




Assignment := A5 Data Analytics II

1. Implement logistic regression using Python/R to perform classification on Social_Network_Ads.csv dataset.
2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix , accuracy_score,precision_score,recall_score
```

```
df = pd.read_csv('/content/Social_Network.zip')
df
```

	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	
...	
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




Next steps:

[Generate code with df](#)[View recommended plots](#)

Convert gender columns into integer , because operation is performed on integer value not on string

```
df['Gender'].replace({'Male':0 , 'Female':1} , inplace=True)
```

df

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

400 rows × 5 columns

Next steps:

[Generate code with df](#)[View recommended plots](#)

df.columns

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

Divide the dataset columns into two groups x and y. X is independent variable and y is dependent on x.

```
x = df[['User ID', 'Gender', 'Age', 'EstimatedSalary']]
y = df[['Purchased']]
```

Use train_test_split() function

```
x_train , x_test , y_train , y_test = train_test_split(x,y,test_size=0.25,random_state=29)
```

```
model = LogisticRegression()
model.fit(x_train,y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d
y = column_or_1d(y, warn=True)
```

```
▼ LogisticRegression
LogisticRegression()
```

```
y_pred = model.predict(x_test)
```

```
y_pred
```

```
array([0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0])
```

```
model.score(x_train,y_train)
```

```
0.7833333333333333
```

```
model.score(x,y)
```

```
0.785
```

Confusion Matrix. TP = True Positive, FP = False Positive(value=false , model=true), TN =True Negative(value=true , model =false) , FN = False Negative(value= false, model=false)

```
cm = confusion_matrix(y_test,y_pred)
cm
```

```
array([[64,  5],
       [16, 15]])
```

```
tn ,fp , fn,tp = confusion_matrix(y_test,y_pred).ravel()
```

```
print(tn ,fp , fn,tp)
```

```
64 5 16 15
```

Find Accuracy

```
a = accuracy_score(y_test,y_pred)
a
```

```
0.79
```

Error rate (Formula = 1 - accuracy_score)

```
e = 1 - a  
e
```

```
0.20999999999999996
```

Precision

```
p = precision_score(y_test , y_pred)  
p
```

```
0.75
```

Recall

```
r = recall_score(y_test , y_pred)  
r
```

```
0.4838709677419355
```