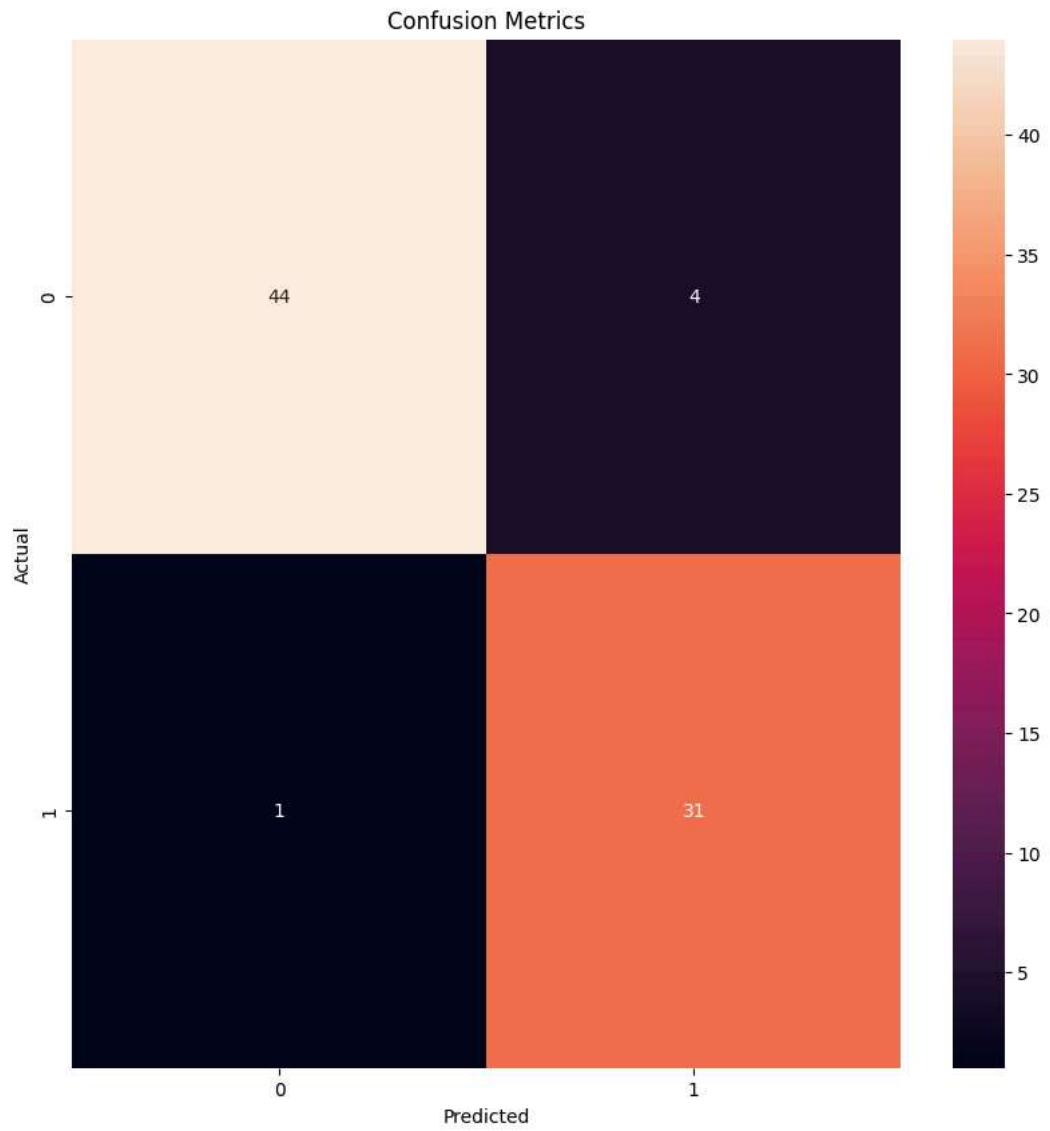
```
import pandas as pd


df=pd.read_csv("/content/Admission_Predict.csv")


from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix, classification_report
import matplotlib.pyplot as plt
import seaborn as sns


df['Chance of Admit ']=df['Chance of Admit '].apply(lambda x:1 if x>0.75 else 0)


X=df[["GRE Score","TOEFL Score","University Rating","SOP","LOR ","CGPA","Research"]]
y=df["Chance of Admit "]


scaler=StandardScaler()
X_scaled=scaler.fit_transform(X)


xtrain,xtest,ytrain,ytest=train_test_split(X_scaled,y,test_size=0.2,random_state=42)


clf=DecisionTreeClassifier()
clf.fit(xtrain,ytrain)
```

```
DecisionTreeClassifier
DecisionTreeClassifier()
```

```
ypred=clf.predict(xtest)


print(classification_report(ytest,ypred))
```

```
              precision    recall  f1-score   support

