Precision or Recall

So basically if we want to focus more on minimising False Negatives, we would want our Recall to be as close to 100% as possible without precision being too bad and if we want to focus on minimising False positives, then our focus should be to make Precision as close to 100% as possible.

Logarithmic Loss

Logarithmic Loss or Log Loss, works by penalising the false classifications. It works well for multi-class classification. When working with Log Loss, the classifier must assign probability to each class for all the samples. Suppose, there are N samples belonging to M classes, then the Log Loss is calculated as below :

$$LogarithmicLoss = \frac{-1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} * \log(p_{ij})$$

where,

y_ij, indicates whether sample i belongs to class j or not

p_ij, indicates the probability of sample i belonging to class j

Log Loss has no upper bound and it exists on the range [0, ∞). Log Loss nearer to 0 indicates higher accuracy, whereas if the Log Loss is away from 0 then it indicates lower accuracy.

In general, minimising Log Loss gives greater accuracy for the classifier.

Area Under Curve

Area Under Curve(AUC) is one of the most widely used metrics for evaluation. It is used for binary classification problem. AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example. Before defining AUC, let us understand two basic terms :

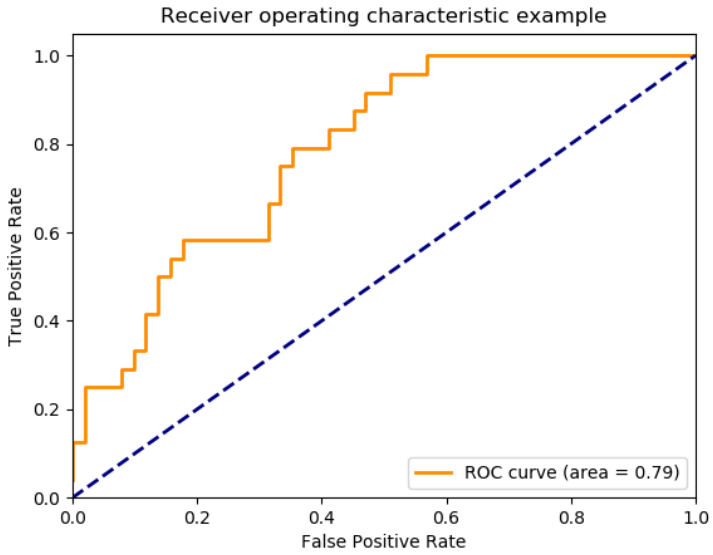
- **True Positive Rate (Sensitivity)** : True Positive Rate is defined as *TP/ (FN*+TP). True Positive Rate corresponds to the proportion of positive data points that are correctly considered as positive, with respect to all positive data points.

$$TruePositiveRate = \frac{TruePositive}{FalseNegative + TruePositive}$$

- **False Positive Rate (Specificity)** : False Positive Rate is defined as *FP / (*FP+TN). False Positive Rate corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points.

$$FalsePositiveRate = \frac{FalsePositive}{FalsePositive + TrueNegative}$$

False Positive Rate and True Positive Rate both have values in the range [0, 1]. FPR and TPR bot hare computed at threshold values such as (0.00, 0.02, 0.04, …., 1.00) and a graph is drawn. AUC is the area under the curve of plot False Positive Rate vs True Positive Rate at different points in [0, 1].



As evident, AUC has a range of [0, 1]. The greater the value, the better is the performance of our model.

F1 Score

F1 Score is used to measure a test's accuracy

F1 Score is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances).

High precision but lower recall, gives you an extremely accurate, but it then misses a large number of instances that are difficult to classify. The greater the F1 Score, the better is the performance of our model. Mathematically, it can be expressed as :

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

F1 Score tries to find the balance between precision and recall.

