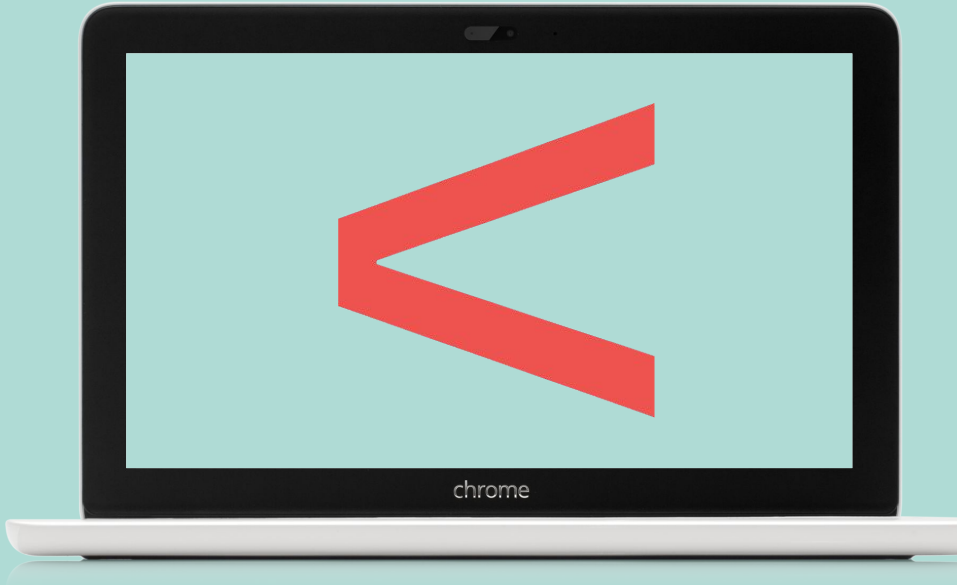
Evaluation metrics for ML

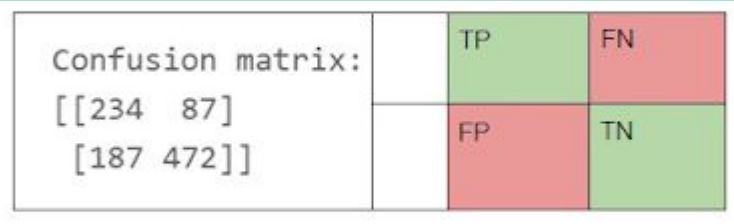


Why is it important to evaluate our ML model?

Model evaluation allows you to determine if the model you've built will do a good job of **predicting the target on new and future data**.

To do this, you will evaluate the prediction on data for which you already know the target answer, and use this assessment as a proxy for predictive accuracy on future data.

Confusion matrix



Tabular visualisation of the predictions made by your model vs their actual class.

- TN / True Negative: when a case was negative and predicted negative
- TP / True Positive: when a case was positive and predicted positive
- FN / False Negative: when a case was positive but predicted negative
- FP / False Positive: when a case was negative but predicted positive

Accuracy score

- **Accuracy score:**
  Number of correct predictions divided by the total number of predictions.

- **Formula:**
  (TP + TN) / (TP + TN + FP + FN)

- Doesn't work well with unbalanced data.

Classification report

	precision	recall	f1-score	support
low	0.56	0.73	0.63	321
medium	0.84	0.72	0.78	659

- Precision = TP / (TP + FP)

► "What percent of your predictions were correct?"

► The best value is 1 and the worst value is 0

- Recall = TP / (TP + FN)

► "What percent of the positive cases did you catch?"

► The best value is 1 and the worst value is 0

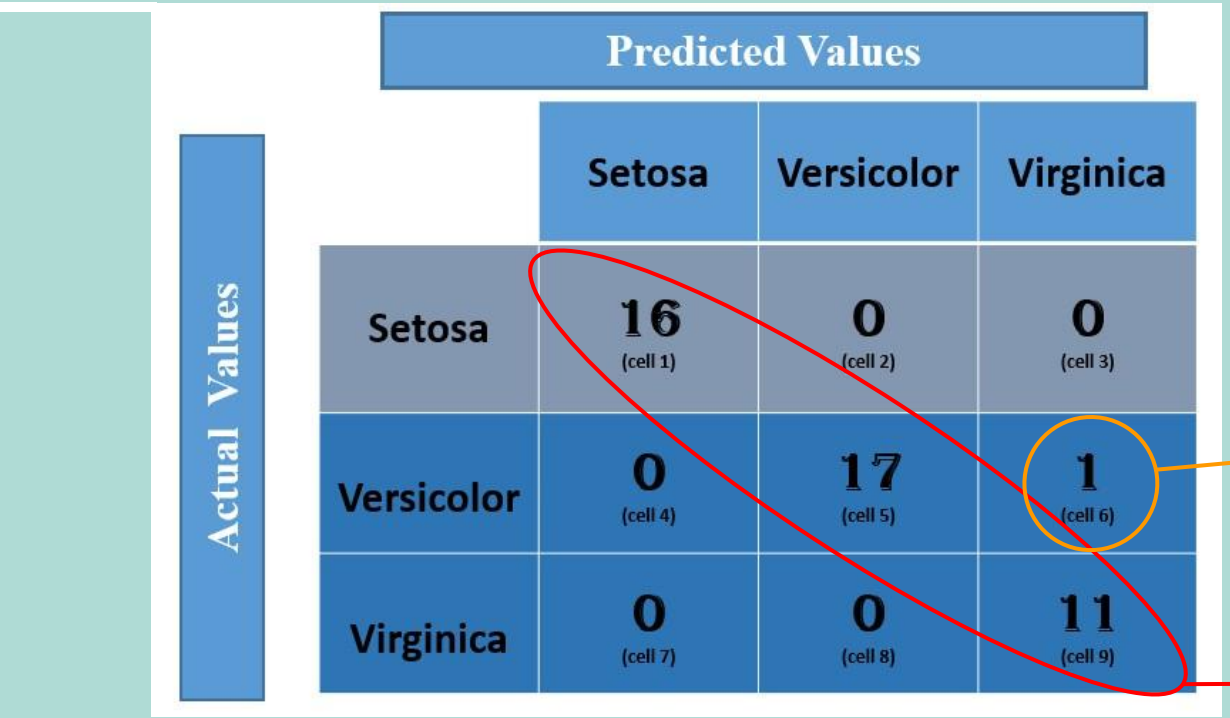
- F1 = 2 * (precision * recall) / (precision + recall)

► It is defined as the harmonic mean of the model's precision and recall

► The best value is 1 and the worst value is 0

► As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy.

Confusion matrix for multiclass classification



https://www.analyticsvidhya.com/blog/2021/06/confusion-matrix-for-multi-class-classification/

Cohen’s kappa coefficient

- Cohen’s Kappa is a means for evaluating the prediction performance of classifiers amongst themselves

- It compares the Accuracy (number of instances that were classified correctly) with an Expected Accuracy (random chance).

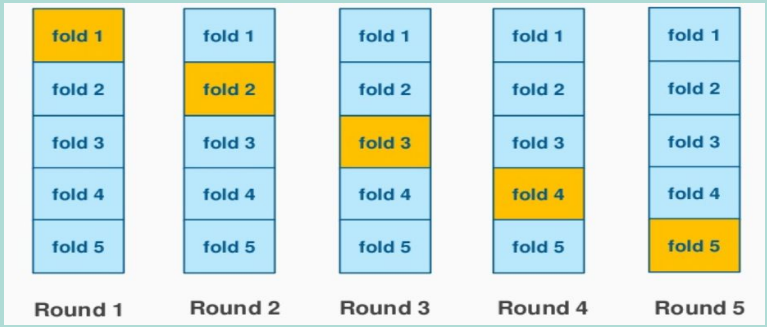
► Cohen’s kappa takes imbalance in class distribution into account.

► Its value is a number between -1 and 1. Scores above .8 are generally considered good agreement; zero or lower means no agreement (practically random labels).

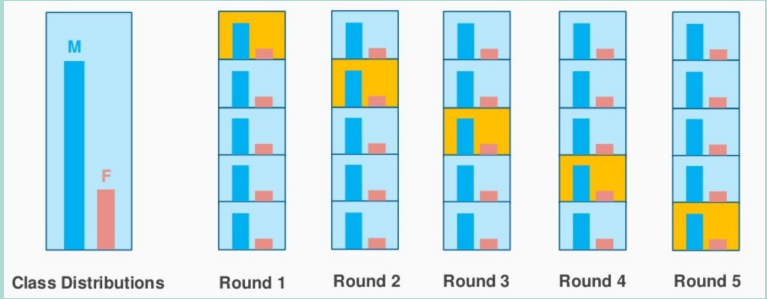
► Kappa scores can be computed for binary or multiclass problems, but not for multilabel problems (except by manually computing a per-label score) and not for more than two annotators.

Stratified K-Fold Cross-Validation (for imbalanced data)

Regular cross-validation (random folds)



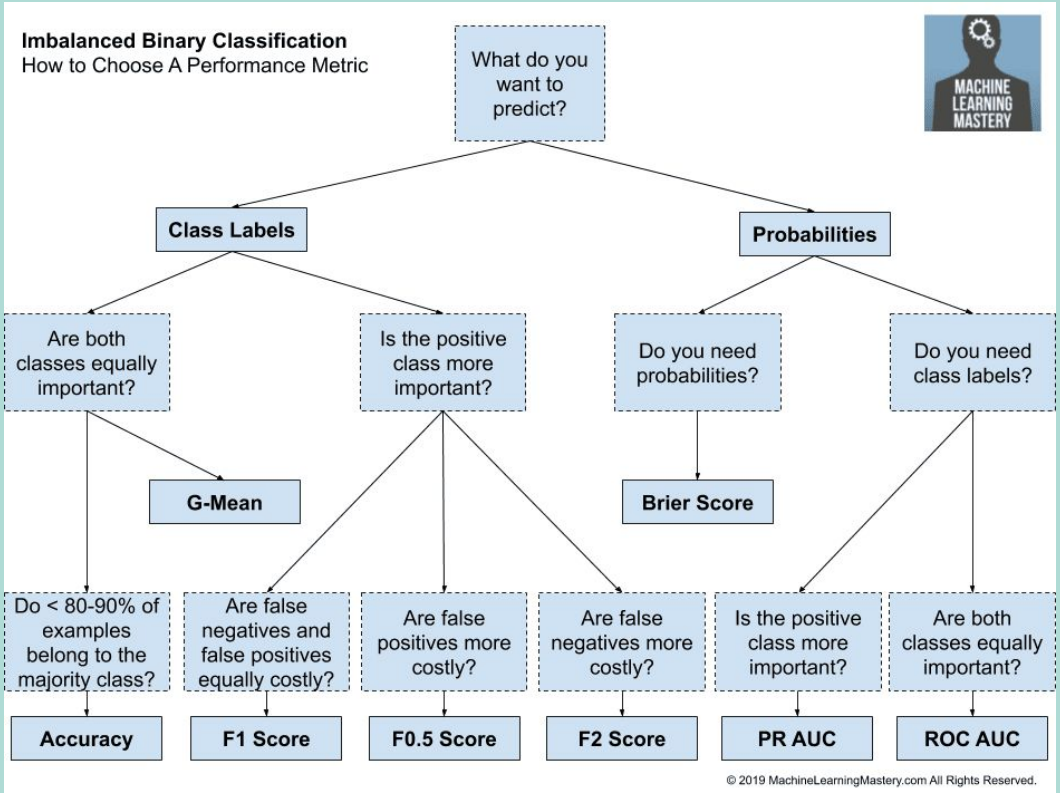
K-fold cross-validation (non-random folds)



The stratified k fold cross-validation is an extension of the cross-validation technique used for classification problems. It maintains the same class ratio throughout the K folds as the ratio in the original dataset.

In other words, we divide the data into k subsets of equal size, where each fold will have the same ratio of instances of target variable as in the whole dataset.

Guide to choose an Evaluation Metric



Source: machinelearningmastery.com

If you are using an imbalanced dataset in your model, the results will look poor with high false negative values/low sensitivity ratio.
One metric you may want to use in addition is the AUC/ROC. This works well for comparing results for imbalanced data. There are many options to improve your model for a better sensitivity result, such as trying different algorithms or tuning the parameters of these algorithms.

Using different weights for predicting a majority vs predicting a minority label:
Up-sample or down-sample the training data to help balance the prediction across minority and majority, or use SMOTE for datasets with few features.
Choose a modeling algorithm that is better at handling imbalanced data, such as machine learning/neural networks, or a classical algorithm such as bagging or boosting algorithms and/or decision trees.

Resources:
https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/

https://www.analyticsvidhya.com/blog/2017/03/imbalanced-data-classification/