Dealing with Imbalanced Dataset

Imbalanced data

How to resolve?

    Upsampling

    Downsampling

Introuce the KNearest Neighbor

Hyper parameter tuning using GridSearch CV

Idnetify the best model

Conclude

Imbalanced dataset

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

In [2]:
```python
bank=pd.read_csv('bank.csv')
bank
```

Out[2]:

	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 × 17 columns

In [3]:
```python
bank['y'].value_counts()
```

Out[3]:
```
no     39922
yes     5289
Name: y, dtype: int64
```

In [4]:
```python
5289/(5289+39922)
```

Out[4]:
```
0.11698480458295547
```

Yes class >>>> 11.7%

No class >>>> 88.3%

This is imbalanced data

Resolving imbalance

Upsampling

Downsampling

Split the data into two according to the class

bank_yes=bank[bank['y']=='yes'] bank_yes.shape

In [6]:
```
bank_no=bank[bank['y']=='no']
bank_no.shape
```

Out[6]: (39922, 17)

Upsampling

On Minority class.

Increased the contribution.

In [8]:
```
from sklearn.utils import resample

bank_yes_up=resample(bank_yes, replace=True,random_state=100,n_samples=15000)
bank_yes_up.shape
```

Out[8]: (15000, 17)

Downsampling

On majority class

Reduce the contrinution of majority class

In [9]:
```
bank_no_down=resample(bank_no,replace=False,random_state=100,n_samples=25000)
bank_no_down.shape
```

Out[9]: (25000, 17)

Creating a dataset by combaining

In [10]:
```
bank_new=pd.concat([bank_yes_up,bank_no_down])
bank_new.shape
```

Out[10]: (40000, 17)

In [11]:
```
bank_new.head(25)
```

Out[11]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
42571	38	admin.	married	secondary	no	11303	no	no	cellular	28	dec	473	2	216	2	failure	yes
3190	41	blue-collar	married	secondary	no	1384	yes	no	unknown	15	may	1162	4	-1	0	unknown	yes
10049	42	entrepreneur	married	tertiary	no	5345	no	no	unknown	11	jun	878	3	-1	0	unknown	yes
31962	31	blue-collar	married	secondary	no	1406	yes	yes	cellular	13	apr	1091	2	-1	0	unknown	yes
43024	51	management	married	tertiary	no	346	no	no	cellular	12	feb	122	1	92	5	success	yes
18015	55	management	married	tertiary	no	-375	no	no	cellular	30	jul	814	2	-1	0	unknown	yes
44802	31	management	single	tertiary	no	3340	no	no	cellular	15	sep	213	2	469	3	success	yes
39724	43	admin.	married	secondary	no	132	no	no	cellular	27	may	187	2	71	1	success	yes
43431	78	retired	divorced	primary	no	1389	no	no	cellular	8	apr	335	1	-1	0	unknown	yes
31180	25	technician	single	secondary	no	1231	yes	no	cellular	27	feb	412	5	-1	0	unknown	yes
41153	67	retired	divorced	tertiary	no	443	no	no	cellular	18	aug	441	1	-1	0	unknown	yes
32027	36	technician	married	secondary	no	15485	no	no	cellular	14	apr	461	1	-1	0	unknown	yes
26961	47	technician	married	secondary	no	0	no	no	cellular	21	nov	591	1	10	1	failure	yes
40152	30	technician	single	tertiary	no	0	yes	no	cellular	5	jun	159	2	-1	0	unknown	yes
30176	69	management	married	tertiary	no	840	no	no	telephone	5	feb	128	3	-1	0	unknown	yes
36173	40	services	married	secondary	no	-192	yes	no	cellular	11	may	666	1	-1	0	unknown	yes
32028	33	management	married	tertiary	no	0	yes	no	cellular	14	apr	535	3	328	1	failure	yes
36692	51	blue-collar	married	secondary	no	518	yes	no	cellular	12	may	918	1	-1	0	unknown	yes
44547	44	management	married	tertiary	no	1791	no	no	telephone	12	aug	201	1	182	2	success	yes
40862	32	technician	married	secondary	no	484	yes	no	cellular	12	aug	668	2	463	1	success	yes
44041	48	admin.	single	secondary	no	1544	yes	no	telephone	30	jun	263	1	450	2	failure	yes
21224	47	management	married	tertiary	no	682	no	no	cellular	18	aug	638	6	-1	0	unknown	yes
38732	32	blue-collar	single	secondary	no	217	yes	no	cellular	15	may	692	3	-1	0	unknown	yes
42785	54	admin.	married	secondary	no	0	no	no	cellular	28	jan	161	1	98	2	failure	yes
39972	34	technician	married	tertiary	no	127	no	no	cellular	3	jun	117	1	-1	0	unknown	yes

