In [1]:

```
# RandomForest Model ===> Supervised ML Model

# 1).  R.F. ===> '100 DECISION TREE MODEL'

# Target Data Categorical  ===> R.F.Classifier()  ===> 100 dt  ===> 70 yes , 30 no ===>
# majority ==> final prediction (yes)

# Target Data Numerical ===> R.F.Regressor() ===> mean > value ===> final prediction
```

In [2]:

```
import numpy as np
import pandas as pd
```

In [3]:

```
df = pd.read_csv("D:\Summer Training Video\ML\Social_Network_Ads.csv")
```

In [4]:

```
df.head()
```

Out[4]:

	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 [6]:

```
from sklearn.preprocessing import LabelEncoder
```

In [8]:

```
lb = LabelEncoder()
```

In [9]:

```
df['Gender'] = lb.fit_transform(df['Gender'])
```

In [10]:

```
df.head()
```

Out[10]:

	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

In [12]:

```
x = df.drop(columns = ['Purchased'] , axis = 1)
y = df['Purchased']
```

In [16]:

```
from sklearn.model_selection import train_test_split
```

In [17]:

```
x_train , x_test , y_train , y_test = train_test_split(x,y,test_size = 0.2 , random_state = 4
In [18]: from sklearn.ensemble import RandomForestClassifier

In [19]: rf = RandomForestClassifier()

In [20]: rf.fit(x_train , y_train)

Out[20]: ▾ RandomForestClassifier
         RandomForestClassifier()

In [21]: y_pred = rf.predict(x_test)

In [22]: from sklearn.metrics import accuracy_score

In [23]: accuracy_score(y_test , y_pred)

Out[23]: 0.9125

In [24]: from sklearn.linear_model import LogisticRegression

In [25]: lr = LogisticRegression()

In [26]: lr.fit(x_train , y_train)

Out[26]: ▾ LogisticRegression
         LogisticRegression()

In [27]: y_pred = lr.predict(x_test)

In [28]: accuracy_score(y_test , y_pred)

Out[28]: 0.7375

In [29]: from sklearn.tree import DecisionTreeClassifier

In [30]: dt = DecisionTreeClassifier()

In [31]: dt.fit(x_train , y_train)

Out[31]: ▾ DecisionTreeClassifier
         DecisionTreeClassifier()

In [32]: y_pred = dt.predict(x_test)

In [33]: accuracy_score(y_test , y_pred)

Out[33]: 0.9
```

In [ ]:

```python

```

RadomForestRegressor

In [37]:

```python
df = pd.read_csv("D:\\Summer Training Video\\ML\\tips.csv")
```

In [38]:

```python
df.head()
```

Out[38]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [40]:

```python
from sklearn.preprocessing import LabelEncoder
```

In [41]:

```python
lb = LabelEncoder()
```

In [42]:

```python
df['sex'] = lb.fit_transform(df['sex'])
df['smoker'] = lb.fit_transform(df['smoker'])
df['day'] = lb.fit_transform(df['day'])
df['time'] = lb.fit_transform(df['time'])
```

In [43]:

```python
df.head(2)
```

Out[43]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	0	0	2	0	2
1	10.34	1.66	1	0	2	0	3

In [44]:

```python
x = df.drop(columns = ['total_bill'] , axis = 1)
y = df['total_bill']
```

In [45]:

```python
from sklearn.model_selection import train_test_split
```

In [46]:

```python
from sklearn.linear_model import LinearRegression
```

In [47]:

```python
lr = LinearRegression()
```

In [49]:

```python
lr.fit(x_train , y_train)
```

Out[49]:

```
▾ LinearRegression
LinearRegression()
```

```
In [50]: y_pred = lr.predict(x_test)
```

```
In [51]: from sklearn.metrics import r2_score
```

```
In [55]: r2_score(y_test , y_pred)
```

```
Out[55]: 0.5626536669422029
```

```
In [56]: from sklearn.tree import DecisionTreeRegressor
```

```
In [57]: dt = DecisionTreeRegressor()
```

```
In [58]: dt.fit(x_train , y_train)
```

```
Out[58]: ▾ DecisionTreeRegressor
         DecisionTreeRegressor()
```

```
In [60]: y_pred = dt.predict(x_test)
```

```
In [61]: r2_score(y_test , y_pred)
```

```
Out[61]: 0.5054945054945056
```

```
In [62]: from sklearn.ensemble import RandomForestClassifier
```

```
In [63]: rf = RandomForestClassifier()
```

```
In [65]: rf.fit(x_train , y_train)
```

```
Out[65]: ▾ RandomForestClassifier
         RandomForestClassifier()
```

```
In [67]: y_pred = rf.predict(x_test)
```

```
In [68]: r2_score(y_test , y_pred)
```

```
Out[68]: 0.7252747252747254
```

```
In [ ]:
```