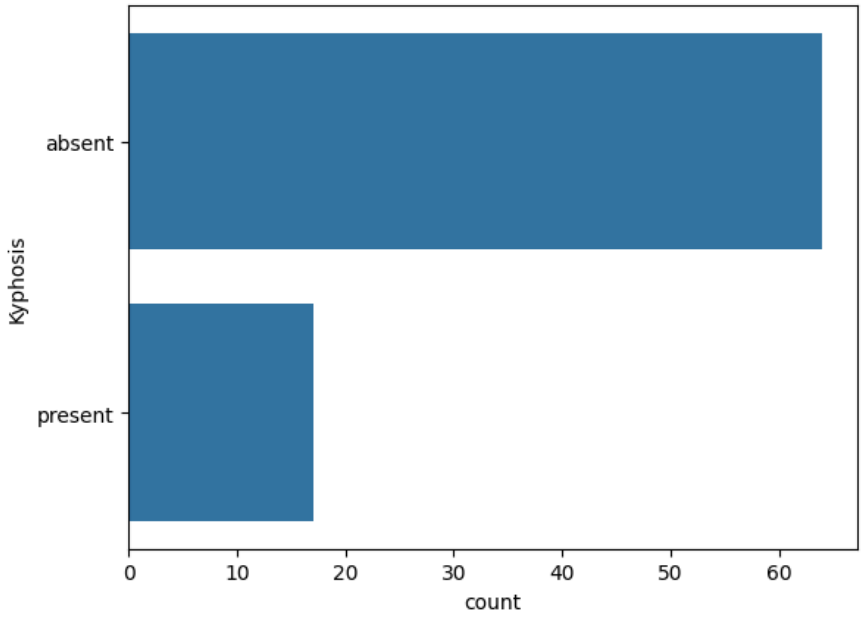
```
sns.pairplot(df, hue = 'Kyphosis')
```

```
<seaborn.axisgrid.PairGrid at 0x7ff159264790>
```




```
sns.countplot(df['Kyphosis'])
```

```
<Axes: xlabel='count', ylabel='Kyphosis'>
```



Train Test Split

```
from sklearn.model_selection import train_test_split
```

```
X = df.drop('Kyphosis', axis = 1)
y = df['Kyphosis']
```

```
X_train, X_test, y_train, y_test = train_test_split( X, y,test_size=0.3, random_state=101)
```

Train the model

```
from sklearn.tree import DecisionTreeClassifier
```

```
dtree = DecisionTreeClassifier(criterion= 'entropy')
```

```
dtree.fit(X_train, y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy')
```

Prediction evaluation

```
predictions = dtree.predict(X_test)
```

```
from sklearn.metrics import classification_report,confusion_matrix
```

```
confusion_matrix(y_test,predictions)
```

```
array([[15,  2],
       [ 2,  6]])
```

```
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

      absent       0.88      0.88      0.88        17
     present       0.75      0.75      0.75         8

    accuracy                           0.84        25
   macro avg       0.82      0.82      0.82        25
weighted avg       0.84      0.84      0.84        25
```

```
from IPython.display import Image
from six import StringIO
from sklearn.tree import export_graphviz
import pydot

features = list(df.columns[1:])

features
```

```
['Age', 'Number', 'Start']
```

```
df.columns
```

```
Index(['Kyphosis', 'Age', 'Number', 'Start'], dtype='object')
```

```
dot_data = StringIO()
export_graphviz(dtree,out_file = dot_data, feature_names = features, filled = True, rounded = True)

graph = pydot.graph_from_dot_data(dot_data.getvalue())
```

```
Image(graph[0].create_png())
```

Show hidden output

```
from sklearn.ensemble import RandomForestClassifier
```

```
rfc = RandomForestClassifier(n_estimators=100)
rfc.fit(X_train,y_train)
```

```
RandomForestClassifier
RandomForestClassifier()
```

```
rfc_pred = rfc.predict(X_test)
```

```
confusion_matrix(y_test,rfc_pred)
```

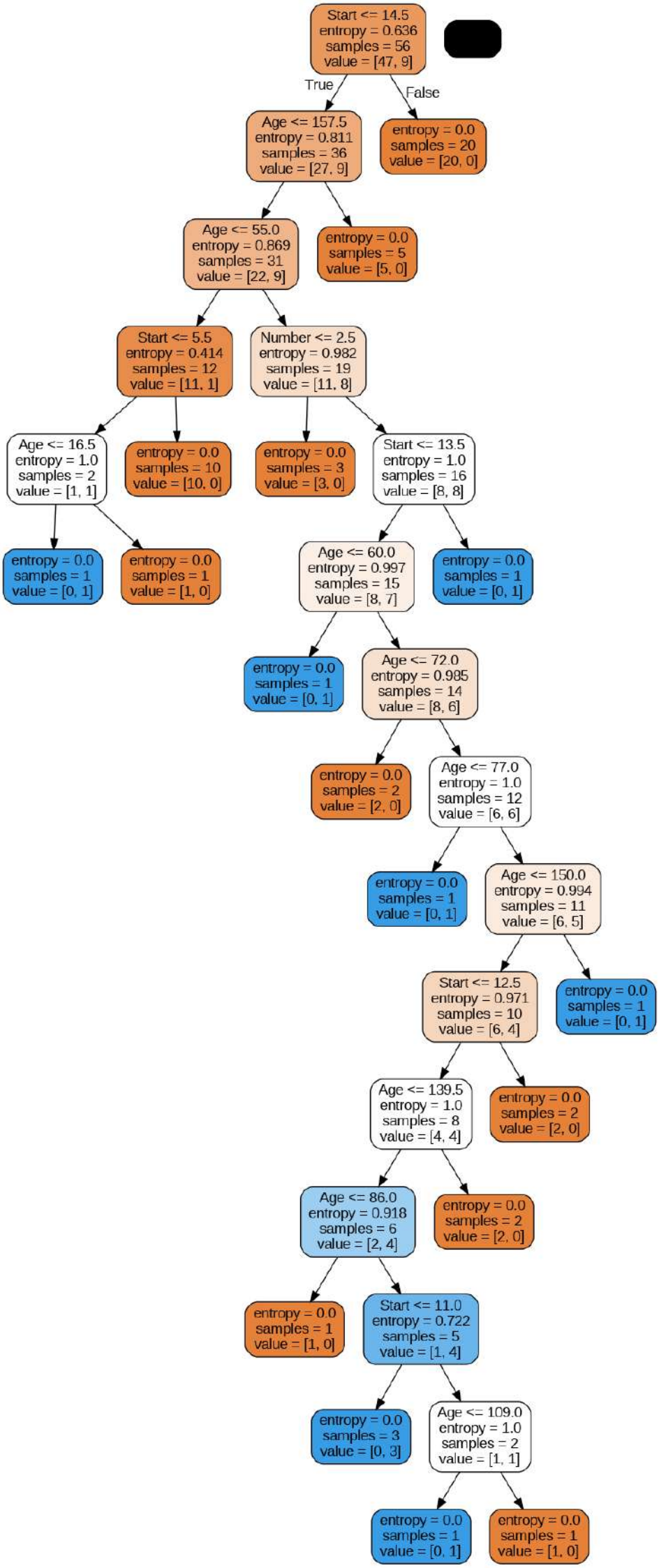
```
array([[16,  1],
       [ 6,  2]])
```

```
print(classification_report(y_test,rfc_pred))
```

```
              precision    recall  f1-score   support

      absent       0.73      0.94      0.82        17
     present       0.67      0.25      0.36         8

    accuracy                           0.72        25
   macro avg       0.70      0.60      0.59        25
weighted avg       0.71      0.72      0.67        25
```


Start <= 14.5
entropy = 0.636
samples = 56
value = [47, 9]
True
False
Age <= 157.5
entropy = 0.811
samples = 36
value = [27, 9]
entropy = 0.0
samples = 20
value = [20, 0]
Age <= 55.0
entropy = 0.869
samples = 31
value = [22, 9]
entropy = 0.0
samples = 5
value = [5, 0]
Start <= 5.5
entropy = 0.414
samples = 12
value = [11, 1]
Number <= 2.5
entropy = 0.982
samples = 19
value = [11, 8]
Age <= 16.5
entropy = 1.0
samples = 2
value = [1, 1]
entropy = 0.0
samples = 10
value = [10, 0]
entropy = 0.0
samples = 3
value = [3, 0]
Start <= 13.5
entropy = 1.0
samples = 16
value = [8, 8]
entropy = 0.0
samples = 1
value = [0, 1]
entropy = 0.0
samples = 1
value = [1, 0]
Age <= 60.0
entropy = 0.997
samples = 15
value = [8, 7]
entropy = 0.0
samples = 1
value = [0, 1]
entropy = 0.0
samples = 1
value = [0, 1]
Age <= 72.0
entropy = 0.985
samples = 14
value = [8, 6]
entropy = 0.0
samples = 2
value = [2, 0]
Age <= 77.0
entropy = 1.0
samples = 12
value = [6, 6]
entropy = 0.0
samples = 1
value = [0, 1]
Age <= 150.0
entropy = 0.994
samples = 11
value = [6, 5]
Start <= 12.5
entropy = 0.971
samples = 10
value = [6, 4]
entropy = 0.0
samples = 1
value = [0, 1]
Age <= 139.5
entropy = 1.0
samples = 8
value = [4, 4]
entropy = 0.0
samples = 2
value = [2, 0]
Age <= 86.0
entropy = 0.918
samples = 6
value = [2, 4]
entropy = 0.0
samples = 2
value = [2, 0]
entropy = 0.0
samples = 1
value = [1, 0]
Start <= 11.0
entropy = 0.722
samples = 5
value = [1, 4]
entropy = 0.0
samples = 3
value = [0, 3]
Age <= 109.0
entropy = 1.0
samples = 2
value = [1, 1]
entropy = 0.0
samples = 1
value = [0, 1]
entropy = 0.0
samples = 1
value = [1, 0]