

Import Necessary Libraries

```
In [75]: import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_auc_score
```

Business Problem

Whether the client has subscribed a term deposit or not

Binomial ('yes' or 'no')

Data Collection for Performing Logistic Regression

```
In [76]: bank_data=pd.read_csv('bank-full.csv',sep=';')
```

```
Out[76]:
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no
...
45206	51	technician	married	tertiary	no	825	no	no	cellular	17	nov	977	3	-1	0	unknown	yes
45207	71	retired	divorced	primary	no	1729	no	no	cellular	17	nov	456	2	-1	0	unknown	yes
45208	72	retired	married	secondary	no	5715	no	no	cellular	17	nov	1127	5	184	3	success	yes
45209	57	blue-collar	married	secondary	no	668	no	no	telephone	17	nov	508	4	-1	0	unknown	no
45210	37	entrepreneur	married	secondary	no	2971	no	no	cellular	17	nov	361	2	188	11	other	no

45211 rows x 17 columns

Data Understanding by performing Initial Analysis

```
In [77]: bank_data.shape
```

```
Out[77]: (45211, 17)
```

```
In [78]: bank_data.isna().sum() # No null values
```

```
Out[78]: age          0
job          0
marital      0
education    0
default      0
balance      0
housing      0
loan         0
contact      0
day          0
month        0
duration     0
campaign     0
pdays       0
previous     0
poutcome     0
y            0
dtype: int64
```

```
In [79]: bank_data.dtypes
```

```
Out[79]: age          int64
job          object
marital      object
education    object
default      object
balance      int64
housing      object
loan         object
contact      object
day          int64
month        object
duration     int64
campaign     int64
pdays       int64
previous     object
poutcome     object
y            object
dtype: object
```

```
In [80]: bank_data.describe
```

```
Out[80]: <bound method NDFrame.describe of
0      58      management      married      tertiary      no      2143      yes      no      unknown      5      may      261      1      -1      0      unknown      no
1      44      technician      single      secondary      no      29      yes      no      unknown      5      may      151      1      -1      0      unknown      no
2      33      entrepreneur      married      secondary      no      2      yes      yes      unknown      5      may      76      1      -1      0      unknown      no
3      47      blue-collar      married      unknown      no      1506      yes      no      unknown      5      may      92      1      -1      0      unknown      no
4      33      unknown      single      unknown      no      1      no      no      unknown      5      may      198      1      -1      0      unknown      no
...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...
45206   51      technician      married      tertiary      no      825      no      no      cellular      17      nov      977      3      -1      0      unknown      yes
45207   71      retired      divorced      primary      no      1729      no      no      cellular      17      nov      456      2      -1      0      unknown      yes
45208   72      retired      married      secondary      no      5715      no      no      cellular      17      nov      1127      5      184      3      success      yes
45209   57      blue-collar      married      secondary      no      668      no      no      telephone      17      nov      508      4      -1      0      unknown      no
45210   37      entrepreneur      married      secondary      no      2971      no      no      cellular      17      nov      361      2      188      11      other      no
```

```
[45211 rows x 17 columns]>
```

```
In [81]: sns.pairplot(bank_data)
```

```
Out[81]: <seaborn.axisgrid.PairGrid at 0x102569400>
```



```
In [82]: bank_data['y'].value_counts()
```

```
Out[82]: no      39922
yes       5289
Name: y, dtype: int64
```

```
In [83]: count_no_sub=len(bank_data[bank_data['y']!='no'])
count_sub=len(bank_data[bank_data['y']=='yes'])
(count_sub/(count_sub+count_no_sub))*100
```

```
Out[83]: 11.698480458295547
```

```
In [84]: bank_data=pd.get_dummies(data=bank_data,columns=['job','marital','education','contact','poutcome','month'])
bank_data
```

```
Out[84]:
```

	age	default	balance	housing	loan	day	duration	campaign	pdays	previous	y	job_admin	job_blue-collar	job_entrepreneur	job_housemaid	job_management	job_retired	job_self-employed	job_services
0	58	no	2143	yes	no	5	261	1	-1	0	no	0	0	0	0	0	1	0	0
1	44	no	29	yes	no	5	151	1	-1	0	no	0	0	0	0	0	0	0	0
2	33	no	2	yes	yes	5	76	1	-1	0	no	0	0	0	1	0	0	0	0
3	47	no	1506	yes	no	5	92	1	-1	0	no	0	1	0	0	0	0	0	0
4	33	no	1	no	no	5	198	1	-1	0	no	0	0	0	0	0	0	0	0
...
45206	51	no	825	no	no	17	977	3	-1	0	yes	0	0	0	0	0	0	0	0
45207	71	no	1729	no	no	17	456	2	-1	0	yes	0	0	0	0	0	0	1	0
45208	72	no	5715	no	no	17	1127	5	184	3	yes	0	0	0	0	0	0	1	0
45209	57	no	668	no	no	17	508	4	-1	0	no	0	1	0	0	0	0	0	0
45210	37	no	2971	no	no	17	361	2	188	11	no	0	0	1	0	0	0	0	0

45211 rows x 49 columns

```
In [85]: pd.set_option('display.max.columns',None)
bank_data1
```

```
Out[85]:
```

	age	default	balance	housing	loan	day	duration	campaign	pdays	previous	y	job_admin	job_blue-collar	job_entrepreneur	job_housemaid	job_management	job_retired	job_self-employed	job_services
0	58	no	2143	yes	no	5	261	1	-1	0	no	0	0	0	0	0	1	0	0
1	44	no	29	yes	no	5	151	1	-1	0	no	0	0	0	0	0	0	0	0
2	33	no	2	yes	yes	5	76	1	-1	0	no	0	0	0	1	0	0	0	0
3	47	no	1506	yes	no	5	92	1	-1	0	no	0	1	0	0	0	0	0	0
4	33	no	1	no	no	5	198	1	-1	0	no	0	0	0	0	0	0	0	0
...
45206	51	no	825	no	no	17	977	3	-1	0	yes	0	0	0	0	0	0	0	0
45207	71	no	1729	no	no	17	456	2	-1	0	yes	0	0	0	0	0	0	1	0
45208	72	no	5715	no	no	17	1127	5	184	3	yes	0	0	0	0	0	0	1	0
45209	57	no	668	no	no	17	508	4	-1	0	no	0	1	0	0	0	0	0	0
45210	37	no	2971	no	no	17	361	2	188	11	no	0	0	1	0	0	0	0	0

45211 rows x 49 columns

```
In [86]: bank_data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 49 columns):
#   Column              Non-Null Count  Dtype
---  ---
0   age                  45211 non-null  int64
1   default              45211 non-null  object
2   balance              45211 non-null  int64
3   housing              45211 non-null  object
4   loan                 45211 non-null  object
5   day                  45211 non-null  int64
6   duration              45211 non-null  int64
7   campaign              45211 non-null  int64
8   pdays                45211 non-null  int64
9   previous              45211 non-null  int64
10  y                     45211 non-null  object
11  job_admin             45211 non-null  uint8
12  job_blue-collar       45211 non-null  uint8
13  job_entrepreneur       45211 non-null  uint8
14  job_housemaid          45211 non-null  uint8
15  job_management         45211 non-null  uint8
16  job_retired            45211 non-null  uint8
17  job_self-employed      45211 non-null  uint8
18  job_services           45211 non-null  uint8
19  job_student            45211 non-null  uint8
20  job_technician         45211 non-null  uint8
21  job_unemployed         45211 non-null  uint8
22  job_unknown            45211 non-null  uint8
23  marital_divorced       45211 non-null  uint8
24  marital_married        45211 non-null  uint8
25  marital_single         45211 non-null  uint8
26  education_primary      45211 non-null  uint8
27  education_secondary    45211 non-null  uint8
28  education_tertiary     45211 non-null  uint8
29  education_unknown      45211 non-null  uint8
30  contact_cellular       45211 non-null  uint8
31  contact_telephone      45211 non-null  uint8
32  contact_unknown        45211 non-null  uint8
33  poutcome_failure       45211 non-null  uint8
34  poutcome_other         45211 non-null  uint8
35  poutcome_success       45211 non-null  uint8
36  poutcome_unknown       45211 non-null  uint8
37  month_apr              45211 non-null  uint8
38  month_aug              45211 non-null  uint8
39  month_dec              45211 non-null  uint8
40  month_feb              45211 non-null  uint8
41  month_jan              45211 non-null  uint8
42  month_jul              45211 non-null  uint8
43  month_jun              45211 non-null  uint8
44  month_mar              45211 non-null  uint8
45  month_may              45211 non-null  uint8
46  month_nov              45211 non-null  uint8
47  month_oct              45211 non-null  uint8
48  month_sep              45211 non-null  uint8
dtypes: int64(7), object(4), uint8(38)
memory usage: 5.4 MB
```

```
In [87]: bank_data1['default'] = np.where(bank_data1['default'].str.contains("yes"), 1, 0)
bank_data1['housing'] = np.where(bank_data1['housing'].str.contains("yes"), 1, 0)
bank_data1['loan'] = np.where(bank_data1['loan'].str.contains("yes"), 1, 0)
bank_data1['y'] = np.where(bank_data1['y'].str.contains("yes"), 1, 0)
```

```
Out[87]:
```

	age	default	balance	housing	loan	day	duration	campaign	pdays	previous	y	job_admin	job_blue-collar	job_entrepreneur	job_housemaid	job_management	job_retired	job_self-employed	job_services
0	58	0	2143	1	0	5	261	1	-1	0	0	0	0	0	0	0	1	0	0
1	44	0	29	1	0	5	151	1	-1	0	0	0	0	0	0	0	0	0	0
2	33	0	2	1	1	5	76	1	-1	0	0	0	0	1	0	0	0	0	0
3	47	0	1506	1	0	5	92	1	-1	0	0	0	1	0	0	0	0	0	0
4	33	0	1	0	0	5	198	1	-1	0	0	0	0	0	0	0	0	0	0
...
45206	51	0	825	0	0	17	977	3	-1	0	1	0	0	0	0	0	0	0	0
45207	71	0	1729	0	0	17	456	2	-1	0	1	0	0	0	0	0	0	1	0
45208	72	0	5715	0	0	17	1127	5	184	3	1	0	0	0	0	0	0	1	0
45209	57	0	668	0	0	17	508	4	-1	0	0	0	1	0	0	0	0	0	0
45210	37	0	2971	0	0	17	361	2	188	11	0	0	0	1	0	0	0	0	0

45211 rows x 49 columns

```
In [88]: bank_data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 49 columns):
#   Column              Non-Null Count  Dtype
---  ---
0   age                  45211 non-null  int64
1   default              45211 non-null  int64
2   balance              45211 non-null  int64
3   housing              45211 non-null  int32
4   loan                 45211 non-null  int32
5   day                  45211 non-null  int64
6   duration              45211 non-null  int64
7   campaign              45211 non-null  int64
8   pdays                45211 non-null  int64
9   previous              45211 non-null  int64
10  y                     45211 non-null  int32
11  job_admin             45211 non-null  uint8
12  job_blue-collar       45211 non-null  uint8
13  job_entrepreneur       45211 non-null  uint8
14  job_housemaid          45211 non-null  uint8
15  job_management         45211 non-null  uint8
16  job_retired            45211 non-null  uint8
17  job_self-employed      45211 non-null  uint8
18  job_services           45211 non-null  uint8
19  job_student            45211 non-null  uint8
20  job_technician         45211 non-null  uint8
21  job_unemployed         45211 non-null  uint8
22  job_unknown            45211 non-null  uint8
23  marital_divorced       45211 non-null  uint8
24  marital_married        45211 non-null  uint8
25  marital_single         45211 non-null  uint8
26  education_primary      45211 non-null  uint8
27  education_secondary    45211 non-null  uint8
28  education_tertiary     45211 non-null  uint8
29  education_unknown      45211 non-null  uint8
30  contact_cellular       45211 non-null  uint8
31  contact_telephone      45211 non-null  uint8
32  contact_unknown        45211 non-null  uint8
33  poutcome_failure       45211 non-null  uint8
34  poutcome_other         45211 non-null  uint8
35  poutcome_success       45211 non-null  uint8
36  poutcome_unknown       45211 non-null  uint8
37  month_apr              45211 non-null  uint8
38  month_aug              45211 non-null  uint8
39  month_dec              45211 non-null  uint8
40  month_feb              45211 non-null  uint8
41  month_jan              45211 non-null  uint8
42  month_jul              45211 non-null  uint8
43  month_jun              45211 non-null  uint8
44  month_mar              45211 non-null  uint8
45  month_may              45211 non-null  uint8
46  month_nov              45211 non-null  uint8
47  month_oct              45211 non-null  uint8
48  month_sep              45211 non-null  uint8
dtypes: int64(4), int64(7), uint8(38)
memory usage: 4.7 MB
```

Model Building

