Ensemble Methods

Ensemble Methods Objective -

1. The common modeling problem is to choose the best model of all the possible ones

2. Best model is the one which has good predictive power, and which is likely to generalize.

3. For a model to generalize, it needs to be right fit and a model is right fit when bias variance errors are minimized

Model / Data	Pima Indians	German Credit	White Wine
Decision Tree (Regularised	78.6	71.3	54.4
Naïve Bayes	74.2	75.0	62.6
Logistic Regression	77.4	72.8	60.6

Sampling

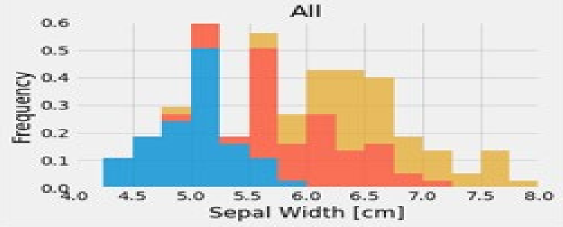
1. It all begins with sampling…. So do bias and variance errors

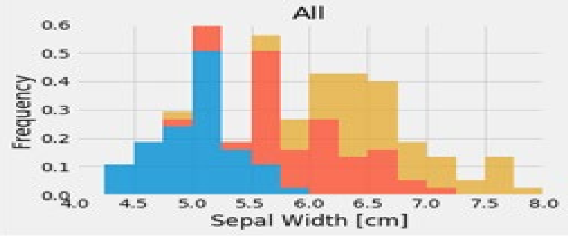
2. For a sample to be close representative of the population we need the right attributes, right size, correct class representation

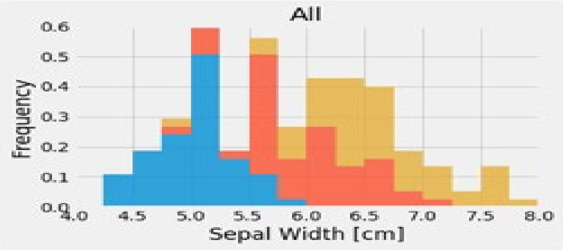
3. A sample being subset of the population can fall short of these requirements. As a result, it represents subset of the patterns, also have unexplained patterns that get clubbed as residuals

4. Problem becomes acute when we are in classification and the data is imbalanced in terms of representing the classes.

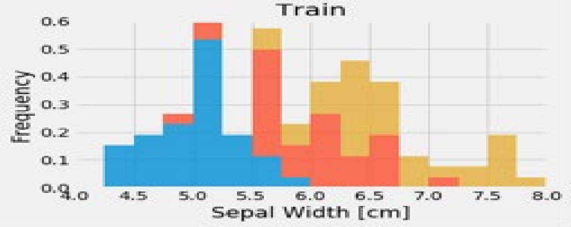
Sample Data (Analytics Base Table)

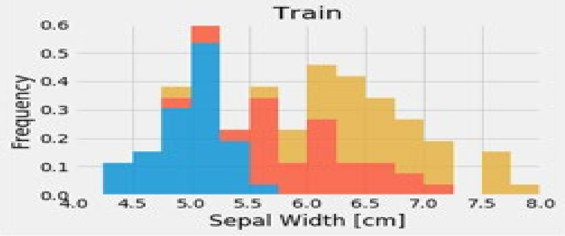


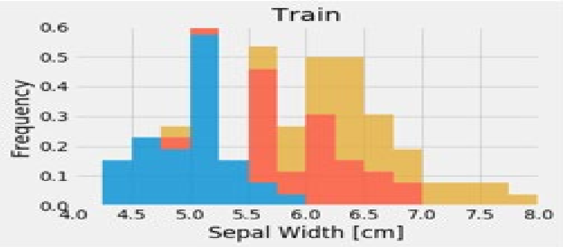




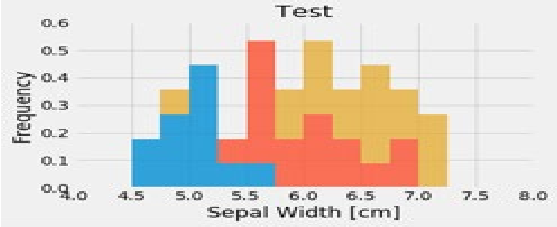
Three Random Training Sets From ABT

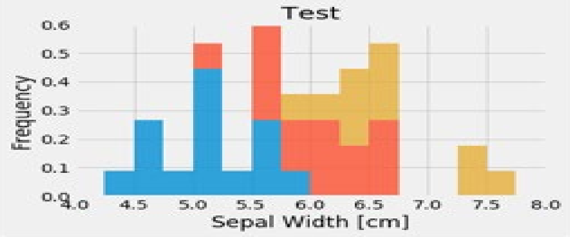


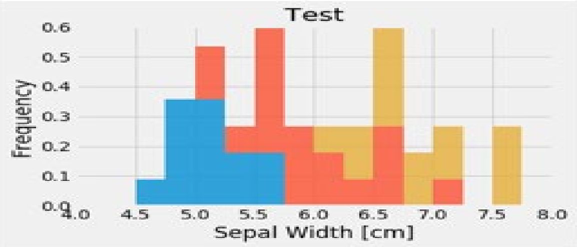




Three Random Test Sets From ABT







5. Imbalanced datasets impact outcome of ML models negatively when the class of importance is under represented

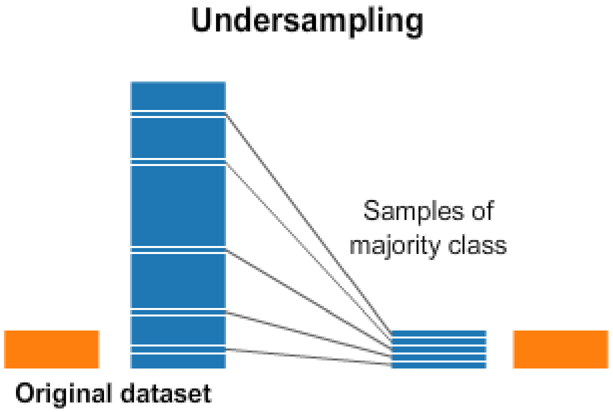
6. The ML algorithms such as decision trees, logistic regressions are designed to reduce overall inaccuracies and hence get biased towards over represented class

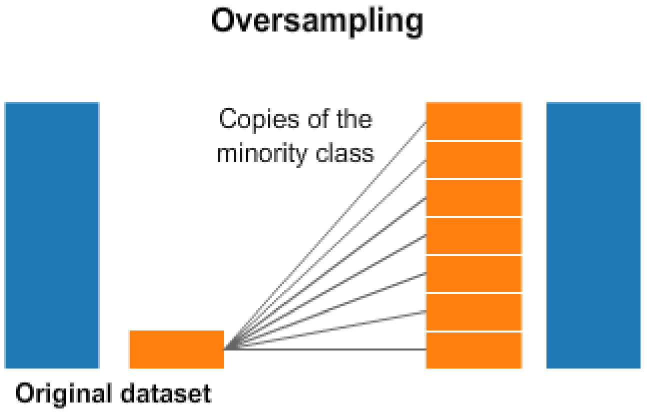
7. When class of importance is under represented, no amount of tuning the models will help

8. Suppose we have 1000 records of which 20 are fraudulent cases and 980 normal cases. We have to predict fraudulent cases accurately. The event rate is only 2%. Conventional classifiers tend to perform higher Type II errors (fraudulent cases identified as normal)

1. We can handle the imbalanced dataset cases to minimize the Type II errors by balancing the class representations

2. To balance the classes we can –
   a. Decrease the frequency of the majority class
   b. Increase the frequency of the minority class   OR


Undersampling


Oversampling

3. Decreasing the frequency of majority class is done using random under sampling. For e.g.
    a. Total observations – 1000
    b. Fraud - 020
    c. Non-fraud - 980
    d. Event rate of interest - 2%
    e. Take 10% of non-fraud cases randomly - 98
    f. Club with the fraud cases – 118 sample size
    g. Modified event rate - 20 / 118 = 17%

Ref: Logistic_up_down_sample_Pima.ipynb

4. Random oversampling is used to increase the frequency of minority class. This is done by replicating them in order to increase their representation. For e.g.
   a. Total observations – 1000
   b. Fraud - 020
   c. Non-fraud - 980
   d. Event rate of interest - 2%
   e. Replicate a % of fraud cases n times e.g. 10 cases 20 times
   f. Sample size changes from 1000 to 1200
   g. Modified event rate - 220/1200 = 18%

5. The simplest implementation of over-sampling is to duplicate random records from the minority class, which can cause overfitting.

6. In under-sampling, the simplest technique involves removing random records from the majority class, which can cause loss of information.

