1. Import Necessary Libraries

In [17]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,classification_report,roc_auc_score,roc_curve,confusion_matrix
from sklearn.tree import DecisionTreeClassifier


import warnings
warnings.filterwarnings('ignore')
```

2. Import Data

In [2]:
```
bank_data = pd.read_csv('bank-full.csv',sep =';')
bank_data.head()
```

Out[2]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown

3. Data Understanding

3.1 Perform Initial Analysis

In [3]:
```
bank_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   age        45211 non-null  int64 
 1   job        45211 non-null  object
 2   marital    45211 non-null  object
 3   education  45211 non-null  object
 4   default    45211 non-null  object
 5   balance    45211 non-null  int64 
 6   housing    45211 non-null  object
 7   loan       45211 non-null  object
 8   contact    45211 non-null  object
 9   day        45211 non-null  int64 
 10  month      45211 non-null  object
 11  duration   45211 non-null  int64 
 12  campaign   45211 non-null  int64 
 13  pdays      45211 non-null  int64 
 14  previous   45211 non-null  int64 
 15  poutcome   45211 non-null  object
 16  y          45211 non-null  object
dtypes: int64(7), object(10)
memory usage: 5.9+ MB
```

In [4]:
```
bank_data.describe()
```

Out[4]:

	age	balance	day	duration	campaign	pdays	previous
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000
mean	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197828	0.580323
std	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128746	2.303441
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000	0.000000
25%	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000000	0.000000
50%	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000000	0.000000
75%	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000000	0.000000
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000	275.000000

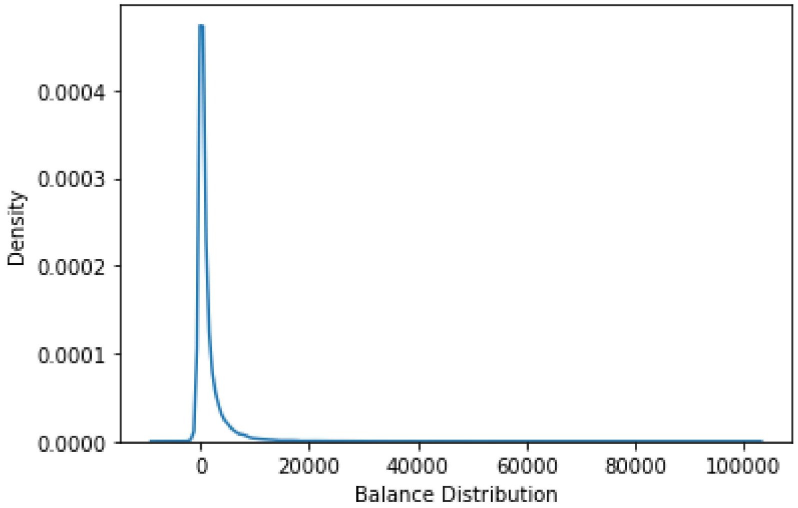
In [5]:
```
bank_data.isna().sum()
```

Out[5]:
```
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
```

3.2 Assumptions Check

Normality Test

In [6]:
```
sns.kdeplot(x = bank_data['balance'])
plt.xlabel('Balance Distribution')
plt.show()
```



Normality Test is Passed

4. Data Preparation

Data Tranformation by using One Hot Encoding for categorical features

In [7]:
```
bank_data
```

Out[7]:

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

45211 rows × 17 columns

In [8]:
```
encoded_bank_data = pd.get_dummies(data=bank_data,columns=['job','marital','education','contact','month','poutcome'])
```

In [9]:
```
encoded_bank_data
```

Out[9]:

	age	default	balance	housing	loan	day	duration	campaign	pdays	previous	...	month_jun	month_mar	month_may	month_nov	montl
0	58	no	2143	yes	no	5	261	1	-1	0	...	0	0	1	0	
1	44	no	29	yes	no	5	151	1	-1	0	...	0	0	1	0	
2	33	no	2	yes	yes	5	76	1	-1	0	...	0	0	1	0	
3	47	no	1506	yes	no	5	92	1	-1	0	...	0	0	1	0	
4	33	no	1	no	no	5	198	1	-1	0	...	0	0	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
45206	51	no	825	no	no	17	977	3	-1	0	...	0	0	0	1	
45207	71	no	1729	no	no	17	456	2	-1	0	...	0	0	0	1	
45208	72	no	5715	no	no	17	1127	5	184	3	...	0	0	0	1	
45209	57	no	668	no	no	17	508	4	-1	0	...	0	0	0	1	
45210	37	no	2971	no	no	17	361	2	188	11	...	0	0	0	1	

45211 rows × 49 columns

In [10]:
```
pd.set_option("display.max.columns", None)
encoded_bank_data
```

Out[10]:

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

45211 rows × 49 columns

In [11]:
```
encoded_bank_data.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  month_apr            45211 non-null  uint8 
```

```
 34  month_aug           45211 non-null  uint8
 35  month_dec           45211 non-null  uint8
 36  month_feb           45211 non-null  uint8
 37  month_jan           45211 non-null  uint8
 38  month_jul           45211 non-null  uint8
 39  month_jun           45211 non-null  uint8
 40  month_mar           45211 non-null  uint8
 41  month_may           45211 non-null  uint8
 42  month_nov           45211 non-null  uint8
 43  month_oct           45211 non-null  uint8
 44  month_sep           45211 non-null  uint8
 45  poutcome_failure    45211 non-null  uint8
 46  poutcome_other      45211 non-null  uint8
 47  poutcome_success    45211 non-null  uint8
 48  poutcome_unknown    45211 non-null  uint8
dtypes: int64(7), object(4), uint8(38)
memory usage: 5.4+ MB
```

In [14]:
```
encoded_bank_data['default'] = np.where(encoded_bank_data['default'].str.contains("yes"), 1, 0)
encoded_bank_data['housing'] = np.where(encoded_bank_data['housing'].str.contains("yes"), 1, 0)
encoded_bank_data['loan'] = np.where(encoded_bank_data['loan'].str.contains("yes"), 1, 0)
encoded_bank_data['y'] = np.where(encoded_bank_data['y'].str.contains("yes"), 1, 0)
encoded_bank_data
```

Out[14]:

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

45211 rows × 49 columns

5. Model Building

In [15]:
```
#separate X & y
X = encoded_bank_data.drop('y',axis=1)
y = encoded_bank_data[['y']]
```

In [18]:
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.20,random_state=12)
```

In [19]:
```
X_train.shape,y_train.shape
```

Out[19]:
```
((36168, 48), (36168, 1))
```

In [20]:
```
X_test.shape,y_test.shape
```

Out[20]:
```
((9043, 48), (9043, 1))
```

6. Model Training

===

If Attorney = 1, Client is going to subscribe a term deposit.

If Attorney = 0, Client is not going to subscribe a term deposit.

===

In [21]:
```
logistic_model = LogisticRegression()
logistic_model.fit(X_train,y_train)

dt_model = DecisionTreeClassifier()
dt_model.fit(X_train,y_train)
```

Out[21]:
```
DecisionTreeClassifier()
```

In [22]:
```
logistic_model.coef_
```

Out[22]:
```
array([[-1.72589262e-02, -1.94250817e-02,  2.07870353e-05,
        -7.74155857e-01, -2.27598642e-01, -1.15749898e-02,
         3.78706091e-03, -3.75261112e-01,  2.69550196e-03,
        -1.72573853e-01, -3.23236553e-02, -2.87235198e-01,
        -2.83796444e-02, -8.90599092e-03,  9.07165922e-03,
         1.91805209e-01, -1.97634057e-02, -1.08969703e-01,
         3.53003741e-02, -7.73938128e-02, -2.86320967e-03,
         3.41136984e-03,  2.27791876e-02, -2.03731763e-01,
        -1.45293433e-01, -7.29987494e-02, -3.21933161e-01,
         5.49615239e-02,  1.37243788e-02,  1.08142824e-01,
         5.94714289e-02, -4.93860261e-01,  3.62811763e-02,
         3.99236806e-02,  3.69060263e-02, -2.24869665e-02,
        -1.65020455e-02, -9.27508156e-02, -6.40452828e-02,
         1.04731173e-01, -4.79184220e-01, -7.26891357e-02,
         1.14869136e-01,  8.87012670e-02, -1.63219794e-01,
        -4.02240848e-02,  3.72974617e-01, -4.95776746e-01]])
```

In [23]:
```
logistic_model.intercept_
```

Out[23]:
```
array([-0.32628609])
```

6.Model Testing || 7.Model Evaluation

Train Data

In [24]:
```
y_pred_train = dt_model.predict(X_train)
```

In [25]:
```
print(confusion_matrix(y_train,y_pred_train))
```

```
[[31929     0]
 [    0  4239]]
```

