

Vaibhav Kumar

Roll no : 19

## SVM Bank dataset

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
```

```
In [2]: bank=pd.read_csv('D:\\vk\\TRIM 3\\ML\\DATASET\\bank.csv',sep=';')
```

```
In [3]: bank
```

```
Out[3]:
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month
<b>0</b>	30	unemployed	married	primary	no	1787	no	no	cellular	19	oc
<b>1</b>	33	services	married	secondary	no	4789	yes	yes	cellular	11	may
<b>2</b>	35	management	single	tertiary	no	1350	yes	no	cellular	16	apr
<b>3</b>	30	management	married	tertiary	no	1476	yes	yes	unknown	3	jul
<b>4</b>	59	blue-collar	married	secondary	no	0	yes	no	unknown	5	may
...	...	...	...	...	...	...	...	...	...	...	..
<b>4516</b>	33	services	married	secondary	no	-333	yes	no	cellular	30	ju
<b>4517</b>	57	self-employed	married	tertiary	yes	-3313	yes	yes	unknown	9	may
<b>4518</b>	57	technician	married	secondary	no	295	no	no	cellular	19	aug
<b>4519</b>	28	blue-collar	married	secondary	no	1137	no	no	cellular	6	feb
<b>4520</b>	44	entrepreneur	single	tertiary	no	1136	yes	yes	cellular	3	apr

4521 rows × 17 columns

```
In [4]: bank.shape
```

```
Out[4]: (4521, 17)
```

```
In [5]: bank.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4521 entries, 0 to 4520
Data columns (total 17 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   age         4521 non-null    int64  
 1   job          4521 non-null    object  
 2   marital      4521 non-null    object  
 3   education    4521 non-null    object  
 4   default      4521 non-null    object  
 5   balance      4521 non-null    int64  
 6   housing      4521 non-null    object  
 7   loan          4521 non-null    object  
 8   contact       4521 non-null    object  
 9   day           4521 non-null    int64  
 10  month         4521 non-null    object  
 11  duration     4521 non-null    int64  
 12  campaign     4521 non-null    int64  
 13  pdays         4521 non-null    int64  
 14  previous     4521 non-null    int64  
 15  poutcome     4521 non-null    object  
 16  y             4521 non-null    object  
dtypes: int64(7), object(10)
memory usage: 600.6+ KB
```

In [6]: `bank.head()`

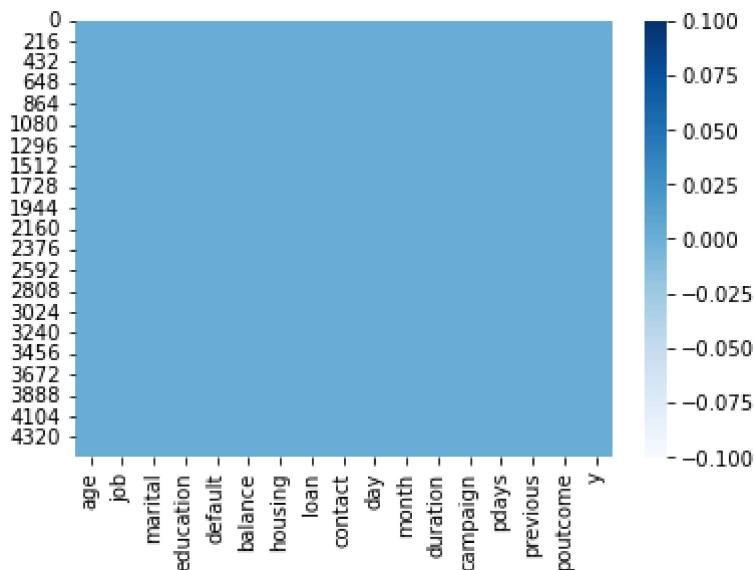
	age	job	marital	education	default	balance	housing	loan	contact	day	month	d
0	30	unemployed	married	primary	no	1787	no	no	cellular	19	oct	
1	33	services	married	secondary	no	4789	yes	yes	cellular	11	may	
2	35	management	single	tertiary	no	1350	yes	no	cellular	16	apr	
3	30	management	married	tertiary	no	1476	yes	yes	unknown	3	jun	
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	5	may	