Ref: Logistic_up_down_sample_Pima.ipynb

Imblearn Techniques

1. Python imbalanced-learn module – provides more sophisticated resampling techniques

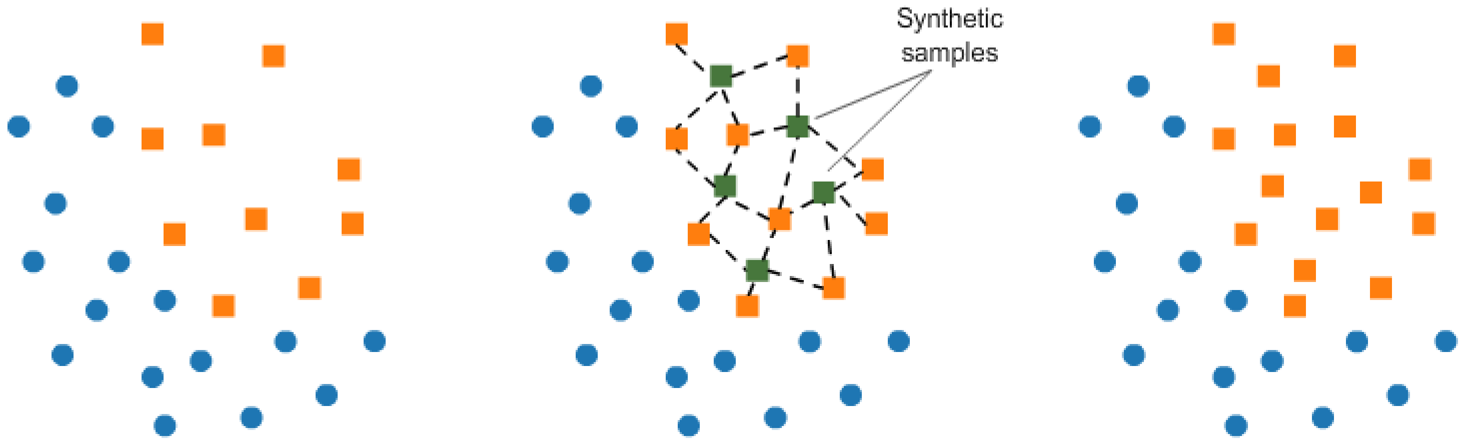
2. For example, we can cluster the records of the majority class, and do the under-sampling by removing records from each cluster, thus seeking to preserve information.

3. In over-sampling, instead of creating exact copies of the minority class records, we can introduce small variations into those copies, creating more diverse synthetic samples.

Ref: Logistic_up_down_sample_Pima.ipynb

Imblearn Techniques

4. SMOTE (Synthetic Minority Oversampling TEchnique)
    a. consists of synthesizing elements for the minority class, based on those that already exist.
    b. It works randomly picking a point from the minority class and computing the k-nearest neighbors for this point.
    c. Synthetic points are added between the chosen point and its neighbors.

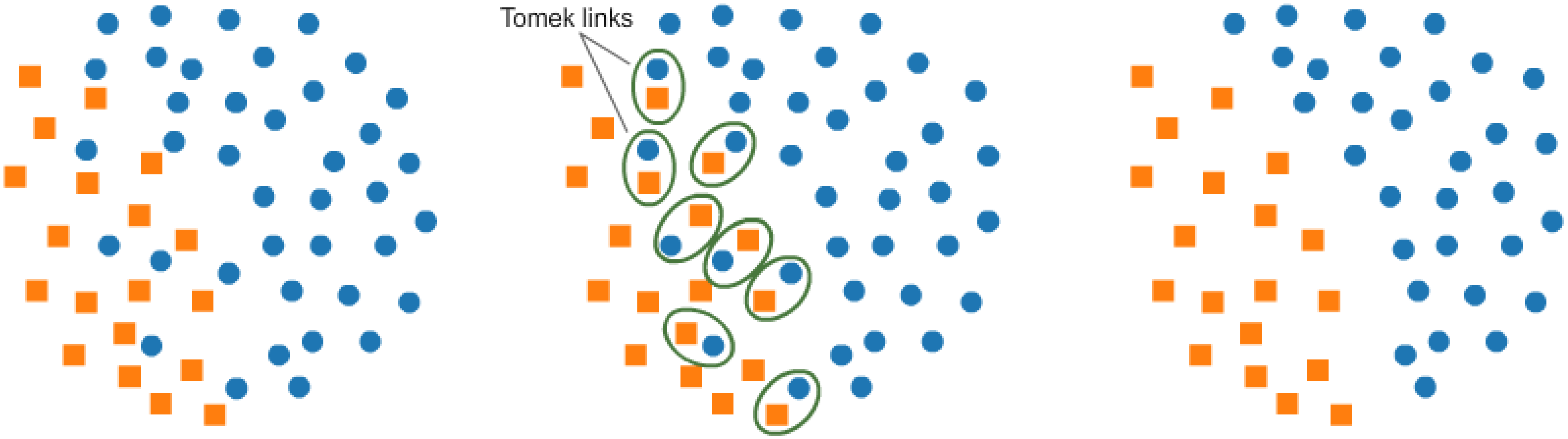


Ref: Logistic_up_down_sample_Pima.ipynb

Imblearn Techniques

5. Tomek links T-Link
    a. Tomek links are pairs of very close instances, but of opposite classes.

    b. Removing the instances of the majority class of each pair increases the space between the two classes, facilitating the classification process.



6. Cluster centroid based under sampling -

   a. Method that under samples the majority class by replacing a cluster of majority samples by the cluster centroid of a KMeans algorithm.

   b. This algorithm keeps N majority samples by fitting the KMeans algorithm with N cluster to the majority class and using the coordinates of the N cluster centroids as the new majority samples.

The imbalanced-learn documentation:
http://contrib.scikit-learn.org/imbalanced-learn/stable/index.html

The imbalanced-learn GitHub:
https://github.com/scikit-learn-contrib/imbalanced-learn

Comparison of the combination of over- and under-sampling algorithms:
http://contrib.scikit-learn.org/imbalanced-learn/stable/auto_examples/combine/plot_comparison_combine.html

Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002):
https://www.jair.org/media/953/live-953-2037-jair.pdf