In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score,confusion_matrix,classification_rep
from sklearn.linear_model import LogisticRegression
```

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.shape
```

Out[5]:
```
(400, 5)
```

In [6]:
```
a.size
```

Out[6]:
```
2000
```

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

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 [19]:
```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.25,random_state
sc=StandardScaler()
X_train=sc.fit_transform(X_train)
X_test=sc.transform(X_test)
model=LogisticRegression()
model.fit(X_train,Y_train)
y_pred=model.predict(X_test)
```

In [12]:
```
cm=confusion_matrix(Y_test,y_pred)
accuracy=accuracy_score(Y_test,y_pred)
cr=classification_report(Y_test,y_pred)
cm
```

Out[12]:
```
array([[61,  2],
       [12, 25]])
```

In [13]:
```
accuracy
```

Out[13]:
```
0.86
```

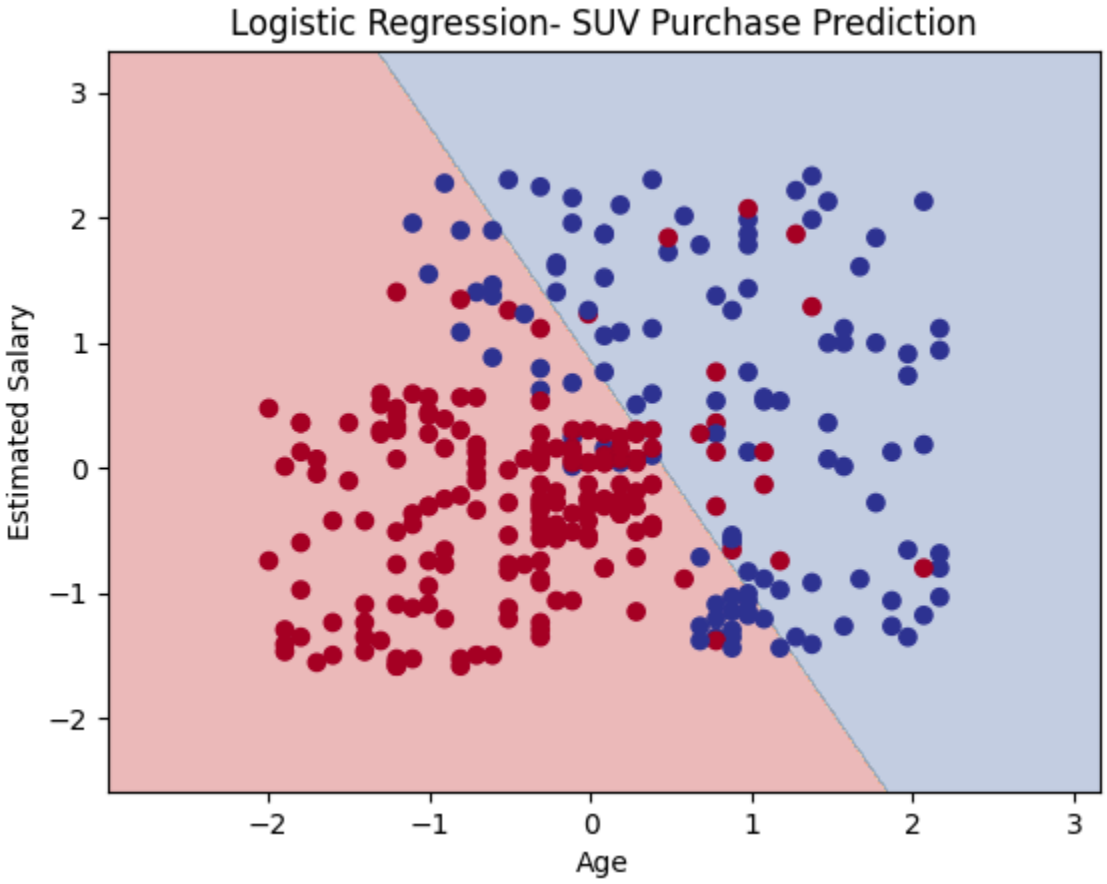
In [15]:
```
print("\n Classification Report\n",cr)
```

```
Classification Report
              precision    recall  f1-score   support

           0       0.84      0.97      0.90        63
           1       0.93      0.68      0.78        37

    accuracy                           0.86       100
   macro avg       0.88      0.82      0.84       100
weighted avg       0.87      0.86      0.85       100
```

In [20]:
```
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=plt.cm.RdYlBu)
plt.scatter(X_set[:,0],X_set[:,1],c=Y_set,cmap=plt.cm.RdYlBu)
plt.xlabel('Age')
```

```
plt.ylabel('Estimated Salary')
plt.title('Logistic Regression- SUV Purchase Prediction')
```

Out[20]: Text(0.5, 1.0, 'Logistic Regression- SUV Purchase Prediction')



In [ ]: