In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:

```
data = pd.read_csv('bank-full (1).csv',sep=';')
data.head()
```

Out[2]:

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

In [3]:

```
data.shape
```

Out[3]:

```
(45211, 17)
```

In [4]:

```
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 [5]:

```
data.isnull().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
```

In [6]:

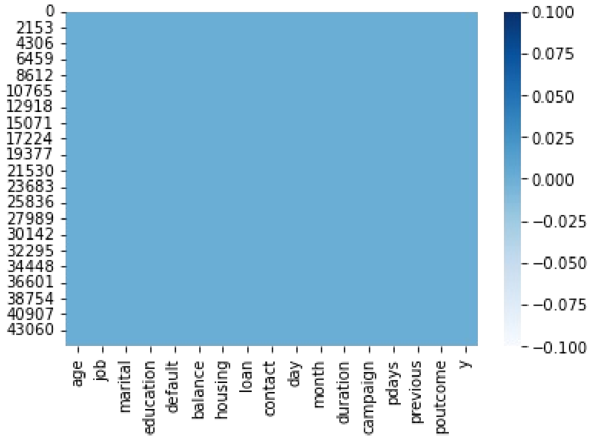
```
import seaborn as sns
```

In [7]:

```python
sns.heatmap(data.isnull(),cmap='Blues')
# there are no null values
```

Out[7]:

```
<AxesSubplot:>
```



In [8]:

```python
data.dtypes
```

Out[8]:

```
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      int64
poutcome     object
y            object
dtype: object
```

In [9]:

```python
# there are no duplicates
data.duplicated()
```

Out[9]:

```
0        False
1        False
2        False
3        False
4        False
         ...  
45206    False
45207    False
45208    False
45209    False
45210    False
Length: 45211, dtype: bool
```

In [10]:

```python
data.head(2)
```

Out[10]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcom
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknow
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknow

In [11]:

```python
# checking the categorical values
categorical = [var for var in data.columns if data[var].dtype=='O']

print('There are {} categorical variables\n'.format(len(categorical)))
print('The categorical variables are :', categorical)
```

```
There are 10 categorical variables

The categorical variables are : ['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact', 'month',
'poutcome', 'y']
```

In [12]:

```
data[categorical].isnull().sum()
```

Out[12]:

```
job          0
marital      0
education    0
default      0
housing      0
loan         0
contact      0
month        0
poutcome     0
y            0
dtype: int64
```

In [13]:

```
# covert  the categorical into dummies
data = pd.get_dummies(data,columns=['job','marital','education','contact','month','poutcome'])
```

In [14]:

```
data.head()
```

Out[14]:

	age	default	balance	housing	loan	day	duration	campaign	pdays	previous	...	month_jun	month_mar	month_may	month_nov	month
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	

5 rows × 49 columns

In [15]:

```
# convert the binary to string
data['default'] = np.where(data['default'].str.contains('yes'),1,0)
data['housing'] = np.where(data['housing'].str.contains('yes'),1,0)
data['loan'] = np.where(data['loan'].str.contains('yes'),1,0)
data['y']= np.where(data['y'].str.contains('yes'),1,0)
data
```

Out[15]:

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

45211 rows × 49 columns

In [16]:

```
data.iloc[:,15:30]
```

Out[16]:

	job_management	job_retired	job_self-employed	job_services	job_student	job_technician	job_unemployed	job_unknown	marital_divorced	mar

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

45211 rows × 15 columns

In [17]:

```
X = data.drop(columns='y')
X
```

Out[17]:

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

45211 rows × 48 columns

In [18]:

```
y_data = data.loc[:,'y']
y_data
```

Out[18]:

```
0        0
1        0
2        0
3        0
4        0
        ..
45206    1
45207    1
45208    1
45209    0
45210    0
Name: y, Length: 45211, dtype: int32
```

In [19]:

```
y_data.unique()
```

Out[19]:

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

In [20]:

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

In [21]:

```
X_train, X_test, y_train, y_test = train_test_split(X,y_data,test_size=0.3)
X_train, X_test, y_train, y_test
```

Out[21]:

```
(        age  default  balance  housing  loan  day  duration  campaign  pdays  \
 44745   62        0     2801        1     1    9       261         1    183
 44864   46        0     7485        0     0   23       145         1    779
```

```
34749   34        0      649        1     0    6       224         1    363
18099   42        0     2613        1     0   30       174         6     -1
1622    32        0      536        1     0    9       208         2     -1
...    ...      ...      ...      ...   ...  ...       ...       ...    ...
8120    60        0      -79        1     0    2        49         1     -1
39381   54        0     1188        1     0   22        89         1     -1
18512   35        0     -195        0     0   31       309         4     -1
29574   33        0     1883        1     0    3       121         5    256
655     33        0     -349        1     0    6       191         1     -1

       previous  ...  month_jun  month_mar  month_may  month_nov  month_oct  \
44745         1  ...          0          0          0          0          0
44864         2  ...          0          0          0          0          0
34749         1  ...          0          0          1          0          0
18099         0  ...          0          0          0          0          0
1622          0  ...          0          0          1          0          0
...         ...  ...        ...        ...        ...        ...        ...
8120          0  ...          1          0          0          0          0
39381         0  ...          0          0          1          0          0
18512         0  ...          0          0          0          0          0
29574         1  ...          0          0          0          0          0
655           0  ...          0          0          1          0          0

