Handling Imbalanced Dataset

Why:

When data set is imbalanced, it predicts the majority class correctly but minor class accuracy is very low.

1. **Random Undersampling and Oversampling**

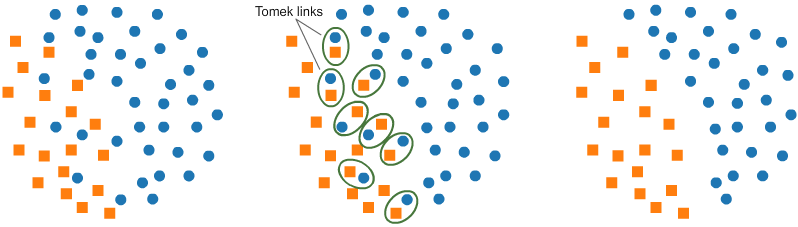
How:

imbalanced-learn(imblearn) is a Python Package to tackle the curse of imbalanced datasets.

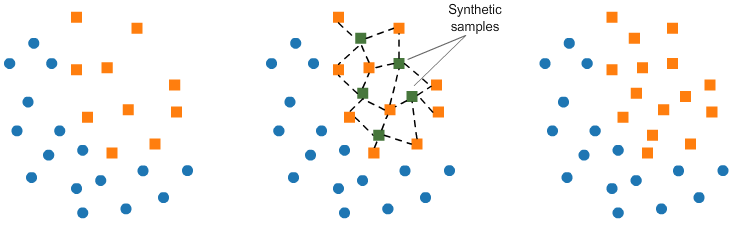
It provides a variety of methods to undersample and oversample.

There are a variety of other methods in the imblearn package for both undersampling(Cluster Centroids, NearMiss, etc.) and oversampling(ADASYN and bSMOTE) that you can check out.

1. Undersampling using Tomek Links: tomek links pair of opposite classes in close vicinity. Remove Tomek links.



1. In SMOTE (Synthetic Minority Oversampling Technique) we synthesize elements for the minority class, in the vicinity of already existing elements.



### Class weights in the models

### (<https://www.kdnuggets.com/2020/01/5-most-useful-techniques-handle-imbalanced-datasets.html>)

Most of the machine learning models provide a parameter called class\_weights. For example, in a random forest classifier using, class\_weights we can specify a higher weight for the minority class using a dictionary.

from sklearn.linear\_model import **LogisticRegression***clf =* **LogisticRegression**(***class\_weight={0:1,1:10}***

### 3- Ensembling Methods (Ensemble of Sampler):

### 4.Change your Evaluation Metric

**PERFORMANCE METRICS: CLASSIFICATIONS PROBLEMS**

1. Confusion matrix
2. Recall
3. Precision
4. F Score
5. AUC ROC Curve
6. Log Loss/Binary Crossentropy
7. Categorical Crossentropy

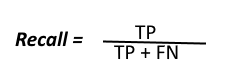
Confusion Matrix

Used for describing performance of classification problems.

The number of correct and incorrect predictions are summarized with count values and broken down by each class. T/F(True/False)

There are problems with accuracy. It assumes equal costs for both kinds of errors.

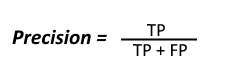


Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples.

High Recall. low P : indicates most of the sample are correctly recognized for class but there are lot of false postives

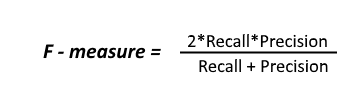
Low Recall, High P : Missed lot of positive examples but those predicted are indeed postive

.

Precision: divide the total number of correctly classified positive examples by the total number of predicted positive examples.

High Precision indicates an example labelled as

positive is indeed positive



* Harmonic mean of precision and recall,
* Will always be near to smaller value of P/R
* The high the better
* Problem is give equal weightage to P/R

When to use?

Accuracy: for classification problems which are well balanced and not skewed or No class imbalance.

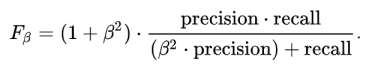
Precision: when we want to be very sure of our prediction. For example: If we are building a system to predict if we should decrease the credit limit on a particular account, we want to be very sure about our prediction or it may result in customer dissatisfaction.

Caveat: high precision may leave a lot of card holders untouched.

Recall: when we want to capture as many positives as possible. For example: If we are building a system to predict if a person has cancer or not, we want to capture the disease even if we are not very sure

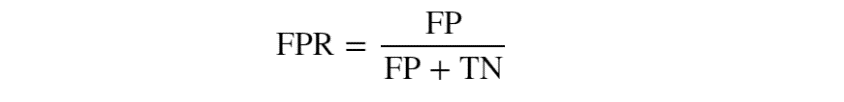
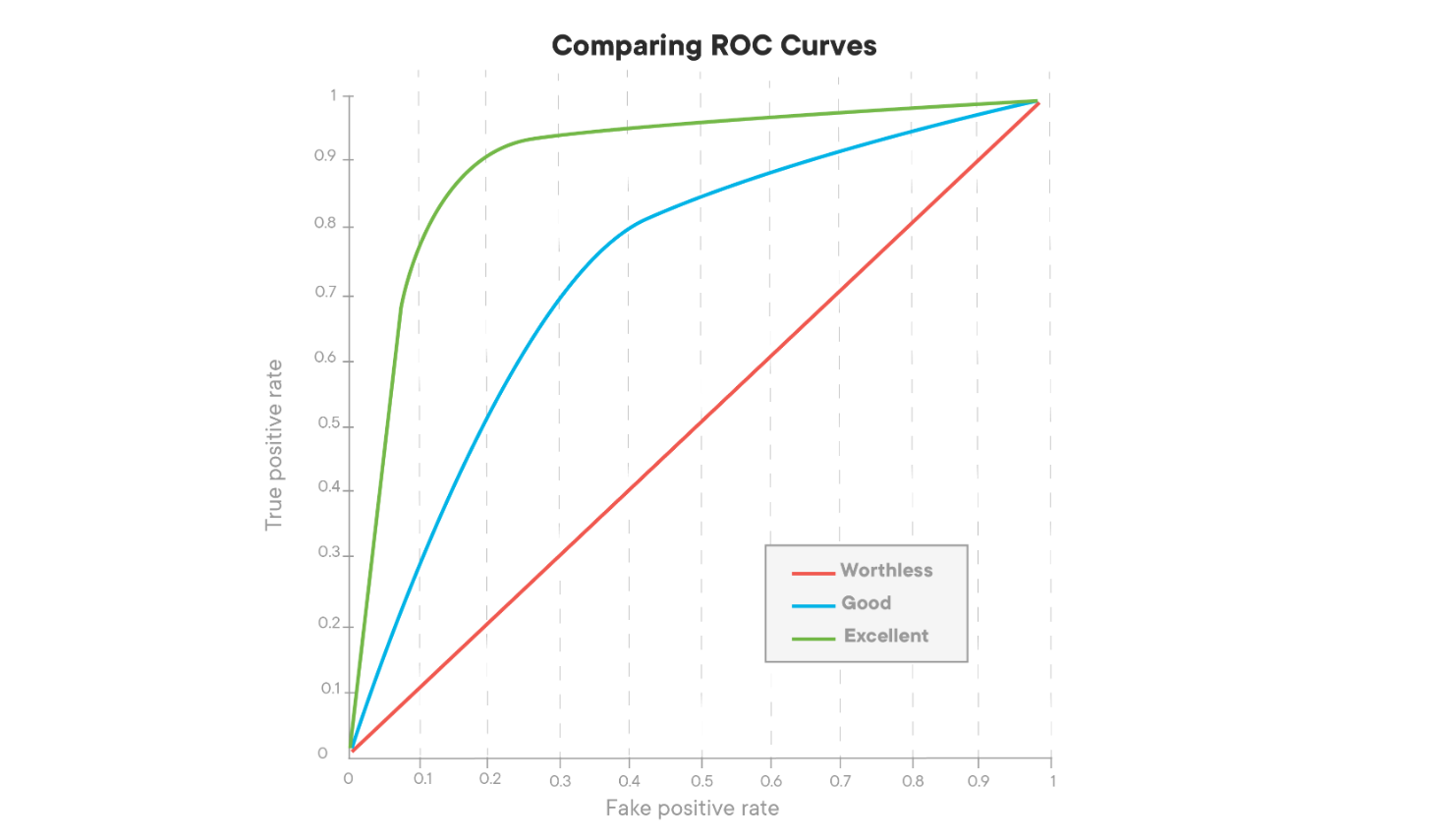
F Score: maintains a balance between the precision and recall for your classifier. If you are a police inspector and you want to catch criminals, you want to be sure that the person you catch is a criminal (Precision) and you also want to capture as many criminals (Recall) as possible. The F1 score manages this tradeoff. Caveat : Equal weightage to P/R

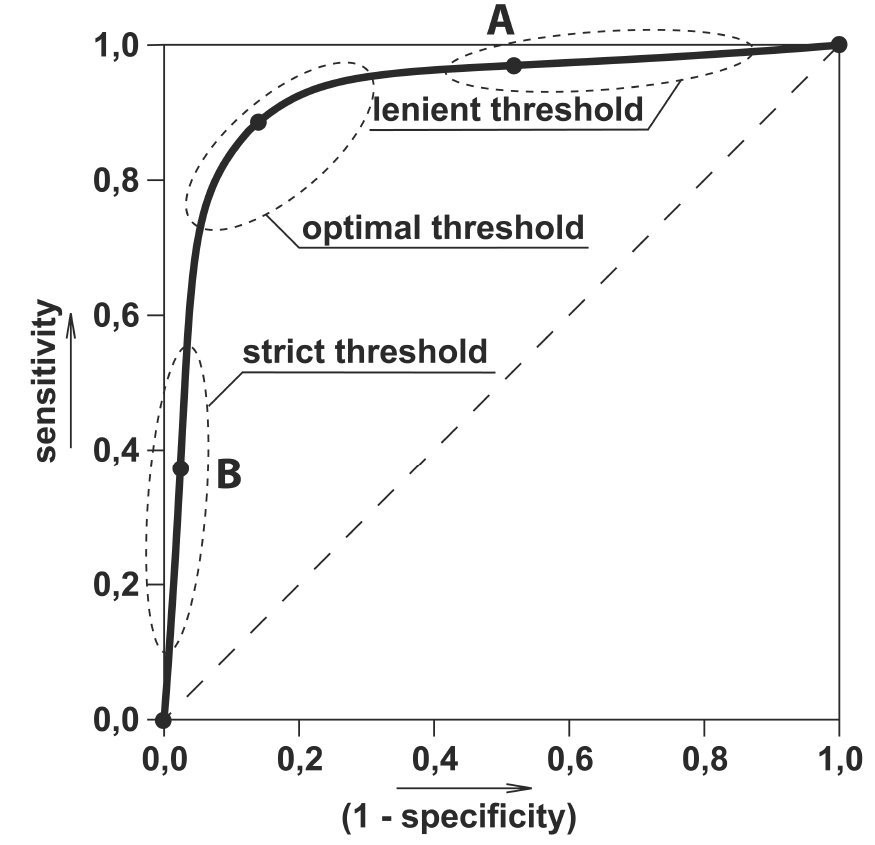
To solve this, we can do this by creating a weighted F1 metric as below where beta manages the tradeoff between precision and recall.



Here we give β times as much importance to recall as precision.

**AUC (Area Under Curve) ROC(Receiver Operating Characteristics) Curve**

* The AUC RO: graph which shows the performance of a classification model at all thresholds.
* ROC is a probability curve and AUC represents degree of separability.
* ROC plots the following parameters:
  + True Positive Rate (TPR)/ recall/sensitivity: TP/(TP+FN)
  + False Positive Rate (FPR): ratio of false postives upon all negative values
* Both the RPR and FPR are within the range [0, 1].
* The curve is the FPR vs TPR at different points in the range [0, 1]
* The best performing classification models will have a curve similar to the green line in the graph below
* The green line has the largest Area Under the Curve. The higher the AUC, the better your model is performing.
* A classifier with only 50–50 accuracy is realistically no better than randomly guessing, which makes the model worthless (red line).
* The ROC graph summarizes all of the confusion matrices that each threshold produce
* If we are comparing 2 models, then the model with ROC curve having maximum area is selected
* 



Here we can use the ROC curves to decide on a Threshold value.

The choice of threshold value will also depend on how the classifier is intended to be used.

If it is a cancer classification application you don’t want your threshold to be as big as 0.5. Even if a patient has a 0.3 probability of having cancer you would classify him to be 1.

Otherwise, in an application for reducing the limits on the credit card, you don’t want your threshold to be as less as 0.5. You are here a little worried about the negative effect of decreasing limits on customer satisfaction.

When to Use?

AUC is scale-invariant. It measures how well predictions are ranked, rather than their absolute values. So, for example, if you as a marketer want to find a list of users who will respond to a marketing campaign. AUC is a good metric to use since the predictions ranked by probability is the order in which you will create a list of users to send the marketing campaign.

Another benefit of using AUC is that it is classification-threshold-invariant like log loss. It measures the quality of the model’s predictions irrespective of what classification threshold is chosen, unlike F1 score or accuracy which depend on the choice of threshold.

Links:

<https://www.geeksforgeeks.org/confusion-matrix-machine-learning/#:~:text=A%20confusion%20matrix%20is%20a,the%20performance%20of%20an%20algorithm.>

<https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226>

<https://towardsdatascience.com/evaluation-metrics-for-classification-problems-in-machine-learning-d9f9c7313190>