In [571]:

```
# fifth practical Social _Network_Ad
```

In [572]:

```
import pandas as pd
```

In [573]:

```
import numpy as np
```

In [574]:

```
df=pd.read_csv(r'C:\Users\Rutu\Documents\Social_Network_Ads.csv')
```

In [575]:

```
df.head()
```

Out[575]:

	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 [576]:

```
df.tail()
```

Out[576]:

	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 [577]:

```
df.describe()
```

Out[577]:

	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 [578]:
```
df.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 [579]:
```
df.isnull()
```

Out[579]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
395	False	False	False	False	False
396	False	False	False	False	False
397	False	False	False	False	False
398	False	False	False	False	False
399	False	False	False	False	False

400 rows × 5 columns

In [580]:
```
df.isnull().sum()
```

Out[580]:
```
User ID            0
Gender             0
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

```
In [581]: df.dtypes
```

```
Out[581]: User ID              int64
          Gender              object
          Age                  int64
          EstimatedSalary      int64
          Purchased            int64
          dtype: object
```

```
In [582]: df['Gender']=df['Gender'].astype('string')
```

```
In [583]: df.dtypes
```

```
Out[583]: User ID              int64
          Gender              string
          Age                  int64
          EstimatedSalary      int64
          Purchased            int64
          dtype: object
```

```
In [584]: df.shape
```

```
Out[584]: (400, 5)
```

```
In [585]: df.columns
```

```
Out[585]: Index(['User ID', 'Gender', 'Age', 'EstimatedSalary', 'Purchased'], dtype='obje
          ct')
```

```
In [586]: #from sklearn.model_selection import train_test_split
```

```
In [587]: #x_train,x_test,y_train,y_test=train_test_split()
```

```
In [588]: from sklearn.datasets import make_classification
          x,y =make_classification(n_samples=400,n_features=5,n_clusters_per_class=1,n_clas
```

```
In [589]: x[0:4]
```

```
Out[589]: array([[ 0.73482358, -0.68424645, -0.20910637,  1.93598577,  0.11280744],
                 [ 0.65289568,  0.34365565, -0.45591234,  0.73078877,  1.36483182],
                 [ 0.10624064, -0.39197574, -0.26471719,  1.54908018,  0.4318939 ],
                 [ 0.17565544,  1.96341786, -0.79799196, -1.31055739,  3.21907587]])
```

```
In [590]: y[0:4]
```

```
Out[590]: array([0, 0, 0, 0])
```

```
In [591]: x.shape,y.shape
```

```
Out[591]: ((400, 5), (400,))
```

```
In [592]: from sklearn.model_selection import train_test_split
```

```
In [593]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

```
In [594]: from sklearn.linear_model import LogisticRegression
```

```
In [595]: model=LogisticRegression()
```

```
In [596]: model.fit(x_train,y_train)

Out[596]: LogisticRegression()
```

```
In [597]: y_pred=model.predict(x_test)
```

```
In [598]: y_pred.shape

Out[598]: (80,)
```

```
In [599]: y_pred

Out[599]: array([1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1,
                 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0,
                 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0,
                 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1])
```

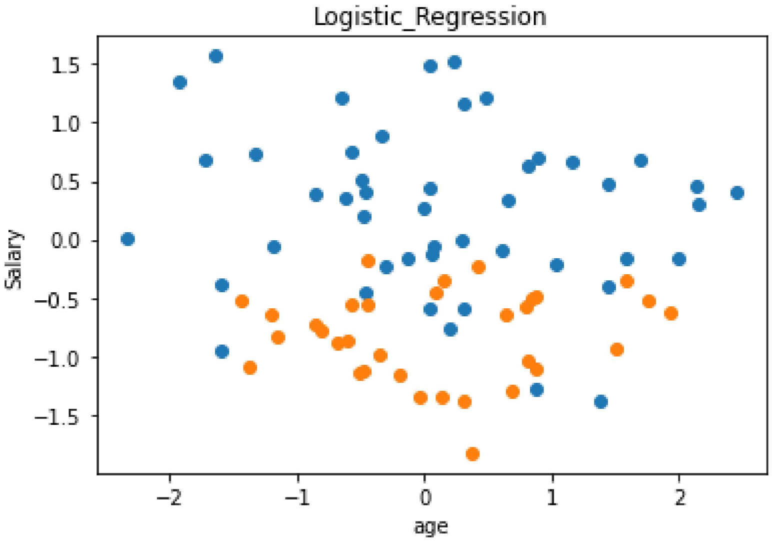
```
In [600]: from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
```

```
In [601]: accuracy_score(y_test,y_pred)

Out[601]: 0.85
```

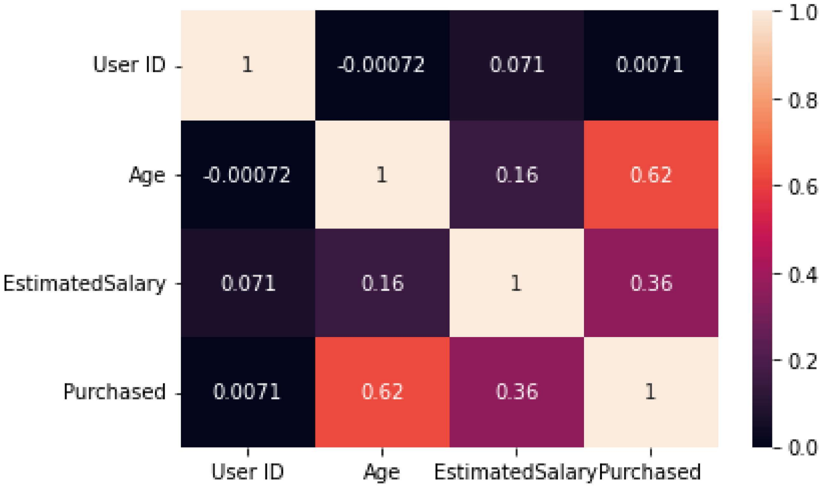
```
In [602]: confusion_matrix(y_test,y_pred)#[[tp,fp] ,[fn,tn]]

Out[602]: array([[43,  4],
                 [ 8, 25]], dtype=int64)
```

```
In [603]: print(classification_report(y_test,y_pred))

                        precision    recall  f1-score   support

                     0       0.84      0.91      0.88        47
                     1       0.86      0.76      0.81        33

              accuracy                           0.85        80
             macro avg       0.85      0.84      0.84        80
          weighted avg       0.85      0.85      0.85        80
```

In [604]:
```
for i, j in enumerate(np.unique(y_test)):
    plt.scatter(x_test[y_test==j,0],x_test[y_test==j,1])
plt.title("Logistic_Regression")
plt.xlabel("age")
plt.ylabel("Salary")
plt.show()
```



In [607]:
```
import matplotlib.pyplot as plt
import seaborn as sns
#plt.figure(figsize=(10,5))
sns.heatmap(df.corr(),annot=True)
plt.show()
```



In [623]:
```
from sklearn.metrics import precision_score,recall_score,f1_score
```

In [614]:
```
recall_score(y_test,y_pred)
```

Out[614]:
```
0.7575757575757576
```

In [613]:

```
precision_score(y_test,y_pred)
```

Out[613]:

```
0.8620689655172413
```

In [617]:

```
f1_score(y_test,y_pred)
```

Out[617]:

```
0.8064516129032258
```

In [ ]: