

```
In [1]: import numpy as np
import pandas as pd
```

```
In [3]: # oversampling code
# np.random.randint(1, 151, 700)
```

```
In [ ]:
```

```
In [13]: ! pip install imbalanced-learn
```

```
Requirement already satisfied: imbalanced-learn in /Users/mohit/opt/anaconda3/lib/python3.8/site-packages (0.9.1)
Requirement already satisfied: joblib>=1.0.0 in /Users/mohit/opt/anaconda3/lib/python3.8/site-packages (from imbalanced-learn) (1.0.1)
Requirement already satisfied: scipy>=1.3.2 in /Users/mohit/opt/anaconda3/lib/python3.8/site-packages (from imbalanced-learn) (1.6.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /Users/mohit/opt/anaconda3/lib/python3.8/site-packages (from imbalanced-learn) (2.1.0)
Requirement already satisfied: numpy>=1.17.3 in /Users/mohit/opt/anaconda3/lib/python3.8/site-packages (from imbalanced-learn) (1.20.1)
Requirement already satisfied: scikit-learn>=1.1.0 in /Users/mohit/opt/anaconda3/lib/python3.8/site-packages (from imbalanced-learn) (1.1.2)
```

```
In [14]: churn = pd.read_csv('Churn.csv')
```

```
In [15]: churn.shape
```

```
Out[15]: (3333, 21)
```

```
In [17]: churn.head()
```

```
Out[17]:
```

	Account Length	VMail Message	Day Mins	Eve Mins	Night Mins	Intl Mins	CustServ Calls	Churn	Intl Plan	VMail Plan	...	Day Charge	Eve Calls	Eve Charge	Night Calls	Night Charge	Intl Calls	Intl Charge	State	A C
0	128	25	265.1	197.4	244.7	10.0	1	0	0	1	...	45.07	99	16.78	91	11.01	3	2.70	KS	
1	107	26	161.6	195.5	254.4	13.7	1	0	0	1	...	27.47	103	16.62	103	11.45	3	3.70	OH	
2	137	0	243.4	121.2	162.6	12.2	0	0	0	0	...	41.38	110	10.30	104	7.32	5	3.29	NJ	
3	84	0	299.4	61.9	196.9	6.6	2	0	1	0	...	50.90	88	5.26	89	8.86	7	1.78	OH	
4	75	0	166.7	148.3	186.9	10.1	3	0	1	0	...	28.34	122	12.61	121	8.41	3	2.73	OK	

5 rows × 21 columns

```
In [36]: churn['Churn'].value_counts(normalize=True)
```

```
Out[36]: 0    0.855086
         1    0.144914
         Name: Churn, dtype: float64
```

```
In [ ]:
```

```
In [19]: X = churn[['VMail Message', 'CustServ Calls', 'Eve Mins']].values
         y = churn['Churn'].values
```

```
In [21]: from sklearn.neighbors import KNeighborsClassifier
        from sklearn.model_selection import cross_val_score
```

```
In [25]: knn = KNeighborsClassifier(n_neighbors=5)
```

```
In [39]: res = cross_val_score(knn, X, y, cv = 5, scoring='recall')
        res.mean()
```

```
Out[39]: 0.08898195876288659
```

```
In [ ]:
```

SMOTE

```
In [52]: from imblearn.over_sampling import SMOTE
```

```
In [29]: sm = SMOTE()
        X_sm, y_sm = sm.fit_resample(X, y)
```

```
In [30]: X_sm.shape
```

```
Out[30]: (5700, 3)
```

```
In [32]: pd.Series(y_sm).value_counts()
```

```
Out[32]: 0    2850
        1    2850
        dtype: int64
```

```
In [50]: knn2 = KNeighborsClassifier(n_neighbors=5)
```

```
In [51]: res2 = cross_val_score(knn2, X_sm, y_sm, cv = 5, scoring='f1')  
res2.mean()
```

```
Out[51]: 0.7753788881280601
```

```
In [ ]:
```

```
In [ ]:
```

Entropy

```
In [55]: -(0.5*np.log2(0.5) + 0.5*np.log2(0.5))
```

```
Out[55]: 1.0
```

```
In [56]: -(0.3*np.log2(0.3) + 0.7*np.log2(0.7))
```

```
Out[56]: 0.8812908992306927
```

```
In [ ]:
```