In [56]:
```
import numpy
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
```

In [57]:
```
data = pd.read_csv(r'C:\Users\DELL\Documents\Social_Network_Ads.csv')
data.head()
```

Out[57]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

In [58]:
```
print(data.describe())
```

```
            User ID         Age  EstimatedSalary    Purchased
count  4.000000e+02  400.000000       400.000000  400.000000
mean   1.569154e+07   37.655000     69742.500000    0.357500
std    7.165832e+04   10.482877     34096.960282    0.479864
min    1.556669e+07   18.000000     15000.000000    0.000000
25%    1.562676e+07   29.750000     43000.000000    0.000000
50%    1.569434e+07   37.000000     70000.000000    0.000000
75%    1.575036e+07   46.000000     88000.000000    1.000000
max    1.581524e+07   60.000000    150000.000000    1.000000
```

In [59]:
```
print(data.isnull().sum())
```

```
User ID            0
Gender             0
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

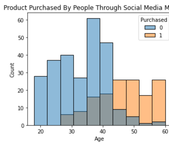
In [60]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   User ID          400 non-null    int64 
 1   Gender           400 non-null    object
 2   Age              400 non-null    int64 
 3   EstimatedSalary  400 non-null    int64 
 4   Purchased        400 non-null    int64 
dtypes: int64(4), object(1)
memory usage: 15.8+ KB
```

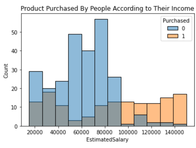
In [61]:
```
data['Purchased'].value_counts()
```

Out[61]:
```
0    257
1    143
Name: Purchased, dtype: int64
```

In [62]:
```
plt.figure(figsize=(5, 4))
plt.title("Product Purchased By People Through Social Media Marketing")
sns.histplot(x="Age", hue="Purchased", data=data)
plt.show()
```

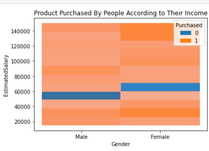


In [63]:
```
plt.title("Product Purchased By People According to Their Income")
sns.histplot(x="EstimatedSalary", hue="Purchased", data=data)
plt.show()
```



In [64]:
```
plt.title("Product Purchased By People According to Their Income")
sns.histplot(x="Age",y="EstimatedSalary",hue="Purchased", data=data)
plt.show()
```

In [65]:
```
plt.title("Product Purchased By People According to Their Income")
sns.histplot(x="Gender",y="EstimatedSalary",hue="Purchased", data=data)
plt.show()
```



In [111]:
```
#feature engineering
X=np.array(data[["Age", "EstimatedSalary"]])
y= np.array(data[["Purchased"]])
#Splitting Data into Train and Test
X_train,X_test, y_train, y_test=train_test_split(X,y,test_size=0.2,random_state=0)
```

In [115]:
```
X_train.shape,X_test.shape,y_train.shape,y_test.shape
```

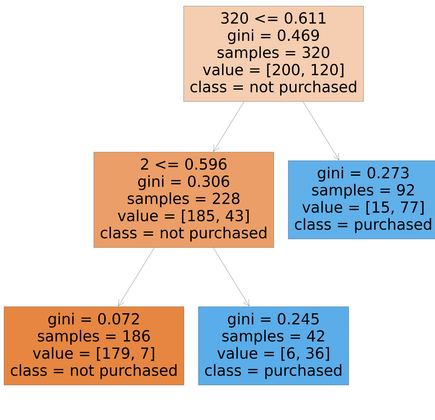
Out[115]:
```
((320, 2), (80, 2), (320, 1), (80, 1))
```

In [116]:
```
#Feature Scaling
from sklearn.preprocessing import StandardScaler
sc_X=StandardScaler()
X_train=sc_X.fit_transform(X_train)
X_test=sc_X.fit_transform(X_test)
```

In [117]:
```
#Model Built
from sklearn.tree import DecisionTreeClassifier
clf=DecisionTreeClassifier(criterion='entropy',random_state=0)
clf.fit(X_train,y_train)
```

Out[117]:
```
DecisionTreeClassifier(criterion='entropy', random_state=0)
```

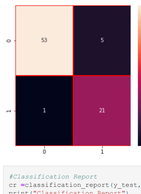
In [149]:
```
#Model Evaluation ---->Plotting Decision Tree
from sklearn import tree
plt.figure(figsize=(50,50))
tree.plot_tree(clf,class_names=['not purchased','purchased'],feature_names=X_train.shape,filled=True)
plt.show()
```



In [119]:
```
#Accuracy Score
from sklearn.metrics import confusion_matrix,classification_report,accuracy_score
y_pred = clf.predict(X_test)
y_train = clf.predict(X_train)
from sklearn.metrics import accuracy_score
print("Train Accuracy is:",accuracy_score(y_train,y_train))
print("Test Accuracy is:",accuracy_score(y_test,y_pred))
```

```
Train Accuracy is: 1.0
Test Accuracy is: 0.925
```

In [84]:
```
#Confusion Matrix
cm=confusion_matrix(y_test,y_pred)
import seaborn as sns
import matplotlib.pyplot as plt
f, ax = plt.subplots(figsize =(5,5))
sns.heatmap(cm,annot = True,linewidths=0.5,linecolor="red",fmt = ".0f",ax=ax)
plt.show()
```



In [85]:
```
#Classification Report
cr =classification_report(y_test,y_pred)
print("Classification Report")
print(cr)
```

```
Classification Report
              precision    recall  f1-score   support

           0       0.98      0.91      0.95        58
           1       0.81      0.95      0.88        22

    accuracy                           0.93        80
   macro avg       0.89      0.93      0.91        80
weighted avg       0.93      0.93      0.93        80
```