In [7]: `bank.describe()`

	age	balance	day	duration	campaign	pdays	previous
<b>count</b>	4521.000000	4521.000000	4521.000000	4521.000000	4521.000000	4521.000000	4521.000000
<b>mean</b>	41.170095	1422.657819	15.915284	263.961292	2.793630	39.766645	0.542579
<b>std</b>	10.576211	3009.638142	8.247667	259.856633	3.109807	100.121124	1.693562
<b>min</b>	19.000000	-3313.000000	1.000000	4.000000	1.000000	-1.000000	0.000000
<b>25%</b>	33.000000	69.000000	9.000000	104.000000	1.000000	-1.000000	0.000000
<b>50%</b>	39.000000	444.000000	16.000000	185.000000	2.000000	-1.000000	0.000000
<b>75%</b>	49.000000	1480.000000	21.000000	329.000000	3.000000	-1.000000	0.000000
<b>max</b>	87.000000	71188.000000	31.000000	3025.000000	50.000000	871.000000	25.000000

In [8]: `sns.heatmap(bank.isnull(), cmap='Blues')`

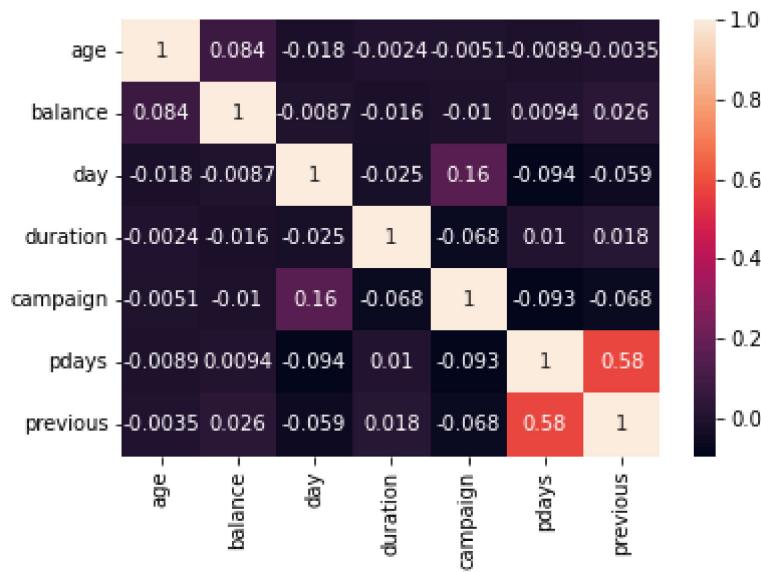
Out[8]: <AxesSubplot:>



In [9]: `#to find corelattion  
cor=bank.corr()`

In [10]: `sns.heatmap(cor, annot=True)`

Out[10]: <AxesSubplot:>



**pdays and previous are highly corelated with value 0.58**

**day and campaign are also corelated with 0.16**

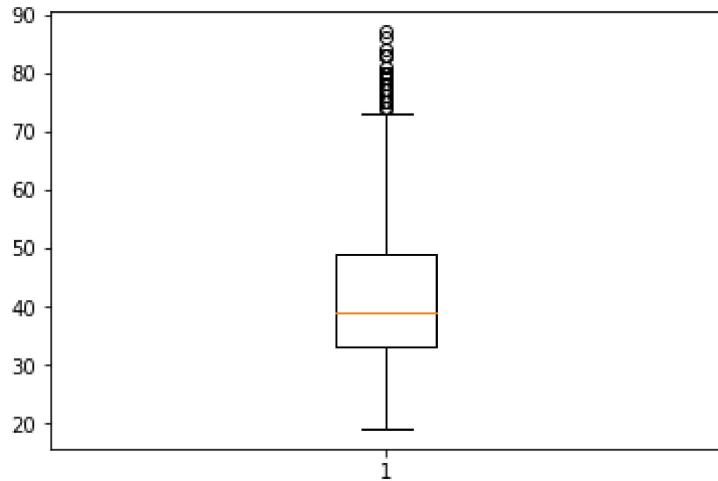
Numeric : age,balance,day,duration,campaign,pdays

categorical:

Ordinal:poutcome,default,housing,loan (using map)

nominal : job,marital,education ,contact,month (get dummies)