- **Precision :** It is the number of correct positive results divided by the number of positive results predicted by the classifier.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

- **Recall :** It is the number of correct positive results divided by the number of ***all*** relevant samples (all samples that should have been identified as positive).

$$Precision = \frac{TruePositives}{TruePositives + FalseNegatives}$$

Mean Absolute Error

Mean Absolute Error is the average of the difference between the Original Values and the Predicted Values. It gives us the measure of how far the predictions were from the actual output. However,

they don’t gives us any idea of the direction of the error i.e. whether we are under predicting the data or over predicting the data. Mathematically, it is represented as :

$$MeanAbsoluteError = \frac{1}{N}\sum_{j=1}^{N}|y_j - \hat{y}_j|$$

Mean Squared Error

Mean Squared Error(MSE) is quite similar to Mean Absolute Error, the only difference being that MSE takes the average of the **square** of the difference between the original values and the predicted values. The advantage of MSE being that it is easier to compute the gradient, whereas Mean Absolute Error requires complicated linear programming tools to compute the gradient. As, we take square of the error, the effect of larger errors become more pronounced then smaller error, hence the model can now focus more on the larger errors.

$$MeanSquaredError = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2$$

What is Cohen's Kappa?

Kappa is similar to Accuracy score, but it takes into account the accuracy that would have happened anyway through random predictions.

Kappa = (Observed Accuracy - Expected Accuracy) / (1 - Expected Accuracy)

Cohen's kappa is shown as an output of caret's `confusionMatrix` function.

7. What is KS Statistic and How to interpret KS Chart?

The KS Statistic and the KS Chart (discussed next) are used to make decisions like: How many customers to target for a marketing campaign? or How many customers should we pay for to show ads etc.

So how to compute the Kolmogorov-Smirnov statistic?

Step 1: Once the prediction probability scores are obtained, the observations are sorted by decreasing order of probability scores. This way, you can expect the rows at the top to be classified as 1 while rows at the bottom to be 0's.

Step 2: All observations are then split into 10 equal sized buckets (bins).

Step 3: Then, KS statistic is the maximum difference between the cumulative percentage of responders or 1's (cumulative true positive rate) and cumulative percentage of non-responders or 0's (cumulative false positive rate).

The significance of KS statistic is, it helps to understand, what portion of the population should be targeted to get the highest response rate (1's).

The KS statistic can be computed using the `ks_stat` function in `InformationValue` package. By setting the `returnKSTable = T`, you can retrieve the table that contains the detailed decile level splits.

How to Interpret ROC Curve?

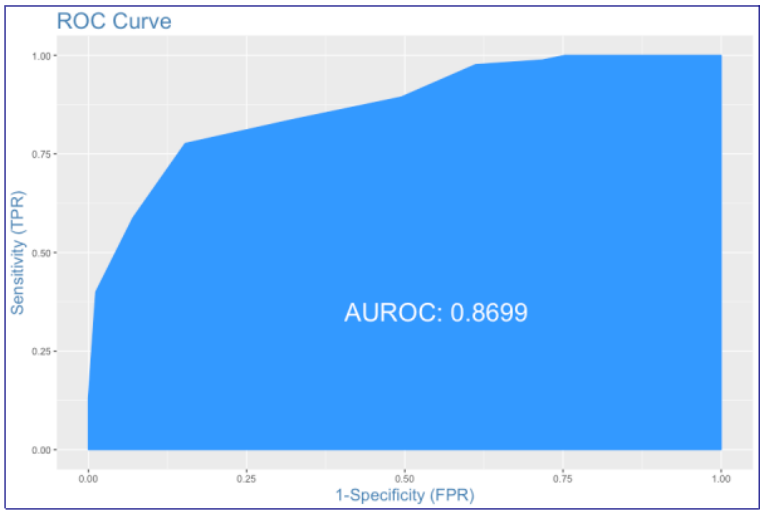
Often, choosing the best model is sort of a balance between predicting the one's accurately or the zeroes accurately. In other words sensitivity and specificity.

But it would be great to have something that captures both these aspects in one single metric.

This is nicely captured by the 'Receiver Operating Characteristics' curve, also called as the ROC curve. In fact, the area under the ROC curve can be used as an evaluation metric to compare the efficacy of the models.

Let's plot the curve and the area using the `plotROC` and `AUROC` functions from `InformationValue` package.

The actuals are contained in `y_act` and the predictions are contained in pred. You need to pass it in the same order to get the curve right.



ROC Plot

The area under the ROC curve is also shown. But how to interpret this plot?

Interpreting the ROC plot is very different from a regular line plot. Because, though there is an X and a Y-axis, you don't read it as: for an X value of 0.25, the Y value is .9.

Instead, what we have here is a line that traces the probability cutoff from 1 at the bottom-left to 0 in the top right.

This is a way of analyzing how the sensitivity and specificity perform for the full range of probability cutoffs, that is from 0 to 1.

Ideally, if you have a perfect model, all the events will have a probability score of 1 and all non-events will have a score of 0. For such a model, the area under the ROC will be a perfect 1.

So, if we trace the curve from bottom left, the value of probability cutoff decreases from 1 towards 0. If you have a good model, more of the real events should be predicted as events, resulting in high sensitivity and low FPR. In that case, the curve will rise steeply covering a large area before reaching the top-right.

Therefore, the larger the area under the ROC curve, the better is your model.

The ROC curve is the only metric that measures how well the model does for different values of prediction probability cutoffs. The `optimalCutoff` function from `InformationValue` can be used to know what cutoff gives the best sensitivity, specificity or both.

11. What is Somers-D Statistic?

Again, Somers D is an evaluation metric to judge the efficacy of the model.

Somers D = (#Concordant Pairs - #Discordant Pairs - #Ties) / Total Pairs

12. What is Gini Coefficient?

Gini Coefficient is an indicator of how well the model outperforms random predictions. It can be computed from the area under the ROC curve using the following formula:

Gini Coefficient = (2 * AUROC) - 1

13. Conclusion

Clearly, just the accuracy score is not enough to judge the performance of the models. One or a combination of the following evaluation metrics is typically required to do the job.

	Truth	
Predicted	**1**	**0**
1	**A**	**B**
0	**C**	**D**

Confusion Matrix

No.	Evaluation Metric	Formula	Interpretation
1	Sensitivity	A /(A + C)	What percentage of all 1's were correctly predicted?
2	Specificity	D/(B+D)	What percentage of all 0's were correctly predicted?
3	Prevalence	(A+C)/(A+B+C+D)	Percentage of True 1's in the sample
4	Detection Rate	A/(A+B+C+D)	Correctly predicted 1's as a percentage of entire sample
5	Detection Prevalence	(A+B)/(A+B+C+D)	What percentage of the full sample was predicted as 1?
6	Balanced Accuracy	(sensitivity+specificity)/2	A balance between correctly predicting the 1's and 0's
7	Precision	A/(A+B)	What percentage of predicted 1's are correct?
8	Recall	A/(A+C)	What percentage of all 1's were correctly predicted?
9	F1 Score	2 * Precision * Recall / (Precision + Recall)	A combination of Precision and Recall
10	Cohen's Kappa	(Observed Accuracy - Expected Accuracy) / (1 - Expected Accuracy)	How the model exceeded random predictions in terms of accuracy

No.	Evaluation Metric	Formula	Interpretation
11	Concordance	Proportion of Concordant Pairs	Proportion of Concordant Pairs
12	Somers D	(Concordant Pairs - Discordant Pairs - Ties) / Total Pairs	A combination of concordance and discordance
13	AUROC	Area Under the ROC Curve	Model's true performance considering all possible probability cutoffs
14	Gini Coefficient	(2 * AUROC) - 1	How the model exceeded random predictions in terms of ROC
15	KS Statistic	Max(Cumulative% 1's - Cumulative% 0's)	Used to decide how many customers to target
16	Youden's J Index	Sensitivity + Specificity - 1	Similar to balanced accuracy

In classification problems, we use two types of algorithms (dependent on the kind of output it creates):

1. **Class output** : Algorithms like SVM and KNN create a class output. For instance, in a binary classification problem, the outputs will be either 0 or 1. However, today we have algorithms which can convert these class outputs to probability. But these algorithms are not well accepted by the statistics community.
2. **Probability output** : Algorithms like Logistic Regression, