1st. from sklearn import datasets

Imbalance degree: ratio of sample size of minority class to that of the majority class i.e., 1:100 (1:10 is considered imbalance). Makes it difficult for algorithm to predict outcome.



If patterns among classes overlap, it is harder to find a rule to separate, boundaries to separate the classes.

**Solutions for Imbalanced Datasets:**

Data level (modify the data – changing the distribution of the data): under-sampling or over-sampling by creating new synthetic data.

Cost sensitive – higher cost to miss-classification of minority class outweighs the cost of miss-classification of the majority class.

Ensemble algorithms – boosting and bagging, with sampling

**Metrics to evaluate algorithms:**

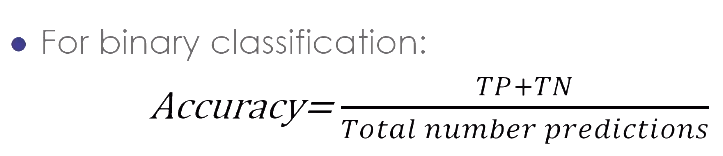
**Independent of the Probability Threshold:**

ROC-AUC

Precision-Recall Curves

**Independent of the Probability Threshold:**

Accuracy (not a good method), the minority class has very little impact on the accuracy (as the minority has few samples). It does not tell us how many of the minority class were predicted incorrectly



Confusion Matrix, then apply Precision, Recall, f-score and FPR,FNR

Let's go over the results in the classification report:

Evaluate the Model

After making predictions on the scaled testing data, we analyze how well our random forest model classifies loan applications by using the confusion\_matrix.

* **Precision:** Precision is the measure of how reliable a positive classification is. From our results, the precision for the good loan applications can be determined by the ratio TP/(TP + FP), which is 50/(50 + 22) = 0.69. The precision for the bad loan applications can be determined as follows: 19/(19 + 34) = 0.358. A low precision is indicative of a large number of false positives—of the 53 loan applications we predicted to be bad applications, 34 were actually good loan applications.
* **Recall:** Recall is the ability of the classifier to find all the positive samples. It can be determined by the ratio: TP/(TP + FN), or 50/(50 + 34) = 0.595 for the good loans and 19/(19 + 22) = 0.463 for the bad loans. A low recall is indicative of a large number of false negatives.
* **F1 score:** F1 score is a weighted average of the true positive rate (recall) and precision, where the best score is 1.0 and the worst is 0.0.
* **Support:** Support is the number of actual occurrences of the class in the specified dataset. For our results, there are 84 actual occurrences for the good loans and 41 actual occurrences for bad loans.

In summary, this model may not be the best one for preventing fraudulent loan applications because the model's accuracy, 0.552, is low, and the precision and recall are not good enough to state that the model will be good at classifying fraudulent loan applications. Modeling is an iterative process: you may need more data, more cleaning, another model parameter, or a different model. It's also important to have a goal that's been agreed upon, so that you know when the model is good enough.

**Rank the Importance of Features**

One nice byproduct of the random forest algorithm is to rank the features by their importance, which allows us to see which features have the most impact on the decision.

To calculate the feature importance, we can use thefeature\_importances\_attribute with the following code:

Now we can clearly see which features, or columns, of the loan application are more relevant. The age and month\_num of the loan application are the more relevant features.

It looks like the amount of total payments received (total\_rec\_prncp) 0.07 (along similar measures like total\_rec\_int, total\_pymt\_inv, total\_pymnt, last\_pymt\_amt) are very important.

In addition the interest rate (int\_rate), outsdanding principal (out\_prncp) and dti and max\_bal\_bc are relevant.

We could improve the model, we can drop some of the lower ranked features.