       month_sep  poutcome_failure  poutcome_other  poutcome_success  \
44745          1                 0               0                 1
44864          1                 1               0                 0
34749          0                 0               1                 0
18099          0                 0               0                 0
1622           0                 0               0                 0
...          ...               ...             ...               ...
8120           0                 0               0                 0
39381          0                 0               0                 0
18512          0                 0               0                 0
29574          0                 1               0                 0
655            0                 0               0                 0

       poutcome_unknown
44745                 0
44864                 0
34749                 0
18099                 1
1622                  1
...                 ...
8120                  1
39381                 1
18512                 1
29574                 0
655                   1

[31647 rows x 48 columns],
       age  default  balance  housing  loan  day  duration  campaign  pdays  \
10547   42        0      167        0     0   16       119         1     -1
40059   33        0      506        0     0    4       176         1     91
26737   32        0     6982        1     0   20       224         2    183
21071   57        0      209        0     0   14        56         4     -1
39637   29        0     2907        1     0   26       150         1     -1
...    ...      ...      ...      ...   ...  ...       ...       ...    ...
38989   31        0     1374        1     0   18       290         2    370
6063    33        0     2065        1     1   26       241         2     -1
20071   50        0      592        0     0    8       445         4     -1
20779   50        0       36        0     0   13       104        10     -1
32548   34        0      703        0     0   17       282         2    150

       previous  ...  month_jun  month_mar  month_may  month_nov  month_oct  \
10547         0  ...          1          0          0          0          0
40059         2  ...          1          0          0          0          0
26737         1  ...          0          0          0          1          0
21071         0  ...          0          0          0          0          0
39637         0  ...          0          0          1          0          0
...         ...  ...        ...        ...        ...        ...        ...
38989         1  ...          0          0          1          0          0
6063          0  ...          0          0          1          0          0
20071         0  ...          0          0          0          0          0
20779         0  ...          0          0          0          0          0
32548         2  ...          0          0          0          0          0

       month_sep  poutcome_failure  poutcome_other  poutcome_success  \
10547          0                 0               0                 0
40059          0                 0               0                 1
26737          0                 1               0                 0
21071          0                 0               0                 0
39637          0                 0               0                 0
...          ...               ...             ...               ...
38989          0                 1               0                 0
6063           0                 0               0                 0
20071          0                 0               0                 0
20779          0                 0               0                 0
32548          0                 1               0                 0
```

```
       poutcome_unknown
10547                 1
40059                 0
26737                 0
21071                 1
39637                 1
...                 ...
38989                 0
6063                  1
20071                 1
20779                 1
32548                 0

[13564 rows x 48 columns],
44745    1
44864    0
34749    0
18099    0
1622     0
        ..
8120     0
39381    0
18512    0
29574    0
655      0
Name: y, Length: 31647, dtype: int32,
10547    0
40059    0
26737    0
21071    0
39637    0
        ..
38989    0
6063     0
20071    0
20779    0
32548    0
Name: y, Length: 13564, dtype: int32)
```

In [22]:

```
from sklearn.linear_model import LogisticRegression
```

In [23]:

```
logistic_model = LogisticRegression()
```

In [24]:

```
logistic_model.fit(X_train,y_train)
```

```
C:\Users\rajesh\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs fail
ed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Out[24]:

```
LogisticRegression()
```

In [25]:

```
# for train data
y_pred = logistic_model.predict(X_train)

# for test data
y_pred_test = logistic_model.predict(X_test)
```

In [26]:

```
from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
```

In [27]:

```
# for train data
confusion_matrix(y_train,y_pred)
```

Out[27]:

```
array([[27607,   381],
       [ 3083,   576]], dtype=int64)
```

In [28]:

```
confusion_matrix(y_test,y_pred_test)
```

Out[28]:

```
array([[11788,   146],
       [ 1394,   236]], dtype=int64)
```

In [29]:

```
#The model accuracy is calculated by (a+d)/(a+b+c+d)
# for train data
print((31340+868)/(31340+590+3370+868))
print(accuracy_score(y_train,y_pred))
```

```
0.8905109489051095
0.890542547476854
```

In [30]:

```
# for test data
print(accuracy_score(y_test,y_pred_test))
```

```
0.8864641698613979
```

In [31]:

```
# for train data
print(classification_report(y_train,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.90      0.99      0.94     27988
           1       0.60      0.16      0.25      3659

    accuracy                           0.89     31647
   macro avg       0.75      0.57      0.60     31647
weighted avg       0.87      0.89      0.86     31647
```

In [32]:

```
# for test data
print(classification_report(y_test,y_pred_test))
```

```
              precision    recall  f1-score   support

           0       0.89      0.99      0.94     11934
           1       0.62      0.14      0.23      1630

    accuracy                           0.89     13564
   macro avg       0.76      0.57      0.59     13564
weighted avg       0.86      0.89      0.85     13564
```

In [33]:

```
from pickle import dump,load
```

In [34]:

```
logistic_model.predict(X_test)
```

Out[34]:

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

In [ ]:

In [ ]:

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js