Numerosity Data Reduction

Random sampling

Example – Random sampling to speed up tuning

In [46]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

In [47]:
```
customer_df = pd.read_csv('Customer Churn.csv')
print(customer_df.shape)
print(customer_df.Churn.value_counts())
```

```
(3150, 9)
0    2655
1     495
Name: Churn, dtype: int64
```

In [48]:
```
customer_df_rs = customer_df.sample(1000,random_state=1)
y=customer_df_rs['Churn']
Xs = customer_df_rs.drop(columns=['Churn'])
print(customer_df_rs.shape)
```

```
(1000, 9)
```

In [49]:
```
print(customer_df_rs.Churn.value_counts())
```

```
0    856
1    144
Name: Churn, dtype: int64
```

Stratified sampling

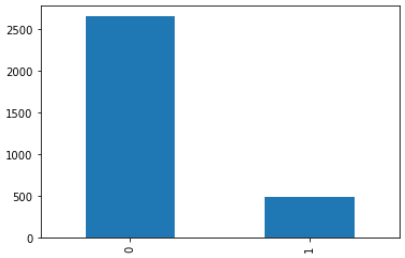
Example – Stratified sampling for imbalanced dataset

In [59]:
```
n,s=len(customer_df),1000
print(n,s)
r = s/n
print('Ratio of each Churn class in sample:',r)
sample_df = customer_df.groupby('Churn').apply(lambda sdf: sdf.sample(round(len(sdf)*r)))
print(sample_df.Churn.value_counts())
```

```
3150 1000
Ratio of each Churn class in sample: 0.31746031746031744
0    843
1    157
Name: Churn, dtype: int64
```

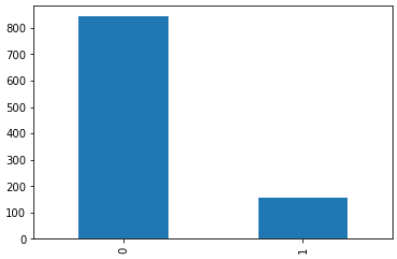
In [60]:
```
customer_df.Churn.value_counts().plot.bar()
```

Out[60]: <AxesSubplot:>



In [61]:
```
sample_df.Churn.value_counts().plot.bar()
```

Out[61]: <AxesSubplot:>



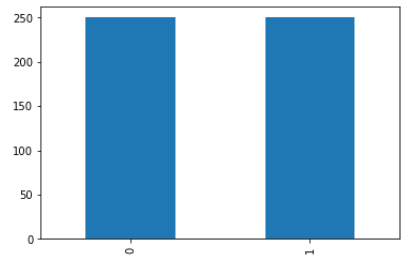
Random Over/Under-sampling

In [65]:
```
n,s=len(customer_df),500
sample_df = customer_df.groupby('Churn').apply(lambda sdf: sdf.sample(250))
print(sample_df.Churn.value_counts())
```

```
0    250
1    250
Name: Churn, dtype: int64
```

In [66]:
```
sample_df.Churn.value_counts().plot.bar()
```

Out[66]: <AxesSubplot:>



In [67]:
```
sample_df
```

Out[67]:

		Call Failure	Complains	Subscription Length	Seconds of Use	Frequency of use	Frequency of SMS	Distinct Called Numbers	Status	Churn
Churn										
0	**1815**	0	0	30	6195	74	171	23	1	0
	410	4	0	35	5738	87	0	7	1	0
	1455	4	0	33	3290	68	14	21	1	0
	2086	0	0	33	1360	38	31	18	0	0
	506	0	0	38	1760	29	264	3	1	0
...	**...**	...	...	...	...	...	...	...	...	...
1	**2178**	8	0	40	498	11	12	6	1	1
	1599	0	0	7	0	0	0	0	1	1
	1476	2	1	30	2505	27	0	9	0	1
	1382	0	1	28	0	0	0	0	1	1
	368	2	0	40	180	8	11	5	0	1

500 rows × 9 columns