In [12]:
```
bank_new.tail(25)
```

Out[12]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
8935	39	blue-collar	married	primary	no	0	yes	no	unknown	4	jun	112	4	-1	0	unknown	no
36257	40	services	married	secondary	no	51	yes	no	cellular	11	may	50	1	-1	0	unknown	no
16162	32	management	married	tertiary	no	663	no	yes	cellular	22	jul	62	3	-1	0	unknown	no
37035	36	technician	married	tertiary	no	421	yes	no	cellular	13	may	793	5	-1	0	unknown	no
13823	33	technician	married	tertiary	no	1746	yes	no	cellular	10	jul	184	1	-1	0	unknown	no
8260	54	self-employed	married	primary	no	277	yes	no	unknown	2	jun	360	3	-1	0	unknown	no
28590	28	technician	married	tertiary	no	203	no	yes	cellular	29	jan	188	3	-1	0	unknown	no
12183	40	blue-collar	married	secondary	no	95	no	yes	unknown	20	jun	61	4	-1	0	unknown	no
17209	37	management	single	unknown	no	242	no	yes	cellular	28	jul	124	3	-1	0	unknown	no
39681	25	management	single	tertiary	no	430	no	yes	cellular	27	may	145	2	-1	0	unknown	no
27824	47	blue-collar	married	primary	no	1452	yes	yes	cellular	28	jan	181	1	177	2	failure	no
30958	46	services	married	secondary	no	692	yes	no	cellular	9	feb	388	3	257	2	failure	no
4985	31	admin.	single	tertiary	no	1583	yes	no	unknown	21	may	207	1	-1	0	unknown	no
24394	46	management	married	secondary	no	1306	yes	no	telephone	17	nov	155	1	-1	0	unknown	no
30421	32	services	married	secondary	no	182	no	no	cellular	5	feb	277	1	169	2	failure	no
10069	53	retired	single	secondary	no	1846	no	no	unknown	11	jun	95	1	-1	0	unknown	no
4289	34	admin.	married	secondary	no	645	yes	no	unknown	19	may	420	1	-1	0	unknown	no
39891	36	blue-collar	married	secondary	no	79	yes	yes	cellular	2	jun	74	2	-1	0	unknown	no
13713	40	blue-collar	married	primary	no	137	no	yes	cellular	10	jul	214	1	-1	0	unknown	no
24560	41	unemployed	married	primary	no	557	yes	no	cellular	17	nov	158	1	173	1	failure	no
27927	43	unknown	single	unknown	no	181	no	no	telephone	28	jan	41	1	-1	0	unknown	no
18014	40	management	divorced	tertiary	no	69	yes	no	cellular	30	jul	149	2	-1	0	unknown	no
35076	40	management	married	tertiary	no	429	yes	no	cellular	6	may	222	2	363	4	failure	no
11238	44	blue-collar	single	unknown	no	4330	no	no	unknown	18	jun	16	9	-1	0	unknown	no
4591	39	admin.	married	secondary	no	2019	yes	no	unknown	20	may	166	1	-1	0	unknown	no

Shuffling the dataset

In [13]:
```
from sklearn.utils import shuffle
bank_new=shuffle(bank_new)
bank_new.head(25)
```

Out[13]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
19808	48	entrepreneur	married	tertiary	no	0	no	yes	cellular	8	aug	85	9	-1	0	unknown	no
34151	32	management	single	tertiary	no	0	yes	no	cellular	30	apr	370	3	-1	0	unknown	yes
14560	45	admin.	married	secondary	no	0	no	yes	telephone	15	jul	533	3	-1	0	unknown	yes
33948	24	management	single	tertiary	no	2845	no	no	cellular	30	apr	779	2	-1	0	unknown	yes
41489	33	self-employed	married	secondary	no	409	no	no	cellular	8	sep	165	1	97	4	success	yes
20190	35	technician	single	tertiary	no	272	no	no	cellular	11	aug	86	2	-1	0	unknown	no
39232	33	management	single	secondary	no	219	yes	no	cellular	18	may	19	4	-1	0	unknown	no
17352	41	admin.	married	secondary	no	239	yes	no	cellular	28	jul	76	12	-1	0	unknown	no
6304	33	admin.	divorced	secondary	no	479	yes	no	unknown	27	may	74	5	-1	0	unknown	no
4790	42	blue-collar	married	primary	no	714	yes	no	unknown	21	may	18	28	-1	0	unknown	no
42580	30	self-employed	single	tertiary	no	916	no	no	cellular	29	dec	449	2	-1	0	unknown	yes
40514	49	housemaid	married	primary	no	889	no	no	telephone	7	jul	388	1	-1	0	unknown	yes
33921	44	blue-collar	married	unknown	no	1529	yes	no	cellular	30	apr	347	1	-1	0	unknown	yes
6540	33	management	married	tertiary	no	1657	yes	no	unknown	27	may	342	4	-1	0	unknown	no
43157	52	technician	married	secondary	no	117	no	no	cellular	26	feb	959	3	186	6	success	yes
9352	28	blue-collar	married	secondary	no	708	yes	no	unknown	6	jun	339	5	-1	0	unknown	no
18377	35	services	single	secondary	no	1742	yes	no	cellular	31	jul	39	2	-1	0	unknown	no
20236	33	management	single	tertiary	no	0	no	no	cellular	11	aug	699	7	-1	0	unknown	yes
15452	39	management	married	primary	no	738	yes	yes	cellular	18	jul	215	3	-1	0	unknown	no
19341	31	technician	single	secondary	no	200	no	no	cellular	6	aug	315	2	-1	0	unknown	no
11715	48	management	married	tertiary	no	5320	yes	no	unknown	20	jun	792	1	-1	0	unknown	yes
43776	49	self-employed	divorced	tertiary	no	3293	no	no	cellular	24	may	260	1	77	9	failure	no
19515	57	retired	married	secondary	no	283	no	no	cellular	7	aug	123	2	-1	0	unknown	no
32649	34	technician	married	secondary	no	1641	yes	no	cellular	17	apr	380	1	-1	0	unknown	no
30989	33	technician	single	secondary	no	0	yes	no	cellular	9	feb	91	3	236	4	other	no

Splitting into target and Features

In [14]:
```
y=bank_new['y']
y.shape
```

Out[14]: (40000,)

In [15]:
```
X=bank_new.drop(['y'],axis=1)
X.shape
```

Out[15]: (40000, 16)

In [16]:
```
X
```

Out[16]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome
19808	48	entrepreneur	married	tertiary	no	0	no	yes	cellular	8	aug	85	9	-1	0	unknown
34151	32	management	single	tertiary	no	0	yes	no	cellular	30	apr	370	3	-1	0	unknown
14560	45	admin.	married	secondary	no	0	no	yes	telephone	15	jul	533	3	-1	0	unknown
33948	24	management	single	tertiary	no	2845	no	no	cellular	30	apr	779	2	-1	0	unknown
41489	33	self-employed	married	secondary	no	409	no	no	cellular	8	sep	165	1	97	4	success
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
42616	65	unknown	married	unknown	no	300	no	no	cellular	12	jan	105	1	-1	0	unknown
36775	35	admin.	single	secondary	no	10	no	yes	cellular	12	may	357	1	175	4	failure
17103	45	blue-collar	single	primary	no	92	no	yes	cellular	25	jul	99	7	-1	0	unknown
38642	33	management	married	secondary	no	1423	yes	no	cellular	15	may	333	2	364	3	failure
1236	34	entrepreneur	married	tertiary	no	10350	yes	no	unknown	8	may	187	3	-1	0	unknown

40000 rows × 16 columns

Converting categorical features to numeric

In [17]:
```
X_new=pd.get_dummies(X)
X_new.shape
```

Out[17]:
```
(40000, 51)
```

In [18]:
```
X_new
```

Out[18]:

	age	balance	day	duration	campaign	pdays	previous	job_admin.	job_blue-collar	job_entrepreneur	...	month_jun	month_mar	month_may	month_nov
19808	48	0	8	85	9	-1	0	0	0	1	...	0	0	0	0
34151	32	0	30	370	3	-1	0	0	0	0	...	0	0	0	0
14560	45	0	15	533	3	-1	0	1	0	0	...	0	0	0	0
33948	24	2845	30	779	2	-1	0	0	0	0	...	0	0	0	0
41489	33	409	8	165	1	97	4	0	0	0	...	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
42616	65	300	12	105	1	-1	0	0	0	0	...	0	0	0	0
36775	35	10	12	357	1	175	4	1	0	0	...	0	0	1	0
17103	45	92	25	99	7	-1	0	0	1	0	...	0	0	0	0
38642	33	1423	15	333	2	364	3	0	0	0	...	0	0	1	0
1236	34	10350	8	187	3	-1	0	0	0	1	...	0	0	1	0

40000 rows × 51 columns

Standardisation of features

In [20]:
```
from sklearn.preprocessing import StandardScaler

scaler=StandardScaler()

X_scaled=scaler.fit_transform(X_new)
X_scaled
```

Out[20]:
```
array([[ 0.5953087 , -0.45017026, -0.91147011, ..., -0.21921049,
        -0.28888227,  0.55166443],
       [-0.79328786, -0.45017026,  1.70869792, ..., -0.21921049,
        -0.28888227,  0.55166443],
       [ 0.33494685, -0.45017026, -0.07778028, ..., -0.21921049,
        -0.28888227,  0.55166443],
       ...,
       [ 0.33494685, -0.42277614,  1.11320519, ..., -0.21921049,
        -0.28888227,  0.55166443],
       [-0.70650057, -0.02645463, -0.07778028, ..., -0.21921049,
        -0.28888227, -1.81269616],
       [-0.61971329,  2.63166865, -0.91147011, ..., -0.21921049,
        -0.28888227,  0.55166443]])
```

Splitting into train and test

In [21]:
```
from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test=train_test_split(X_scaled,y,test_size=0.2, random_state=100)
X_train.shape,X_test.shape,y_train.shape,y_test.shape
```

Out[21]:
```
((32000, 51), (8000, 51), (32000,), (8000,))
```

Model building- K Nearest Neighbor

from sklearn.neighbors import KNeighborsClassifier

knn=KNeighborsClassifier()

knn.fit(X_train, y_train)

Model performance

In [23]:
```
from sklearn.metrics import confusion_matrix,classification_report,roc_curve, roc_auc_score

cm=confusion_matrix(y_test, knn.predict(X_test))
report=classification_report(y_test, knn.predict(X_test))

print('The CM:\n', cm)
print('The report:\n',report)
```

```
The CM:
 [[4338  633]
 [ 601 2428]]
The report:
               precision    recall  f1-score   support

          no       0.88      0.87      0.88      4971
         yes       0.79      0.80      0.80      3029

    accuracy                           0.85      8000
   macro avg       0.84      0.84      0.84      8000
weighted avg       0.85      0.85      0.85      8000
```

Hyper parameter tuning using GridSearchCV

In [24]:
```
from sklearn.model_selection import GridSearchCV

knn_gs=GridSearchCV(knn,{'n_neighbors':range(3,8)})

knn_gs.fit(X_train,y_train)
```

Out[24]:
```
GridSearchCV(estimator=KNeighborsClassifier(),
             param_grid={'n_neighbors': range(3, 8)})
```

In [25]:
```
knn_gs.best_params_
```

Out[25]:
```
{'n_neighbors': 3}
```

Create the best knn model

In [27]:
```
knn_best=KNeighborsClassifier(n_neighbors=3)
knn_best.fit(X_train,y_train)
```

Out[27]:
```
KNeighborsClassifier(n_neighbors=3)
```

In [28]:
```
report=classification_report(y_test,knn_best.predict(X_test))

cm = confusion_matrix(y_test,knn_best.predict(X_test))


print(' The new report:\n',report)

print(' The new CM:\n', cm)
```

```
 The new report:
               precision    recall  f1-score   support

          no       0.92      0.87      0.90      4971
         yes       0.81      0.88      0.84      3029

    accuracy                           0.87      8000
   macro avg       0.86      0.87      0.87      8000
weighted avg       0.88      0.87      0.88      8000

 The new CM:
 [[4348  623]
 [ 378 2651]]
```

In [ ]: