In [21]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,classification_report,confusion_mat
from sklearn.linear_model import Perceptron
from matplotlib.colors import ListedColormap
```

In [2]:
```
a=pd.read_csv('suv dataset.csv')
a
```

Out[2]:

	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
...	...	...	...	...	...
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

400 rows × 5 columns

In [3]:
```
a.head()
```

Out[3]:

	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 [4]:
```
a.tail()
```

Out[4]:

	User ID	Gender	Age	EstimatedSalary	Purchased
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

In [5]:
```
a.size
```

Out[5]: 2000

In [8]:
```
X=a[['Age','EstimatedSalary']]
X
```

Out[8]:

	Age	EstimatedSalary
0	19	19000
1	35	20000
2	26	43000
3	27	57000
4	19	76000
...	...	...
395	46	41000
396	51	23000
397	50	20000
398	36	33000
399	49	36000

400 rows × 2 columns

In [9]:
```
Y=a['Purchased']
Y
```

Out[9]:
```
0      0
1      0
2      0
3      0
4      0
      ..
395    1
396    1
397    1
398    0
399    1
Name: Purchased, Length: 400, dtype: int64
```

In [12]:
```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=
model=Perceptron(max_iter=1000,eta0=0.1,random_state=42)
sc=StandardScaler()
X_train=sc.fit_transform(X_train)
X_test=sc.transform(X_test)
model.fit(X_train,Y_train)
y_pred=model.predict(X_test)
```

In [14]:
```
accuracy=accuracy_score(Y_test,y_pred)
accuracy
```

Out[14]:
```
0.7625
```

In [17]:
```
cr=classification_report(Y_test,y_pred)
print('\n Classification Report \n',cr)
```

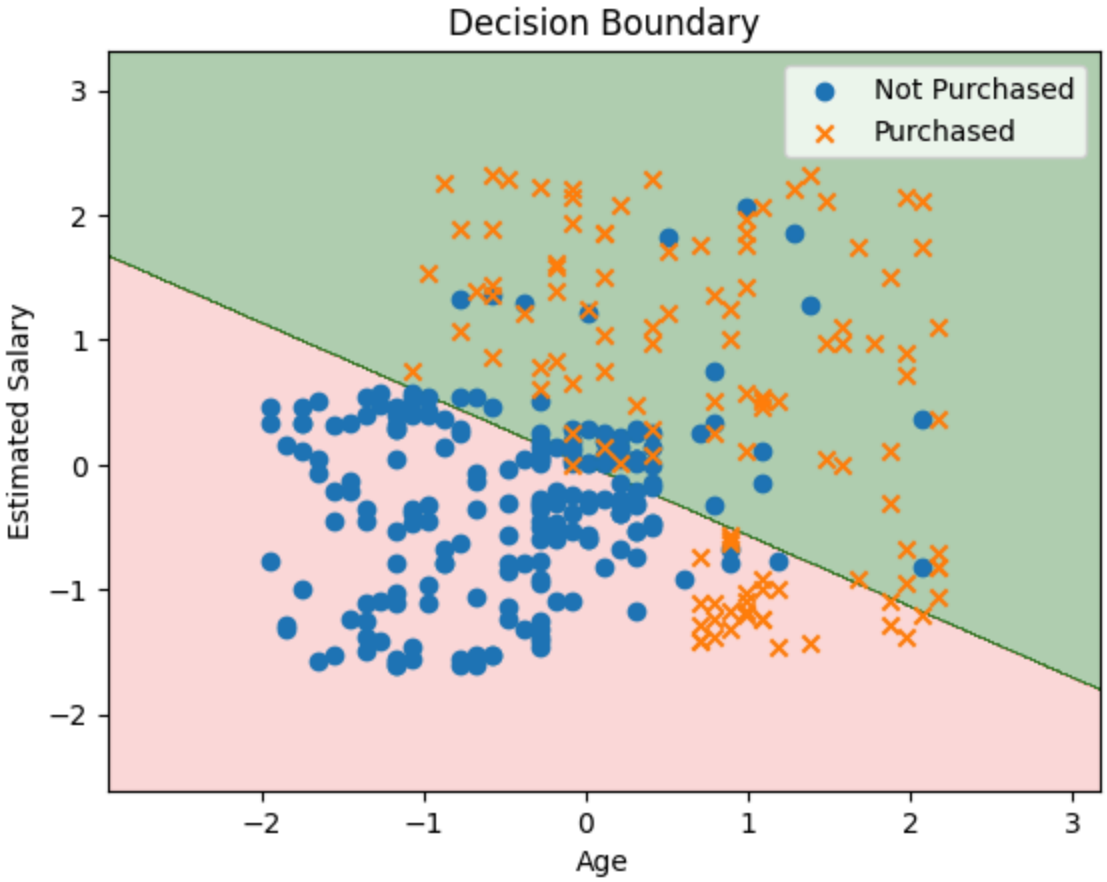
```
Classification Report
              precision    recall  f1-score   support

           0       0.81      0.83      0.82        52
           1       0.67      0.64      0.65        28

    accuracy                           0.76        80
   macro avg       0.74      0.73      0.74        80
weighted avg       0.76      0.76      0.76        80
```

In [25]:
```
X_set,Y_set=X_train,Y_train
X1,X2=np.meshgrid(np.arange(start=X_set[:,0].min()-1,stop=X_set[:,0].max()+1,s
                  np.arange(start=X_set[:,1].min()-1,stop=X_set[:,1].max()+1,s
plt.contourf(X1,X2,model.predict(np.array([X1.ravel(),X2.ravel()]).T).reshape(
             alpha=0.3,cmap=ListedColormap(('lightCoral','darkgreen')))
plt.scatter(X_set[Y_set==0,0],X_set[Y_set==0,1],label='Not Purchased',marker='
plt.scatter(X_set[Y_set==1,0],X_set[Y_set==1,1],label='Purchased',marker='x')
plt.title('Decision Boundary')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
```

Out[25]:
```
<matplotlib.legend.Legend at 0x1e5f1b60550>
```


Decision Boundary
Not Purchased
Purchased
Estimated Salary
Age
3
2
1
0
−1
−2
−2
−1
0
1
2
3

In [ ]: