In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

In [2]:
```
df=pd.read_csv("/home/rmdstic/Documents/TEA-14/Social_Network_Ads.csv
df
```

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
5	15728773	Male	27	58000	0
6	15598044	Female	27	84000	0
7	15694829	Female	32	150000	1
8	15600575	Male	25	33000	0
9	15727311	Female	35	65000	0
10	15570769	Female	26	80000	0
11	15606274	Female	26	52000	0
12	15746139	Male	20	86000	0
13	15704987	Male	32	18000	0
14	15628972	Male	18	82000	0
15	15697686	Male	29	80000	0
16	15733883	Male	47	25000	1
17	15617482	Male	45	26000	1
18	15704583	Male	46	28000	1
19	15621083	Female	48	29000	1
20	15649487	Male	45	22000	1
21	15736760	Female	47	49000	1
22	15714658	Male	48	41000	1
23	15599081	Female	45	22000	1
24	15705113	Male	46	23000	1
25	15631159	Male	47	20000	1
26	15792818	Male	49	28000	1
27	15633531	Female	47	30000	1
28	15744529	Male	29	43000	0
29	15669656	Male	31	18000	0
...	...	...	...	...	...
370	15611430	Female	60	46000	1
371	15774744	Male	60	83000	1
372	15629885	Female	39	73000	0

	User ID	Gender	Age	EstimatedSalary	Purchased
373	15708791	Male	59	130000	1
374	15793890	Female	37	80000	0
375	15646091	Female	46	32000	1
376	15596984	Female	46	74000	0
377	15800215	Female	42	53000	0
378	15577806	Male	41	87000	1
379	15749381	Female	58	23000	1
380	15683758	Male	42	64000	0
381	15670615	Male	48	33000	1
382	15715622	Female	44	139000	1
383	15707634	Male	49	28000	1
384	15806901	Female	57	33000	1
385	15775335	Male	56	60000	1
386	15724150	Female	49	39000	1
387	15627220	Male	39	71000	0
388	15672330	Male	47	34000	1
389	15668521	Female	48	35000	1
390	15807837	Male	48	33000	1
391	15592570	Male	47	23000	1
392	15748589	Female	45	45000	1
393	15635893	Male	60	42000	1
394	15757632	Female	39	59000	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]:
```
df.drop("Gender",axis=1,inplace=True)
df.head()
```

Out[3]:

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

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
User ID            0
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

In [5]:
```
x=df.drop("Purchased",axis=1)
x.head()
```

Out[5]:

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

In [6]:
```
y=df.Purchased
y.head()
```

Out[6]:
```
0    0
1    0
2    0
3    0
4    0
Name: Purchased, dtype: int64
```

In [7]:
```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest =train_test_split(x, y, test_size =0.2,
```

In [8]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
xtrain = sc.fit_transform(xtrain)
xtest = sc.transform(xtest)
xtest
```

```
/home/rmdstic/anaconda3/lib/python3.7/site-packages/sklearn/preproc
essing/data.py:645: DataConversionWarning: Data with input dtype in
t64 were all converted to float64 by StandardScaler.
  return self.partial_fit(X, y)
/home/rmdstic/anaconda3/lib/python3.7/site-packages/sklearn/base.p
y:464: DataConversionWarning: Data with input dtype int64 were all
converted to float64 by StandardScaler.
  return self.fit(X, **fit_params).transform(X)
/home/rmdstic/anaconda3/lib/python3.7/site-packages/ipykernel_launc
her.py:4: DataConversionWarning: Data with input dtype int64 were a
ll converted to float64 by StandardScaler.
  after removing the cwd from sys.path.
```

Out[8]:
```
array([[ 4.88966473e-01, -7.98950822e-01,  4.94607583e-01],
       [-5.50592839e-01, -2.12648508e-02, -5.77359062e-01],
       [ 1.21526737e+00, -3.12897090e-01,  1.46942725e-01],
       [ 6.70428177e-01, -7.98950822e-01,  2.62831011e-01],
       [-7.54886252e-01, -3.12897090e-01, -5.77359062e-01],
       [ 1.74256773e+00, -1.09058306e+00, -1.44652121e+00],
       [-1.15026219e+00, -7.01740076e-01, -1.59138156e+00],
       [-1.50619343e-01, -2.15686344e-01,  2.14601566e+00],
       [-3.55608066e-01, -1.96547978e+00, -5.58617754e-02],
       [-9.75185983e-01,  8.53631867e-01, -7.80163563e-01],
       [-2.52081338e-02, -7.98950822e-01, -6.06331134e-01],
       [ 1.54471657e+00, -9.93372315e-01, -4.32498705e-01],
       [ 1.16706395e+00, -1.18475597e-01, -4.32498705e-01],
       [-1.58913047e+00,  7.59458956e-02,  2.04886868e-01],
       [ 7.79563458e-01, -1.77105829e+00,  4.65635512e-01],
       [ 1.38930771e+00, -6.04529329e-01,  1.36376973e+00],
       [ 9.84339331e-01, -1.18475597e-01,  2.04886868e-01],
       [ 1.65607968e+00, -1.86826903e+00,  4.36663440e-01],
       [-5.24625142e-01,  1.63131784e+00,  1.74040666e+00],
       [ 2.41606414e-01, -3.12897090e-01, -1.38857706e+00],
       [-1.15717272e+00, -3.12897090e-01, -6.64275277e-01],
       [ 7.80471618e-01,  8.53631867e-01,  2.14601566e+00],
       [-1.84944950e-01,  2.70367388e-01, -5.48386991e-01],
       [ 5.49245588e-01,  8.53631867e-01,  1.01610487e+00],
       [-1.08952899e+00, -1.47942605e+00, -1.21474464e+00],
       [ 8.89138629e-01,  1.04805336e+00,  2.05909944e+00],
       [-3.39658507e-01, -9.93372315e-01,  4.94607583e-01],
       [ 9.20154463e-02, -8.96161568e-01,  2.91803083e-01],
       [-8.53265514e-01, -1.18475597e-01, -2.29694204e-01],
       [-1.51345521e+00, -6.04529329e-01,  4.65635512e-01],
       [-1.47843429e+00, -1.67384754e+00,  5.23579655e-01],
       [-7.60704152e-01, -1.18475597e-01,  2.62831011e-01],
       [ 1.75035804e+00,  1.82573933e+00, -2.87638347e-01],
       [-9.79542313e-01, -1.18475597e-01, -4.90442848e-01],
       [ 3.81278572e-01, -1.38221530e+00, -3.45582490e-01],
       [ 7.88049077e-01, -1.96547978e+00, -5.19414919e-01],
       [-9.08226882e-02, -1.57663679e+00,  3.20775154e-01],
       [-1.55447849e+00, -4.10107836e-01, -7.80163563e-01],
       [-2.53071135e-01, -7.01740076e-01, -1.04091221e+00],
       [-1.37870698e+00,  1.04805336e+00, -9.82968063e-01],
       [ 1.06549193e+00, -1.09058306e+00,  5.23579655e-01],
```

```
       [-2.39391976e-01,  2.70367388e-01, -5.19414919e-01],
       [-1.32190441e+00, -1.09058306e+00,  4.07691369e-01],
       [ 1.69914633e+00, -3.12897090e-01, -1.44652121e+00],
       [ 2.00029717e-01,  4.64788881e-01,  1.21890937e+00],
       [-5.15103652e-01, -1.09058306e+00, -3.45582490e-01],
       [ 1.45715009e+00, -1.18475597e-01,  2.91803083e-01],
       [ 8.35145683e-01,  1.33968560e+00,  5.81523798e-01],
       [ 1.01321598e+00, -1.18779381e+00, -1.15680049e+00],
       [-1.09283526e+00,  1.04805336e+00,  4.65635512e-01],
       [ 1.19747311e+00,  1.82573933e+00,  1.50863009e+00],
       [-9.70148534e-01, -4.10107836e-01, -1.30166085e+00],
       [-1.76683183e+00, -3.12897090e-01, -3.74554562e-01],
       [-1.70662366e+00, -4.10107836e-01,  1.30582558e+00],
       [-1.15694568e+00,  2.02016082e+00,  5.23579655e-01],
       [ 7.23725813e-01,  6.59210374e-01, -1.09885635e+00],
       [ 7.62251660e-01, -8.96161568e-01,  3.78719297e-01],
       [-1.70893663e+00, -1.18779381e+00,  2.91803083e-01],
       [ 1.44193841e+00,  1.04805336e+00, -1.21474464e+00],
       [-7.36808194e-01, -1.47942605e+00, -1.44652121e+00],
       [-1.39522414e+00, -6.04529329e-01, -1.50446535e+00],
       [ 1.42143386e+00,  2.11737157e+00, -8.09135634e-01],
       [ 1.60063936e+00, -1.86826903e+00,  1.75914797e-01],
       [ 1.19444714e-01, -2.15686344e-01,  8.42272441e-01],
       [-4.13418121e-01, -1.86826903e+00, -1.27268878e+00],
       [ 1.18546837e+00,  2.11737157e+00,  3.78719297e-01],
       [ 7.83579228e-01, -1.38221530e+00,  5.52551726e-01],
       [ 5.33139939e-01, -1.09058306e+00, -3.45582490e-01],
       [-1.21855865e+00,  1.73156642e-01, -6.64275277e-01],
       [ 3.78752752e-01,  3.67578135e-01,  2.08236764e-03],
       [ 5.14746197e-02, -6.04529329e-01,  2.31984809e+00],
       [ 1.70245260e+00, -3.12897090e-01,  2.04886868e-01],
       [-1.61799293e+00, -1.57663679e+00, -2.00722133e-01],
       [-5.91928306e-01,  6.59210374e-01, -1.38857706e+00],
       [ 1.40000697e+00, -1.09058306e+00,  5.52551726e-01],
       [-8.83036131e-01, -1.96547978e+00,  3.49747226e-01],
       [ 1.09392869e+00,  3.67578135e-01,  2.62831011e-01],
       [-1.32153547e+00,  1.73156642e-01, -2.87638347e-01],
       [ 1.24344871e+00,  1.43689635e+00, -1.04091221e+00],
       [-7.32565384e-01,  8.53631867e-01,  1.07404901e+00]])
```

In [9]:
```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(xtrain, ytrain)
```

```
/home/rmdstic/anaconda3/lib/python3.7/site-packages/sklearn/linear_
model/logistic.py:433: FutureWarning: Default solver will be change
d to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
```

Out[9]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_interc
ept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='war
n',
          tol=0.0001, verbose=0, warm_start=False)
```

In [10]:
```
ytrain_pred = lr.predict(xtrain)
ytest_pred = lr.predict(xtest)
```

In [11]:
```
ytrain_pred
```

Out[11]:
```
array([1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1,
0, 0,
       0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1,
1, 1,
       0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1,
0, 1,
       0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0,
1, 1,
       0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0,
0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0,
1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1,
       0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0,
1, 0,
       1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0,
0, 0,
       1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0,
0, 0,
       0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0,
1, 0,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0,
1, 0,
       0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
0, 1,
       1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

In [12]:
```
ytest_pred
```

Out[12]:
```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
0, 1,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
0, 0,
       1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
0, 1,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1])
```

In [13]:
```
print(lr.predict( [[15624510,19,19000]]))
```

```
[0]
```

In [14]:
```
lr.predict([[4.88966473e-01,-7.98950822e-01,4.94607583e-01]])
```

Out[14]:
```
array([0])
```

In [15]:
```
lr.predict([[-1.50619343e-01,-2.15686344e-01,2.14601566e+00]])
```

Out[15]:
```
array([1])
```

In [17]:
```
from sklearn.metrics import confusion_matrix,classification_report,a
matrix = confusion_matrix(ytest,ytest_pred)
print(matrix)
```

```
[[56  2]
 [ 5 17]]
```

In [18]:
```
score=accuracy_score(ytest,ytest_pred)
score
```

Out[18]:
```
0.9125
```

In [19]:
```
cr=classification_report(ytest,ytest_pred)
print(cr)
```

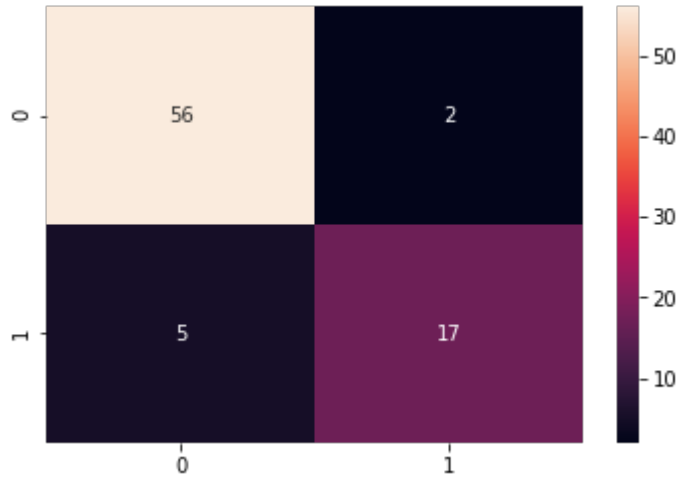
```
              precision    recall  f1-score   support

           0       0.92      0.97      0.94        58
           1       0.89      0.77      0.83        22

   micro avg       0.91      0.91      0.91        80
   macro avg       0.91      0.87      0.89        80
weighted avg       0.91      0.91      0.91        80
```

In [20]:
```
pip install seaborn
```

```
Requirement already satisfied: seaborn in /home/rmdstic/anaconda3/l
ib/python3.7/site-packages (0.9.0)
Requirement already satisfied: matplotlib>=1.4.3 in /home/rmdstic/a
naconda3/lib/python3.7/site-packages (from seaborn) (3.0.3)
Requirement already satisfied: pandas>=0.15.2 in /home/rmdstic/anac
onda3/lib/python3.7/site-packages (from seaborn) (0.24.2)
Requirement already satisfied: numpy>=1.9.3 in /home/rmdstic/anacon
da3/lib/python3.7/site-packages (from seaborn) (1.16.2)
Requirement already satisfied: scipy>=0.14.0 in /home/rmdstic/anaco
nda3/lib/python3.7/site-packages (from seaborn) (1.2.1)
Requirement already satisfied: cycler>=0.10 in /home/rmdstic/anacon
da3/lib/python3.7/site-packages (from matplotlib>=1.4.3->seaborn)
(0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /home/rmdstic/a
naconda3/lib/python3.7/site-packages (from matplotlib>=1.4.3->seabo
rn) (1.0.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=
2.0.1 in /home/rmdstic/anaconda3/lib/python3.7/site-packages (from
matplotlib>=1.4.3->seaborn) (2.3.1)
Requirement already satisfied: python-dateutil>=2.1 in /home/rmdsti
c/anaconda3/lib/python3.7/site-packages (from matplotlib>=1.4.3->se
aborn) (2.8.0)
Requirement already satisfied: pytz>=2011k in /home/rmdstic/anacond
a3/lib/python3.7/site-packages (from pandas>=0.15.2->seaborn) (201
8.9)
Requirement already satisfied: six in /home/rmdstic/anaconda3/lib/p
ython3.7/site-packages (from cycler>=0.10->matplotlib>=1.4.3->seabo
rn) (1.12.0)
Requirement already satisfied: setuptools in /home/rmdstic/anaconda
3/lib/python3.7/site-packages (from kiwisolver>=1.0.1->matplotlib>=
1.4.3->seaborn) (40.8.0)
Note: you may need to restart the kernel to use updated packages.
```

In [21]:
```
import seaborn as sns
sns.heatmap(matrix,annot=True)
```

Out[21]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0e805f3ba8>
```



In [22]:
```
tn, fp, fn, tp = confusion_matrix(ytest,ytest_pred).ravel()
```

In [23]:
```
print(tn, fp, fn, tp)
```

```
56 2 5 17
```

In [24]:
```
print("Error rate:",(fp+fn)/(tn+fp+fn+tp))
```

```
Error rate: 0.0875
```

In [ ]:
```

```