In [26]:
```
print(classification_report(y_train,y_pred_train))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     31929
           1       1.00      1.00      1.00      4239

    accuracy                           1.00     36168
   macro avg       1.00      1.00      1.00     36168
weighted avg       1.00      1.00      1.00     36168
```

In [27]:
```
accuracy_score(y_train,y_pred_train)
```

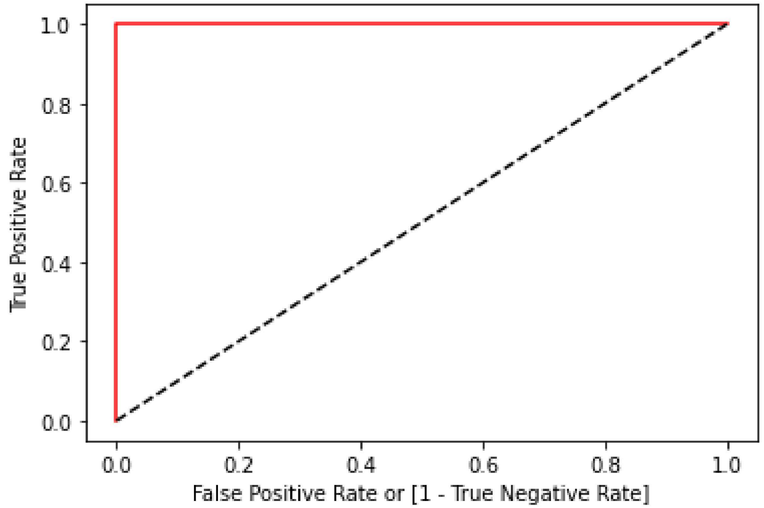
Out[27]:
```
1.0
```

In [33]:
```
fpr, tpr, thresholds = roc_curve(y_train,dt_model.predict_proba (X_train)[:,1])

auc = roc_auc_score(y_train,dt_model.predict_proba (X_train)[:,1])
print('auc accuracy:',auc)

plt.plot(fpr, tpr, color='red', label='logit model ( area  = %0.2f)'%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()
```

```
auc accuracy: 1.0
```



Test Data

In [29]:
```
y_pred_test = logistic_model.predict(X_test)
```

In [30]:
```
print(confusion_matrix(y_test,y_pred_test))
```

```
[[7846  147]
 [ 823  227]]
```

In [31]:
```
print(classification_report(y_test,y_pred_test))
```

```
              precision    recall  f1-score   support

           0       0.91      0.98      0.94      7993
           1       0.61      0.22      0.32      1050

    accuracy                           0.89      9043
   macro avg       0.76      0.60      0.63      9043
weighted avg       0.87      0.89      0.87      9043
```

In [32]:
```
accuracy_score(y_test,y_pred_test)
```

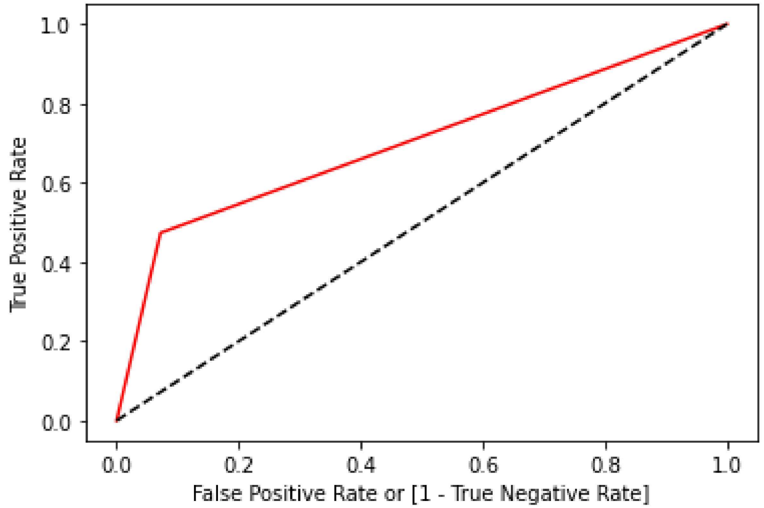
Out[32]:
```
0.892734711931881
```

In [34]:
```
fpr, tpr, thresholds = roc_curve(y_test,dt_model.predict_proba (X_test)[:,1])

auc_test = roc_auc_score(y_test,dt_model.predict_proba (X_test)[:,1])
print('auc accuracy:',auc_test)

plt.plot(fpr, tpr, color='red', label='logit model ( area  = %0.2f)'%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()
```

```
auc accuracy: 0.7007602485508153
```



THE END!!!