Random_Forest_Assignment16

Importing the libraries

In [1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set(color_codes=True)
from sklearn.preprocessing import LabelEncoder
```

Importing the dataset

In [2]:
```python
dataset = pd.read_csv("Social_Network_Ads.csv")
```

In [3]:
```python
dataset.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]:
```python
dataset.shape
```

Out[4]:
```
(400, 5)
```

In [5]:
```python
dataset.columns
```

Out[5]:
```
Index(['User ID', 'Gender', 'Age', 'EstimatedSalary', 'Purchased'], dtype='object')
```

In [6]:
```python
dataset.isna().any()
```

Out[6]:
```
User ID            False
Gender             False
Age                False
EstimatedSalary    False
Purchased          False
dtype: bool
```

In [8]:
```python
dataset.isna().sum()
```

Out[8]:
```
User ID            0
Gender             0
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

In [10]:
```python
from sklearn.preprocessing import LabelEncoder
```

In [11]:
```python
Gender=LabelEncoder()
```

In [12]:
```python
dataset['Gender']=Gender.fit_transform(dataset['Gender'])
```

In [13]:
```python
dataset
```

Out[13]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	1	19	19000	0
1	15810944	1	35	20000	0
2	15668575	0	26	43000	0
3	15603246	0	27	57000	0
4	15804002	1	19	76000	0
...	...	...	...	...	...
395	15691863	0	46	41000	1
396	15706071	1	51	23000	1
397	15654296	0	50	20000	1
398	15755018	1	36	33000	0
399	15594041	0	49	36000	1

400 rows × 5 columns

In [14]:
```python
X=dataset[['Gender','Age','EstimatedSalary']]
y=dataset['Purchased']
```

In [15]:
```python
X
```

Out[15]:

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

400 rows × 3 columns

In [16]:
```python
y
```

Out[16]:
```
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
```

Splitting the dataset into the Training set and Test set

In [17]:
```python
from sklearn.model_selection import train_test_split
```

In [18]:
```python
 X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3)
```

In [19]:
```python
len(X_train)
```

Out[19]:
```
280
```

In [20]:
```python
len(X_test)
```

Out[20]:
```
120
```

Feature Scaling

In [21]:
```python
from sklearn.preprocessing import StandardScaler
```

In [22]:
```python
sc= StandardScaler()
```

In [23]:
```python
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Fitting Random Forest to the Training set

In [24]:
```python
from sklearn.ensemble import RandomForestClassifier
```

In [25]:
```python
clf=RandomForestClassifier(n_estimators=100,criterion='gini',random_state=0)
```

In [26]:
```python
clf.fit(X_train,y_train)
```

Out[26]:
```
RandomForestClassifier(random_state=0)
```

Predicting the Test set results

In [27]:
```python
clf.predict(X_test)
```

Out[27]:
```
array([0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0,
       0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1,
       1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 0, 1, 0, 0], dtype=int64)
```

In [28]:
```python
X_test
```

Out[28]:
```
array([[-1.07417231,  0.13230684,  0.14773116],
       [ 0.93094934, -1.84222166,  0.41887648],
       [-1.07417231,  0.79048301, -1.38875897],
       [ 0.93094934,  1.82475984, -0.24392319],
       [-1.07417231, -1.65417132, -1.35863172],
       [-1.07417231,  1.73073467,  1.08167614],
       [-1.07417231,  0.88450817, -0.63557754],
       [ 0.93094934,  0.41438234,  0.35862196],
       [-1.07417231, -1.84222166,  0.53938551],
       [-1.07417231, -0.33781899,  0.1176039 ],
       [-1.07417231,  0.32035717, -0.51506851],
       [-1.07417231,  0.22633201,  0.1176039 ],
       [ 0.93094934, -1.37209582, -1.44901349],
       [-1.07417231,  1.16658367, -0.96697737],
       [-1.07417231,  0.97853334,  1.98549387],
       [-1.07417231, -1.27807066, -1.0874864 ],
       [-1.07417231, -0.61989449, -0.00290513],
       [ 0.93094934,  0.22633201, -0.27405045],
       [ 0.93094934,  2.01281017,  1.86498484],
       [ 0.93094934,  0.03828167,  0.08747664],
       [-1.07417231,  0.13230684, -0.21379593],
       [ 0.93094934,  0.50840751,  1.83485758],
       [ 0.93094934, -0.99599516, -0.3041777 ],
       [ 0.93094934, -0.71391966,  0.56951276],
       [-1.07417231, -0.43184416,  1.35282146],
       [ 0.93094934, -0.24379382,  0.14773116],
       [-1.07417231,  0.79048301,  0.84065808],
       [ 0.93094934,  0.88450817, -1.2983772 ],
       [-1.07417231,  1.44865917, -1.02723188],
       [-1.07417231, -0.90196999, -0.72595931],
       [ 0.93094934, -1.27807066, -0.39455948],
       [ 0.93094934,  0.69645784, -1.38875897],
       [-1.07417231,  0.32035717,  0.1176039 ],
       [ 0.93094934, -0.24379382, -0.90672285],
       [ 0.93094934,  1.44865917,  0.1176039 ],
       [ 0.93094934, -0.24379382,  1.20218517],
       [ 0.93094934, -1.09002032,  0.38874922],
       [ 0.93094934,  0.32035717, -0.27405045],
       [ 0.93094934, -0.14976866,  2.28676644],
       [-1.07417231,  1.26060884,  2.34702096],
       [ 0.93094934,  0.69645784,  0.3284947 ],
       [-1.07417231,  1.91878501,  0.99129437],
       [ 0.93094934,  1.73073467,  1.95536661],
       [-1.07417231, -0.43184416, -0.51506851],
       [-1.07417231, -1.27807066, -1.23812269],
       [-1.07417231,  1.35463401,  2.1060029 ],
       [-1.07417231,  0.32035717,  0.35862196],
       [-1.07417231,  0.69645784,  1.89511209],
       [ 0.93094934, -1.18404549,  0.65989454],
       [-1.07417231,  1.35463401,  1.38294872],
       [-1.07417231,  0.13230684,  1.98549387],
       [ 0.93094934, -1.09002032, -1.14774091],
       [-1.07417231,  0.88450817,  1.17205791],
       [ 0.93094934,  0.88450817, -0.75608657],
       [-1.07417231,  0.22633201, -0.24392319],
       [ 0.93094934,  0.22633201, -0.63557754],
       [ 0.93094934, -1.27807066, -1.47914075],
       [-1.07417231,  1.07255851,  2.19638467],
       [ 0.93094934,  0.13230684, -0.78621382],
       [-1.07417231,  0.79048301,  0.41887648],
       [ 0.93094934,  0.97853334, -1.05735914],
       [ 0.93094934,  0.03828167, -0.21379593],
       [ 0.93094934,  0.03828167, -0.21379593],
       [-1.07417231, -0.90196999, -0.39455948],
       [-1.07417231,  0.03828167,  0.35862196],
       [-1.07417231,  1.26060884,  1.98549387],
       [ 0.93094934,  2.10683534, -1.02723188],
       [ 0.93094934, -0.52586932,  1.56371226],
       [ 0.93094934,  0.13230684,  1.14193066],
       [-1.07417231, -1.09002032,  0.47913099],
       [-1.07417231, -0.52586932,  2.46752999],
       [ 0.93094934, -1.84222166, -0.48494125],
       [ 0.93094934, -0.14976866,  1.71434855],
       [ 0.93094934, -0.33781899,  1.3226942 ],
       [-1.07417231, -0.05574349,  0.29836745],
       [-1.07417231, -0.71391966,  1.44320323],
       [ 0.93094934,  1.82475984,  0.17785842],
       [-1.07417231,  0.03828167, -0.54519576],
       [-1.07417231, -1.46612099, -0.39455948],
       [-1.07417231, -1.09002032, -1.59964977],
       [-1.07417231,  0.22633201,  0.20798567],
       [-1.07417231,  0.69645784, -1.38875897],
       [ 0.93094934,  0.41438234, -0.42468673],
       [ 0.93094934,  1.26060884, -1.35863172],
       [-1.07417231, -0.05574349,  2.07587564],
       [-1.07417231, -0.52586932,  1.47333049],
       [ 0.93094934, -0.43184416, -1.11761366],
       [-1.07417231, -1.27807066, -0.39455948],
       [-1.07417231,  0.13230684,  0.20798567],
       [-1.07417231, -0.24379382, -0.33430496],
       [ 0.93094934,  1.07255851, -1.20799543],
       [-1.07417231,  0.13230684, -0.27405045],
       [-1.07417231, -0.24379382,  2.37714821],
       [-1.07417231, -0.24379382,  0.26824019],
       [-1.07417231, -0.80794482, -0.63557754],
       [-1.07417231, -0.80794482,  0.44900373],
       [-1.07417231, -0.61989449, -1.02723188],
       [-1.07417231, -0.24379382, -0.54519576],
       [-1.07417231, -0.90196999, -0.27405045],
       [-1.07417231,  0.79048301,  0.17785842],
       [ 0.93094934, -0.05574349,  0.1176039 ],
       [-1.07417231, -0.05574349, -1.05735914],
       [-1.07417231, -0.99599516, -1.53939526],
       [-1.07417231, -1.09002032,  0.35862196],
       [-1.07417231,  2.01281017, -0.78621382],
       [-1.07417231, -1.09002032,  0.1176039 ],
       [ 0.93094934,  1.07255851,  0.59964002],
       [-1.07417231, -0.24379382,  0.08747664],
       [-1.07417231, -0.43184416,  2.43740273],
       [-1.07417231,  0.32035717, -0.24392319],
       [-1.07417231,  0.13230684,  0.08747664],
       [-1.07417231,  0.88450817,  2.28676644],
       [-1.07417231, -1.56014616, -0.00290513],
       [-1.07417231, -0.71391966, -0.18366867],
       [ 0.93094934, -0.05574349, -0.48494125],
       [-1.07417231,  0.03828167, -0.39455948],
       [-1.07417231, -0.61989449,  0.08747664],
       [-1.07417231,  0.41438234,  1.20218517],
       [ 0.93094934, -0.80794482,  0.35862196],
       [-1.07417231, -1.37209582, -0.60545028]])
```