```
In [89]: import warnings
warnings.filterwarnings('ignore')
```

```
In [90]: x=pd.concat([bank_data1.iloc[:,0:10],bank_data1.iloc[:,11:]],axis=1)
y=bank_data1.iloc[:,10]
```

```
In [91]: classifier=LogisticRegression()
classifier.fit(x,y)
```

```
Out[91]: LogisticRegression()
```

Model Predictions

```
In [92]: y_pred=classifier.predict(x)
y_pred
```

```
Out[92]: array([0, 0, ..., 1, 0, 0])
```

```
In [93]: y_pred_bank_data=pd.DataFrame({'actual_y':y,'y_pred_prob':y_pred})
y_pred_bank_data
```

```
Out[93]:
```

	actual_y	y_pred_prob
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
...
45206	1	1
45207	1	0
45208	1	1
45209	0	0
45210	0	0

45211 rows x 2 columns

Testing Model Accuracy

```
In [94]: # confusion matrix for the model accuracy
confusion_matrix=confusion_matrix(y,y_pred)
confusion_matrix
```

```
Out[94]: array([[39159,   763],
               [ 4140, 1149]], dtype=int64)
```

```
In [95]: # The Model accuracy is calculated by (a+d/(a+b+c+d))
(39159+1149)/(39159+763+4140+1149)
```

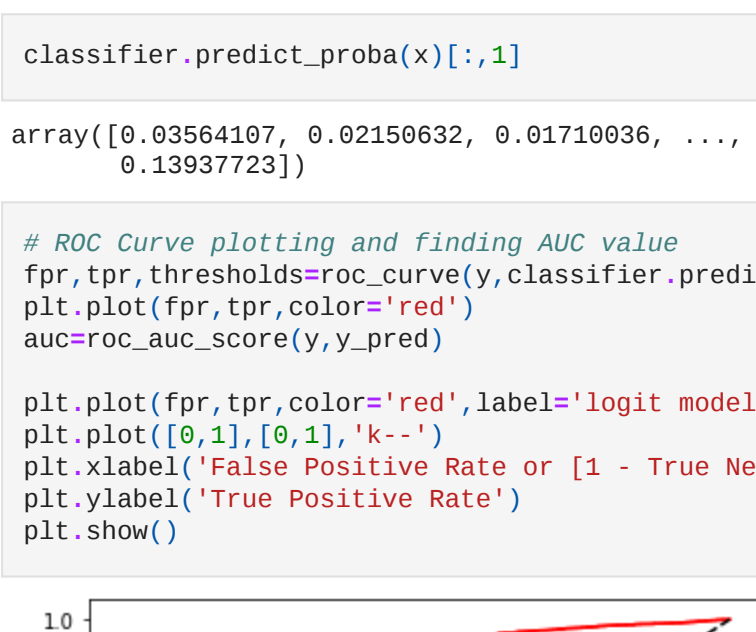
```
Out[95]: 0.8915529406560351
```

```
In [96]: classifier.predict_proba(x)[:,1]
```

```
Out[96]: array([0.03564107, 0.02150632, 0.01710036, ..., 0.63604954, 0.06555453,
               0.03937723])
```

```
In [96]: # ROC Curve plotting and finding AUC value
fpr, tpr, thresholds=roc_curve(y,classifier.predict_proba(x)[:,1])
plt.plot(fpr,tpr,color='red')
auc=roc_auc_score(y,y_pred)
```

```
plt.plot(fpr,tpr,color='red',label='logit model(area = %.2f)'%auc)
plt.plot([0,1],[0,1],k--')
plt.xlabel('False Positive Rate or [1 - True Negative Rate]')
plt.ylabel('True Positive Rate')
plt.show()
```



```
In [97]: print('auc accuracy:',auc)
```

```
auc accuracy: 0.599065531498237
```