           0       0.98      0.92      0.95        48
           1       0.89      0.97      0.93        32

    accuracy                           0.94        80
   macro avg       0.93      0.94      0.94        80
weighted avg       0.94      0.94      0.94        80
```

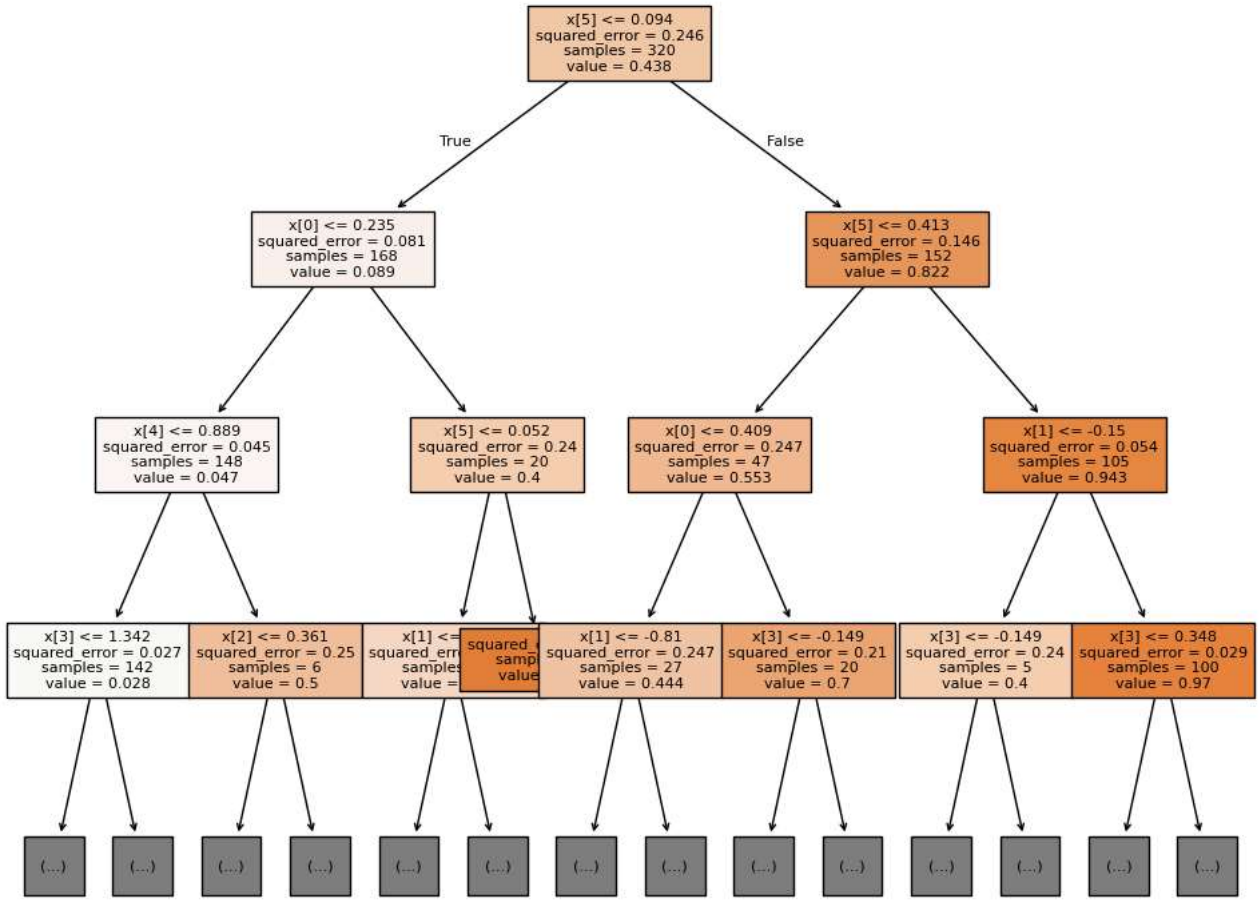
```


)


cm=confusion_matrix(ytest,ypred)


plt.figure(figsize=(10,10))
sns.heatmap(cm,annot=True)
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Metrics")
plt.show()
```




```
from sklearn.tree import DecisionTreeRegressor

model=DecisionTreeRegressor()

model.fit(xtrain,ytrain)
```

```
▾ DecisionTreeRegressor
DecisionTreeRegressor()
```

```
prediction=model.predict(xtest)

from sklearn.tree import plot_tree

plt.figure(figsize=(12,10))
plot_tree(model,fontsize=8,max_depth=3,filled=True)
plt.show()
```



Start coding or generate with AI.