In [29]:
```python
clf.score(X_test,y_test)
```

Out[29]:
```
0.8583333333333333
```

Making the Confusion Matrix

In [30]:
```python
y_pred = clf.predict(X_test)
```

In [31]:
```python
from sklearn.metrics import confusion_matrix
```

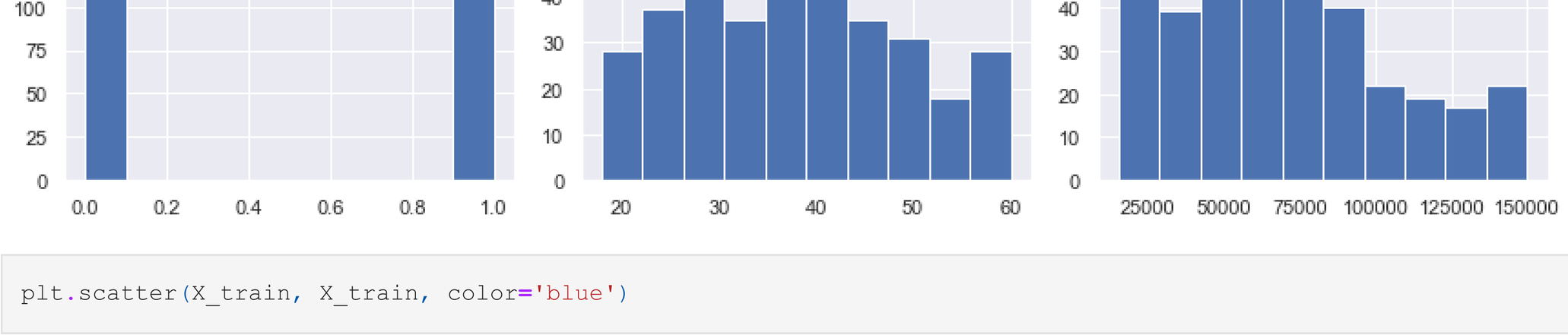
In [32]:
```python
cm = confusion_matrix(y_test, y_pred)
```

