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

In [2]:
```
# Importing the dataset
dataset = pd.read_csv('Social_Network_Ads.csv')
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values
```

In [3]:
```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state =0)
```

In [4]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [5]:
```
# Fitting Random Forest Classification to the Training set
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 50)
classifier.fit(X_train, y_train)
```

Out[5]:
```
RandomForestClassifier(criterion='entropy', n_estimators=10, random_state=50)
```

Random Search CV

In [6]:
```
from sklearn.model_selection import RandomizedSearchCV
```

In [7]:
```
est = RandomForestClassifier(n_jobs=-1)
```

In [9]:
```
from scipy.stats import randint
```

In [10]:
```
rf_p_dist={'max_depth':[3,5,10,None],
              'n_estimators':[10,100,200,300,400,500],
              'max_features':randint(1,3),
               'criterion':['gini','entropy'],
               'bootstrap':[True,False],
               'min_samples_leaf':randint(1,4),
              }
```

In [15]:
```
def hypertuning_rscv(est, p_distr, nbr_iter,X,y):
    rdmsearch = RandomizedSearchCV(est, param_distributions=p_distr,
                                   n_jobs=-1, n_iter=nbr_iter, cv=9)
    rdmsearch.fit(X,y)
    ht_params = rdmsearch.best_params_
    ht_score = rdmsearch.best_score_
    return ht_params, ht_score
```

In [16]:
```
rf_parameters, rf_ht_score = hypertuning_rscv(est, rf_p_dist, 10, X, y)
```

In [17]:
```
rf_parameters
```

Out[17]:
```
{'bootstrap': True,
 'criterion': 'entropy',
 'max_depth': 3,
 'max_features': 2,
 'min_samples_leaf': 1,
 'n_estimators': 500}
```

In [18]:
```
rf_ht_score
```

Out[18]:
```
0.9120089786756453
```

After getting which is best suited

In [19]:
```
claasifier=RandomForestClassifier(n_jobs=-1, n_estimators=300,bootstrap= True,criterion='entropy',max_depth=3,max_features=2,min_samples_leaf= 3)
```

Accuracy

In [20]:
```
y_pred = classifier.predict(X_test)
```

In [21]:
```
from sklearn.metrics import confusion_matrix,accuracy_score
cm = confusion_matrix(y_test, y_pred)

accuracy_score=accuracy_score(y_test,y_pred)
```

In [22]:
```
from sklearn.model_selection import cross_val_score

cross_val=cross_val_score(claasifier,X,y,cv=10,scoring='accuracy').mean()
```

In [23]:
```
accuracy_score
```

Out[23]:
```
0.92
```