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

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

Out[2]:

	id	full_name	age	gender	device_type	ad_position	browsing_history	time_of_day	click
0	670	User670	22.0	NaN	Desktop	Top	Shopping	Afternoon	1
1	3044	User3044	NaN	Male	Desktop	Top	NaN	NaN	1
2	5912	User5912	41.0	Non-Binary	NaN	Side	Education	Night	1
3	5418	User5418	34.0	Male	NaN	NaN	Entertainment	Evening	1
4	9452	User9452	39.0	Non-Binary	NaN	NaN	Social Media	Morning	0
...	...	...	...	...	...	...	...	...	...
9995	8510	User8510	NaN	NaN	Mobile	Top	Education	NaN	0
9996	7843	User7843	NaN	Female	Desktop	Bottom	Entertainment	NaN	0
9997	3914	User3914	NaN	Male	Mobile	Side	NaN	Morning	0
9998	7924	User7924	NaN	NaN	Desktop	NaN	Shopping	Morning	1
9999	3056	User3056	44.0	Male	Tablet	Top	Social Media	Morning	0

10000 rows × 9 columns

In [3]:
```
dataset=pd.get_dummies(dataset,drop_first=True)
dataset.dropna(inplace=True)
dataset
```

Out[3]:

	id	age	click	full_name_User100	full_name_User1000	full_name_User10000	full_name_User1001	full_name_User1002	fu
0	670	22.0	1	False	False	False	False	False	
2	5912	41.0	1	False	False	False	False	False	
3	5418	34.0	1	False	False	False	False	False	
4	9452	39.0	0	False	False	False	False	False	
6	7808	26.0	1	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	
9987	2876	23.0	1	False	False	False	False	False	
9988	2713	52.0	0	False	False	False	False	False	
9990	9540	64.0	0	False	False	False	False	False	
9993	503	43.0	1	False	False	False	False	False	
9999	3056	44.0	0	False	False	False	False	False	

5234 rows × 4015 columns

In [4]:
```
dataset.columns
```

Out[4]:
```
Index(['id', 'age', 'click', 'full_name_User100', 'full_name_User1000',
       'full_name_User10000', 'full_name_User1001', 'full_name_User1002',
       'full_name_User1011', 'full_name_User1012',
       ...
       'device_type_Tablet', 'ad_position_Side', 'ad_position_Top',
       'browsing_history_Entertainment', 'browsing_history_News',
       'browsing_history_Shopping', 'browsing_history_Social Media',
       'time_of_day_Evening', 'time_of_day_Morning', 'time_of_day_Night'],
      dtype='object', length=4015)
```

In [5]:
```
independent=dataset[['id', 'age', 'full_name_User100', 'full_name_User1000',
       'full_name_User10000', 'full_name_User1001', 'full_name_User1002',
       'full_name_User1011', 'full_name_User1012',
       'device_type_Tablet', 'ad_position_Side', 'ad_position_Top',
       'browsing_history_Entertainment', 'browsing_history_News',
       'browsing_history_Shopping', 'browsing_history_Social Media',
       'time_of_day_Evening', 'time_of_day_Morning', 'time_of_day_Night']]
dependent=dataset[['click']]
```

In [6]:
```
independent.shape
```

Out[6]: (5234, 19)

In [7]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(independent,dependent,test_size=1/3,random_state=0)
```

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

In [9]:
```
from sklearn.ensemble import RandomForestClassifier
classifier=RandomForestClassifier(n_estimators=10,criterion='entropy',random_state=0)
classifier.fit(X_train,y_train)
```

```
C:\Users\SowmiGanesh\anaconda3\envs\aiml\Lib\site-packages\sklearn\base.py:1474: DataConversionWarning: A column
-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example usi
ng ravel().
  return fit_method(estimator, *args, **kwargs)
```

Out[9]:
```
▾ RandomForestClassifier
RandomForestClassifier(criterion='entropy', n_estimators=10, random_state=0)
```

In [10]:
```
from sklearn.model_selection import GridSearchCV
param_grid={'criterion':['gini','entropy'],'max_features':['auto','sqrt','log2']}
grid=GridSearchCV(RandomForestClassifier(),param_grid,refit=True,verbose=3,n_jobs=-1,scoring='f1_weighted')
grid.fit(X_train,y_train)
```

```
Fitting 5 folds for each of 6 candidates, totalling 30 fits
```

```
C:\Users\SowmiGanesh\anaconda3\envs\aiml\Lib\site-packages\sklearn\model_selection\_validation.py:547: FitFailed
Warning:
10 fits failed out of a total of 30.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score='raise'.

Below are more details about the failures:
--------------------------------------------------------------------------------
6 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\SowmiGanesh\anaconda3\envs\aiml\Lib\site-packages\sklearn\model_selection\_validation.py", line
895, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\SowmiGanesh\anaconda3\envs\aiml\Lib\site-packages\sklearn\base.py", line 1467, in wrapper
    estimator._validate_params()
  File "C:\Users\SowmiGanesh\anaconda3\envs\aiml\Lib\site-packages\sklearn\base.py", line 666, in _validate_para
ms
    validate_parameter_constraints(
  File "C:\Users\SowmiGanesh\anaconda3\envs\aiml\Lib\site-packages\sklearn\utils\_param_validation.py", line 95,
in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'max_features' parameter of RandomForestClassifier mu
st be an int in the range [1, inf), a float in the range (0.0, 1.0], a str among {'sqrt', 'log2'} or None. Got '
auto' instead.

--------------------------------------------------------------------------------
4 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\SowmiGanesh\anaconda3\envs\aiml\Lib\site-packages\sklearn\model_selection\_validation.py", line
895, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\SowmiGanesh\anaconda3\envs\aiml\Lib\site-packages\sklearn\base.py", line 1467, in wrapper
    estimator._validate_params()
  File "C:\Users\SowmiGanesh\anaconda3\envs\aiml\Lib\site-packages\sklearn\base.py", line 666, in _validate_para
ms
    validate_parameter_constraints(
  File "C:\Users\SowmiGanesh\anaconda3\envs\aiml\Lib\site-packages\sklearn\utils\_param_validation.py", line 95,
in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'max_features' parameter of RandomForestClassifier mu
st be an int in the range [1, inf), a float in the range (0.0, 1.0], a str among {'log2', 'sqrt'} or None. Got '
auto' instead.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\Users\SowmiGanesh\anaconda3\envs\aiml\Lib\site-packages\sklearn\model_selection\_search.py:1051: UserWarning:
One or more of the test scores are non-finite: [       nan 0.79601686 0.79647215        nan 0.79409281 0.7966356
3]
  warnings.warn(
C:\Users\SowmiGanesh\anaconda3\envs\aiml\Lib\site-packages\sklearn\base.py:1474: DataConversionWarning: A column
-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example usi
ng ravel().
  return fit_method(estimator, *args, **kwargs)
```

Out[10]:

```
▸         GridSearchCV
▸ estimator: RandomForestClassifier
    ▸ RandomForestClassifier
```

In [11]:

```
re=grid.cv_results_
grid_prediction=grid.predict(X_test)
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,grid_prediction)
```

In [12]:

```
print(cm)
```

```
[[ 398  265]
 [  78 1004]]
```

In [13]:

```
from sklearn.metrics import classification_report
clf_report=classification_report(y_test,grid_prediction)
```

In [14]:

```
print(clf_report)
```

```
              precision    recall  f1-score   support

           0       0.84      0.60      0.70       663
           1       0.79      0.93      0.85      1082

    accuracy                           0.80      1745
   macro avg       0.81      0.76      0.78      1745
weighted avg       0.81      0.80      0.80      1745
```

In [15]:

```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test,grid.predict_proba(X_test)[:,1])
```

Out[15]:

```
0.9263144615161577
```

In [16]:

```
table=pd.DataFrame.from_dict(re)
dataset.dropna(inplace=True)
table
```

Out[16]:

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_criterion	param_max_features	params	split0_test_s
0	0.002797	0.000436	0.000000	0.000000	gini	auto	{'criterion': 'gini', 'max_features': 'auto'}	
1	0.675090	0.029336	0.037086	0.003187	gini	sqrt	{'criterion': 'gini', 'max_features': 'sqrt'}	0.78
2	0.681624	0.028909	0.041385	0.003337	gini	log2	{'criterion': 'gini', 'max_features': 'log2'}	0.80
3	0.002977	0.002460	0.000000	0.000000	entropy	auto	{'criterion': 'entropy', 'max_features': 'auto'}	
4	0.681772	0.059910	0.036015	0.008029	entropy	sqrt	{'criterion': 'entropy', 'max_features': 'sqrt'}	0.79
5	0.618156	0.022124	0.029342	0.001790	entropy	log2	{'criterion': 'entropy', 'max_features': 'log2'}	0.78

In [17]:

```
id_input=int(input("ID:"))
age_input=int(input("age:"))
full_name_User100_input=int(input("Username100:"))
full_name_User1000_input=int(input("Username1000:"))
full_name_User10000_input=int(input("Username10000:"))
full_name_User1001_input=int(input("Username1001:"))
full_name_User1002_input=int(input("Username1002:"))
full_name_User1011_input=int(input("Username1011:"))
full_name_User1012_input=int(input("Username1012:"))
device_type_Tablet_input=int(input("Tablet:"))
```

```
ad_position_Side_input=int(input("Side Position:"))
ad_position_Top_input=int(input("Top Position:"))
browsing_history_Entertainment_input=int(input("Entertainment:"))
browsing_history_News_input=int(input("News:"))
browsing_history_Shopping_input=int(input("Shopping:"))
browsing_history_Social_Media_input=int(input("SocialMedia:"))
time_of_day_Evening_input=int(input("Evening Time:"))
time_of_day_Morning_input=int(input("Morning Time:"))
time_of_day_Night_input=int(input("Night Time:"))
```

In [18]:

```
ad_click_dataset_Prediction=grid.predict([[id_input,age_input,full_name_User100_input,full_name_User1000_input,
       full_name_User1011_input,full_name_User1012_input,device_type_Tablet_input, ad_position_Side_input, ad_p
        browsing_history_News_input,browsing_history_Shopping_input, browsing_history_Social_Media_input, time_
print("click Prediction{}".format(ad_click_dataset_Prediction))
```

```
click Prediction[0]
```

In [19]:

```
# pickle is used to save model creation
import pickle
# create filename it is pickle extension so we save.sav
filename="finalized_model_randomforestclassifier.sav"
```

In [20]:

```
pickle.dump(classifier, open(filename,"wb"))
```

In [24]:

```
# load the model and rb is used for just read
loaded_model=pickle.load(open("finalized_model_randomforestclassifier.sav","rb"))
# we click it and for prediction we can do it
result=loaded_model.predict([[2,25,1,2,10,15,20,15,20,25,25,20,10,15,20,20,10,30,15]])
```

In [25]:

```
result
```

Out[25]:

```
array([0], dtype=int64)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Loading [MathJax]/extensions/Safe.js