In [11]: `plt.boxplot(bank['age']);`

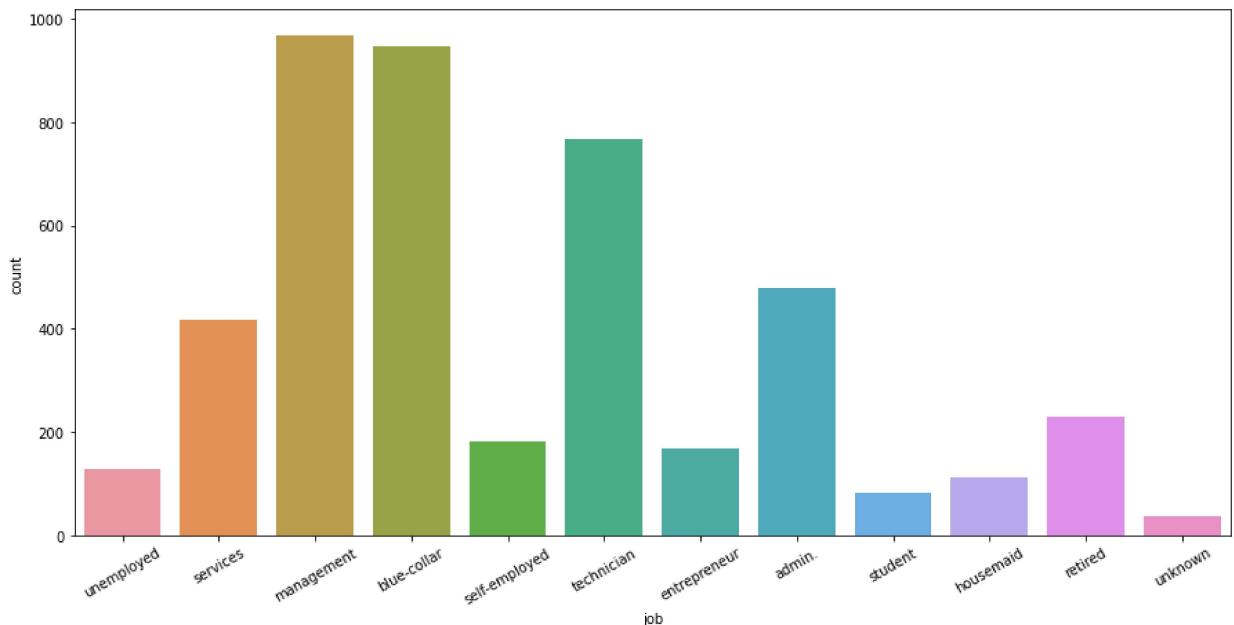


**min age is 20 and max age is 70-90**

In [12]: `bank['job'].unique()`

Out[12]: `array(['unemployed', 'services', 'management', 'blue-collar', 'self-employed', 'technician', 'entrepreneur', 'admin.', 'student', 'housemaid', 'retired', 'unknown'], dtype=object)`

In [13]: `plt.figure(figsize=(15,7))
ax=sns.countplot(x='job',data=bank)
ax.set_xticklabels(ax.get_xticklabels(),rotation=30);`



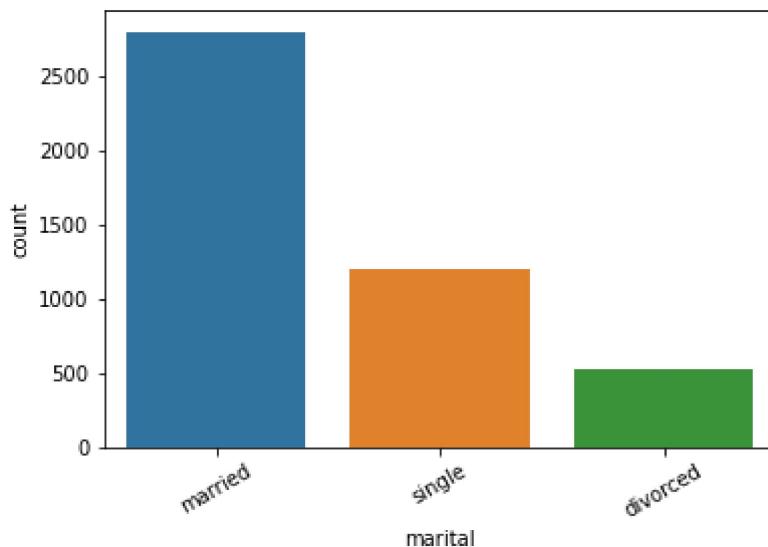
**most of the customers are from management,blue collar and technician**

**less from housemaid, students ,unknown and unemployed**

In [14]: `bank['marital'].unique()`

```
In [14]: array(['married', 'single', 'divorced'], dtype=object)
```

```
In [15]: ax=sns.countplot(x='marital',data=bank)
ax.set_xticklabels(ax.get_xticklabels(),rotation = 30);
```

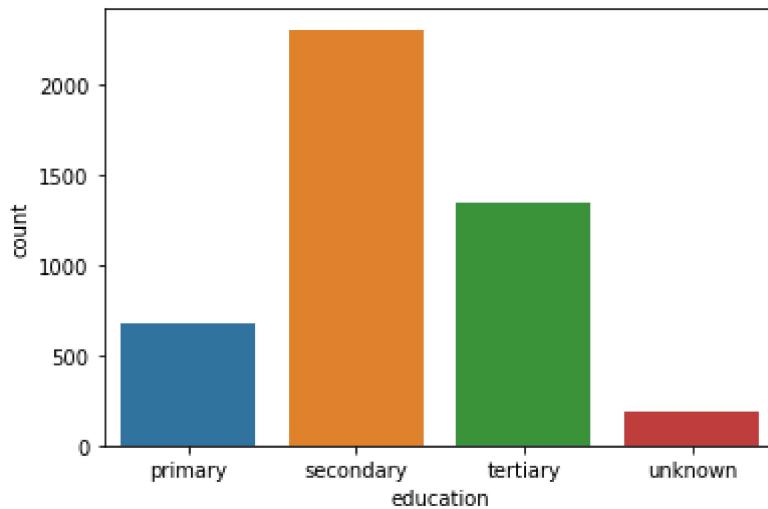


**most are married**

```
In [16]: bank['education'].unique()
```

```
Out[16]: array(['primary', 'secondary', 'tertiary', 'unknown'], dtype=object)
```

```
In [17]: ax=sns.countplot(x='education',data=bank)
```

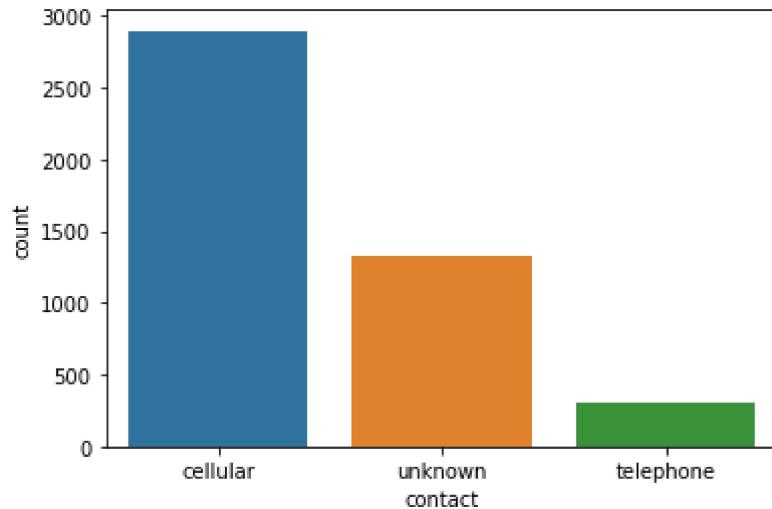


**most of them are educated secondary and teriary**

```
In [18]: bank['contact'].unique()
```

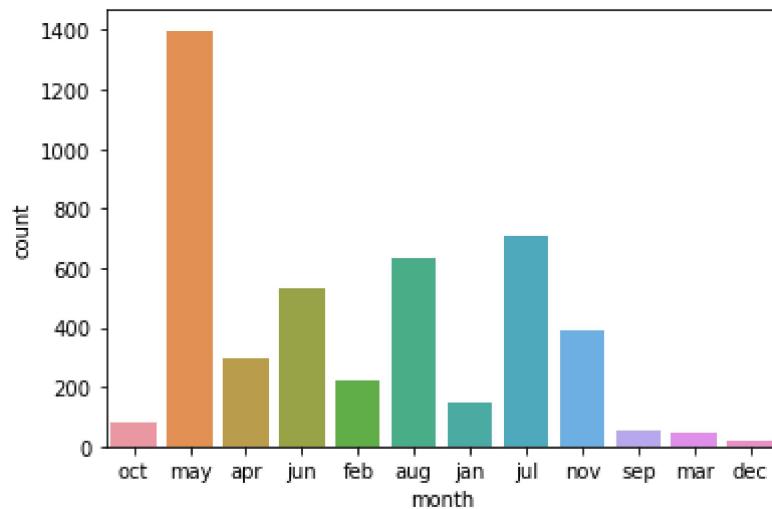
```
Out[18]: array(['cellular', 'unknown', 'telephone'], dtype=object)
```

```
In [19]: ax=sns.countplot(x='contact',data=bank)
```



**most contact are made on cellular network and less on telephone**

```
In [20]: ax=sns.countplot(x='month',data=bank)
```

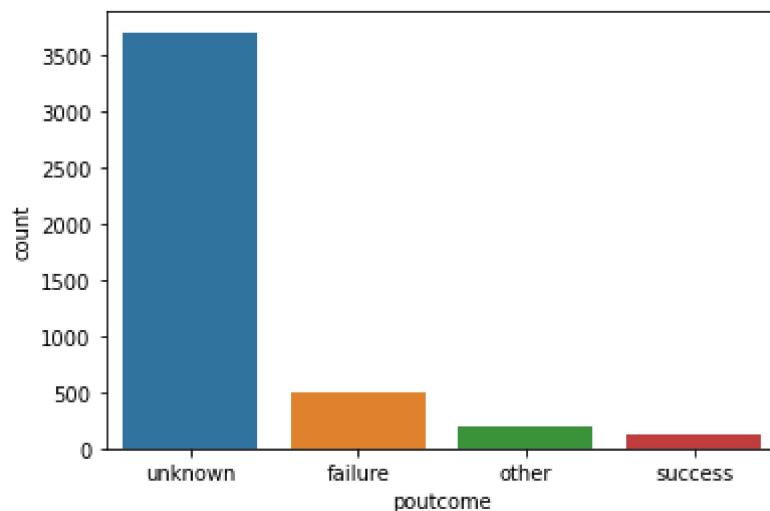


**may month has most count and december has less count**

```
In [21]: bank['poutcome'].unique()
```

```
Out[21]: array(['unknown', 'failure', 'other', 'success'], dtype=object)
```

```
In [22]: ax=sns.countplot(x='poutcome',data=bank)
```



**most of them are unknown**

```
In [23]: bank['poutcome']=bank['poutcome'].map({'failure':-1,'unknown':0,'success':1,'other':0})
```

```
In [24]: bank['default'].unique()
```

```
Out[24]: array(['no', 'yes'], dtype=object)
```

```
In [25]: bank['housing'].unique()
```

```
Out[25]: array(['no', 'yes'], dtype=object)
```

```
In [26]: bank['loan'].unique()
```

```
Out[26]: array(['no', 'yes'], dtype=object)
```

```
In [27]: bank['default']=bank['default'].map({'yes':0,'no':1})  
bank['housing']=bank['housing'].map({'yes':0,'no':1})  
bank['loan']=bank['loan'].map({'yes':0,'no':1})
```

```
In [28]: bank
```

Out[28]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month
<b>0</b>	30	unemployed	married	primary	1	1787	1	1	cellular	19	oct
<b>1</b>	33	services	married	secondary	1	4789	0	0	cellular	11	may
<b>2</b>	35	management	single	tertiary	1	1350	0	1	cellular	16	apr
<b>3</b>	30	management	married	tertiary	1	1476	0	0	unknown	3	jul
<b>4</b>	59	blue-collar	married	secondary	1	0	0	1	unknown	5	may
...	...	...	...	...	...	...	...	...	...	...	..
<b>4516</b>	33	services	married	secondary	1	-333	0	1	cellular	30	ju
<b>4517</b>	57	self-employed	married	tertiary	0	-3313	0	0	unknown	9	may
<b>4518</b>	57	technician	married	secondary	1	295	1	1	cellular	19	aug
<b>4519</b>	28	blue-collar	married	secondary	1	1137	1	1	cellular	6	feb
<b>4520</b>	44	entrepreneur	single	tertiary	1	1136	0	0	cellular	3	apr

4521 rows × 17 columns

In [29]: `nominal=['job', 'marital', 'education', 'contact', 'month']  
data=pd.get_dummies(bank,columns=nominal)`

In [30]: `data.shape`

Out[30]: (4521, 46)

In [31]: `data.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4521 entries, 0 to 4520
Data columns (total 46 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   age              4521 non-null    int64  
 1   default          4521 non-null    int64  
 2   balance           4521 non-null    int64  
 3   housing           4521 non-null    int64  
 4   loan              4521 non-null    int64  
 5   day               4521 non-null    int64  
 6   duration          4521 non-null    int64  
 7   campaign          4521 non-null    int64  
 8   pdays             4521 non-null    int64  
 9   previous          4521 non-null    int64  
 10  poutcome          4521 non-null    int64  
 11  y                 4521 non-null    object 
 12  job_admin.        4521 non-null    uint8  
 13  job_blue-collar  4521 non-null    uint8  
 14  job_entrepreneur 4521 non-null    uint8  
 15  job_housemaid    4521 non-null    uint8  
 16  job_management   4521 non-null    uint8  
 17  job_retired       4521 non-null    uint8  
 18  job_self-employed 4521 non-null    uint8  
 19  job_services      4521 non-null    uint8  
 20  job_student       4521 non-null    uint8  
 21  job_technician   4521 non-null    uint8  
 22  job_unemployed   4521 non-null    uint8  
 23  job_unknown       4521 non-null    uint8  
 24  marital_divorced 4521 non-null    uint8  
 25  marital_married   4521 non-null    uint8  
 26  marital_single    4521 non-null    uint8  
 27  education_primary 4521 non-null    uint8  
 28  education_secondary 4521 non-null    uint8  
 29  education_tertiary 4521 non-null    uint8  
 30  education_unknown 4521 non-null    uint8  
 31  contact_cellular  4521 non-null    uint8  
 32  contact_telephone 4521 non-null    uint8  
 33  contact_unknown   4521 non-null    uint8  
 34  month_apr         4521 non-null    uint8  
 35  month_aug         4521 non-null    uint8  
 36  month_dec         4521 non-null    uint8  
 37  month_feb         4521 non-null    uint8  
 38  month_jan         4521 non-null    uint8  
 39  month_jul         4521 non-null    uint8  
 40  month_jun         4521 non-null    uint8  
 41  month_mar         4521 non-null    uint8  
 42  month_may         4521 non-null    uint8  
 43  month_nov         4521 non-null    uint8  
 44  month_oct         4521 non-null    uint8  
 45  month_sep         4521 non-null    uint8  
dtypes: int64(11), object(1), uint8(34)
memory usage: 574.1+ KB
```

In [32]: `data['y']=data['y'].map({'yes':1,'no':0})`

In [33]: `data.head()`

	age	default	balance	housing	loan	day	duration	campaign	pdays	previous	...	month_dec
0	30	1	1787	1	1	19	79	1	-1	0	...	0
1	33	1	4789	0	0	11	220	1	339	4	...	0
2	35	1	1350	0	1	16	185	1	330	1	...	0
3	30	1	1476	0	0	3	199	4	-1	0	...	0
4	59	1	0	0	1	5	226	1	-1	0	...	0

5 rows × 46 columns

In [34]: `data[['age','balance','day','campaign','pdays']] = StandardScaler().fit_transform(data[['age','balance','day','campaign','pdays'])`

In [35]: `data.shape`

Out[35]: (4521, 46)

In [36]: `data.drop('duration',axis=1,inplace=True)`

In [37]: `data.shape`

Out[37]: (4521, 45)

In [38]: `op = data.pop('y')`

In [39]: `op.shape`

Out[39]: (4521,)

In [40]: `xtr,xts,ytr,yts = train_test_split(data,op,train_size=0.8,random_state=100)`

In [41]: `xtr.shape`

Out[41]: (3616, 44)

In [42]: `ytr.shape`

Out[42]: (3616,)

In [43]: `yts.shape`

Out[43]: (905,)

In [44]: `xts.shape`

Out[44]: (905, 44)