IMBALANCED DATA HANDLING

BASIC TECHNIQUES

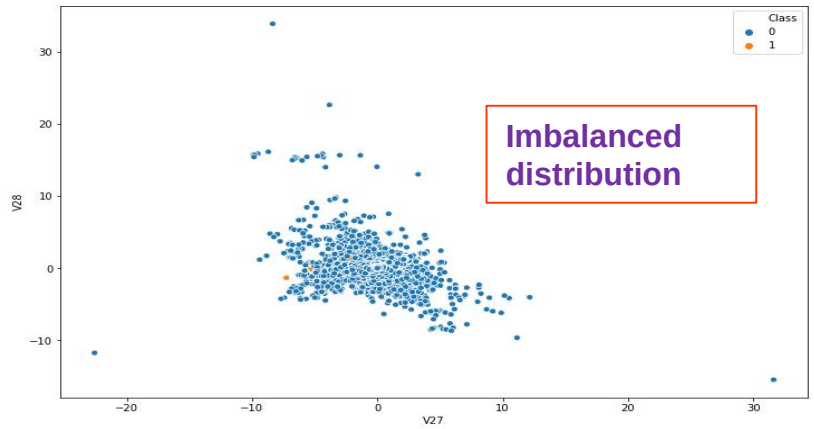
BY CHIRANJIT PATHAK

IMBALANCED DATASET: CREDIT CARD FRAUD

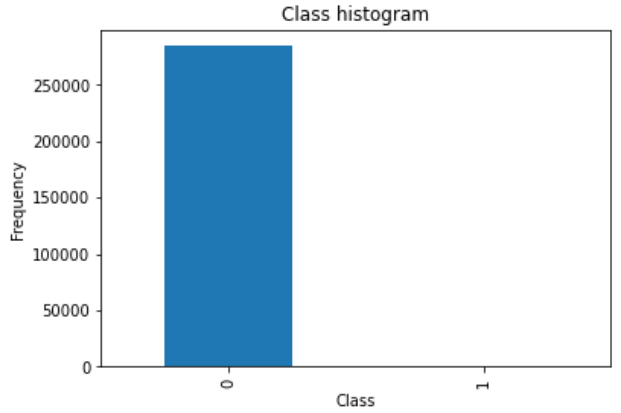
Imbalanced Data Distribution, generally happens when observations in one of the class are much higher or lower than the other classes. As Machine Learning algorithms tend to increase accuracy by reducing the error, they do not consider the class distribution. This problem is prevalent in examples such as Fraud Detection, Anomaly Detection, Facial recognition etc.

The challenge of working with imbalanced datasets is that most machine learning techniques will ignore, and in turn have poor performance on, the minority class, although typically it is performance on the minority class that is most important.

In [2]:
```
data = pd.read_csv('../input/creditcard.csv')
data.head(3)
```



Out[28]:
```
0    284315
1       492
Name: Class, dtype: int64
```



Imbalanced Data Handling Techniques:

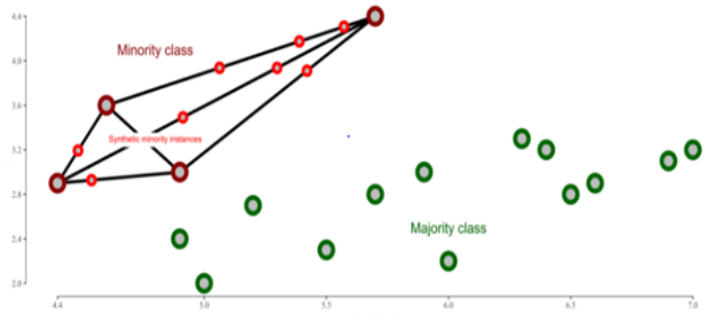
There are mainly 2 algorithms that are widely used for handling imbalanced class distribution.

- Oversampling : Populate Minority to match majority → SMOTE & Random Over Sampler
- Under sampling : Deduct majority class to match minority → Near Miss

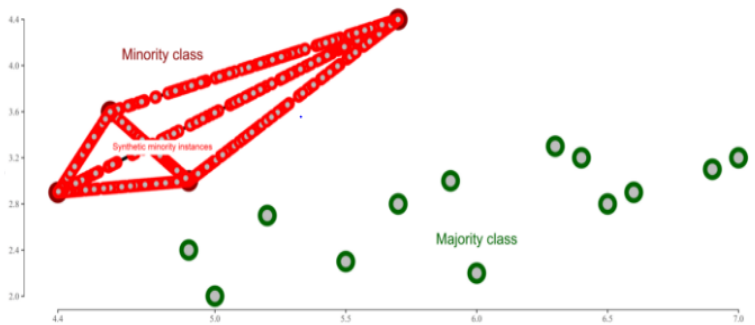
SMOTE (Synthetic Minority Oversampling TEchnique)

SMOTE synthesizes new minority instances between existing minority instances. It generates the virtual training records by linear interpolation for the minority class. These synthetic training records are generated by randomly selecting one or more of the k-nearest neighbors for each example in the minority class. Random Over Sampler increasing minority class examples by replicating them

SMOTE will synthetically generate new instances along these lines which would result into increase in percentage of minority class in comparison to majority class.



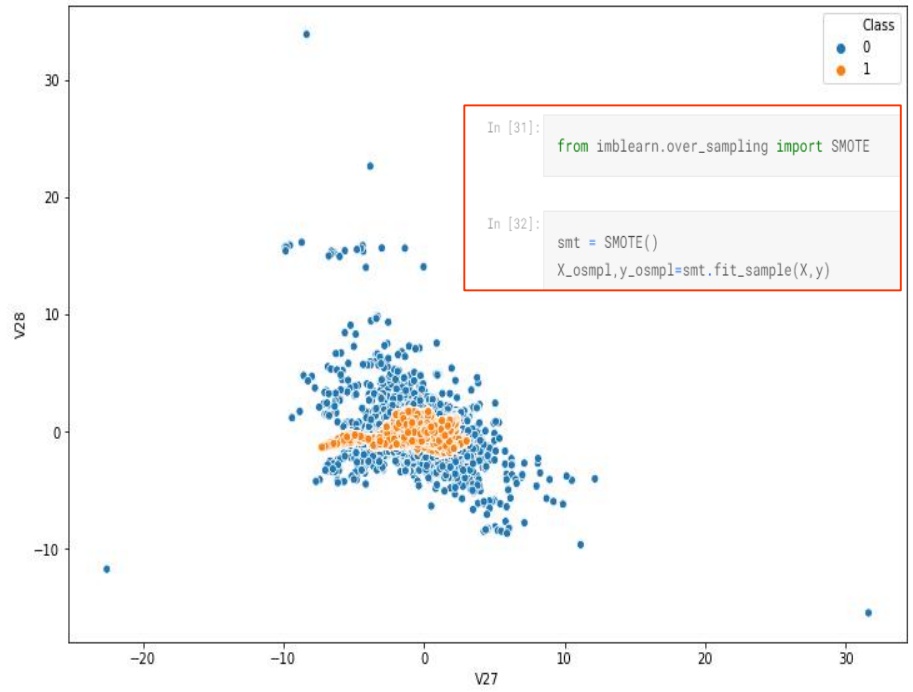
Once we run SOMTE () function for appropriate values of "K" and "dup_size" the sufficient number of synthetic minority instances generated would solve the problem of data imbalance.

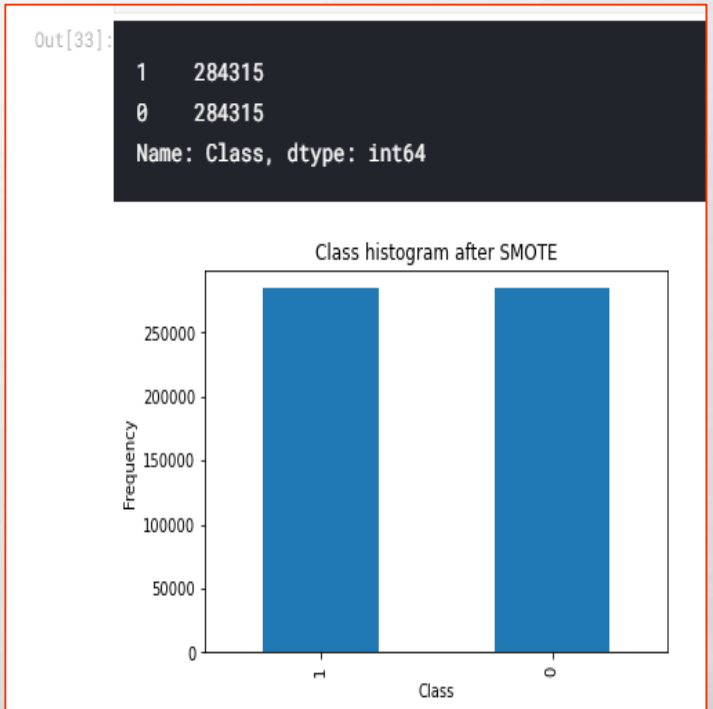


SMOTE:

SMOTE (Synthetic Minority Oversampling TEchnique)

SMOTE synthetically generates new minority instances between existing instances. The new instances created are not just the copy of existing minority cases instead; the algorithm takes sample of feature space for each target class and its neighbors and then generates new instances that combine the features of the target cases with features of its neighbors.

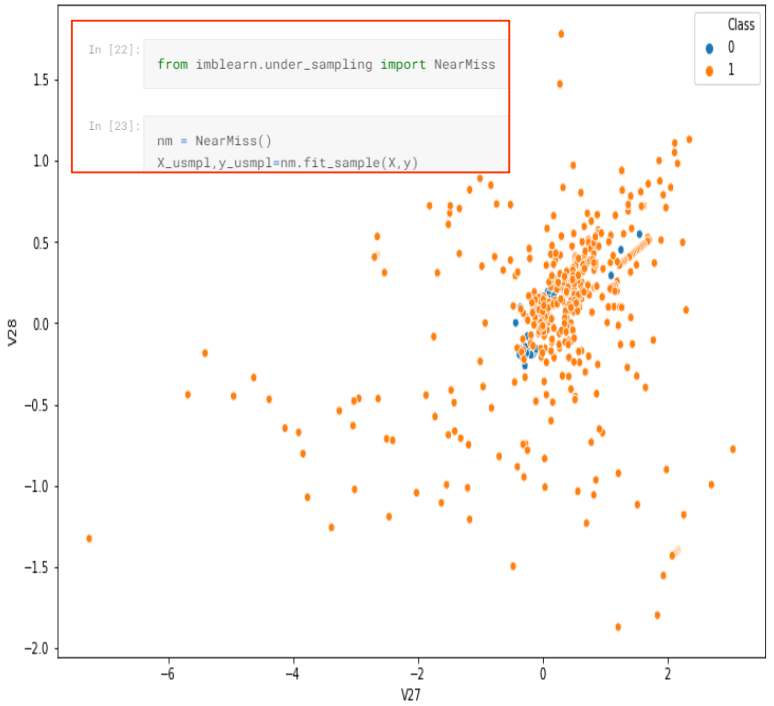


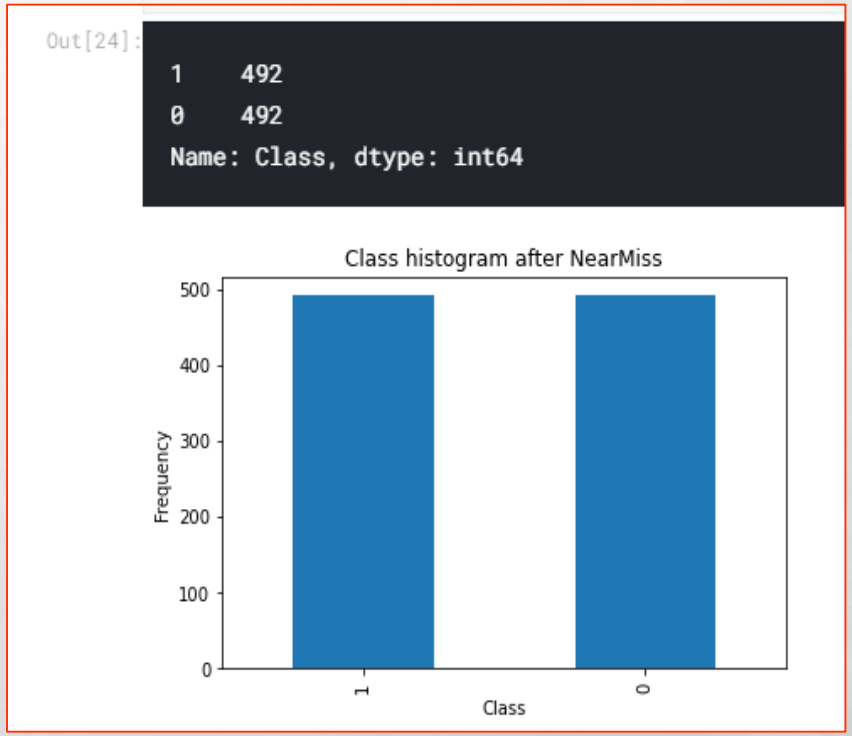


Near Miss:

Near Miss Algorithm

It aims to balance class distribution by randomly eliminating majority class examples. When instances of two different classes are very close to each other, we remove the instances of the majority class to increase the spaces between the two classes. This helps in the classification process.





COMPARISON:

Techniques	Advantages	Short comings
Oversampling: SMOTE	• Easy to fit • Scientific approach not just randomly replicating sample	• Overgeneralization: It blindly generalizes the minority area without regard to the majority class. This strategy is particularly problematic in the case of highly skewed class distributions. • Lack of Flexibility: The number of synthetic samples generated is fixed in advance, thus not allowing for any flexibility in the re-balancing rate. • Overfitting: The minority can lead to model overfitting, since it will introduce duplicate instances, drawing from a pool of instances that is already small
Under sampling: Near Miss	• Easy to use • Randomly chosen cases of the majority class to decrease their effect on the classifier.	• The majority can end up leaving out important instances that provide important differences between the two classes.

References:
1. https://www.fromthegenesis.com/smote-synthetic-minority-oversampling-technique/
2. https://www.geeksforgeeks.org/ml-handling-imbalanced-data-with-smote-and-near-miss-algorithm-in-python/
3. https://medium.com/@saeedAR/smote-and-near-miss-in-python-machine-learning-in-imbalanced-datasets-b7976d9a7a79
4. https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

Thanks for reading!

Lets collaborate and happy to receive any feedback/suggestion/comment at.....

pathak.chiranjit@gmail.com