In [33]:
```python
print(cm)
```

```
[[68 13]
 [ 4 35]]
```

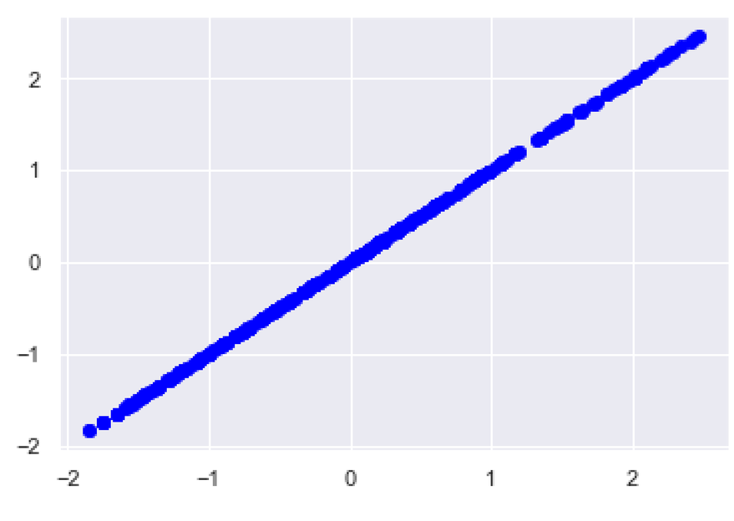
Visualising the Training set results

In [40]:
```python
to_hist = ['Gender','Age','EstimatedSalary']
plt.figure(figsize=(16,10))
for idx, col in enumerate(to_hist):
    plt.subplot(3, 4, idx+1)
    plt.hist(dataset[col])
    plt.title(col)
plt.tight_layout();
```



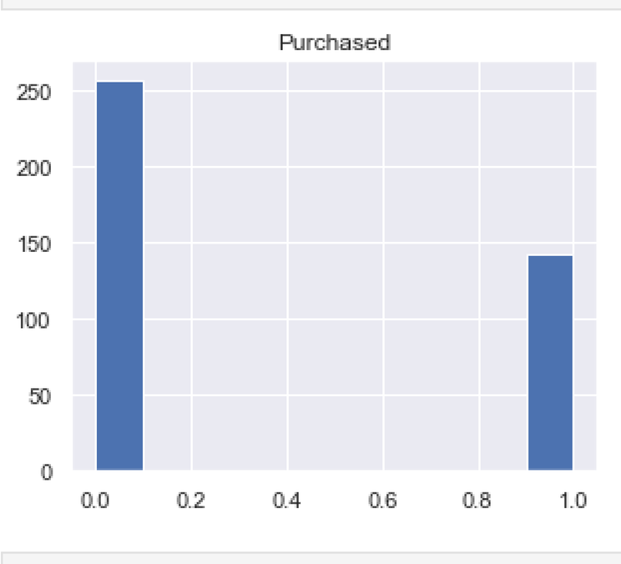
In [47]:
```python
plt.scatter(X_train, X_train, color='blue')
```

Out[47]:
```
<matplotlib.collections.PathCollection at 0x212e5f30ee0>
```



Visualising the Test set results

In [48]:
```python
to_hist = ['Purchased']
plt.figure(figsize=(16,10))
for idx, col in enumerate(to_hist):
    plt.subplot(3, 4, idx+1)
    plt.hist(dataset[col])
    plt.title(col)
plt.tight_layout();
```



In [50]:
```python
plt.scatter(X_test, X_test, color='blue')
```

Out[50]:
```
<matplotlib.collections.PathCollection at 0x212e5b12ee0>
```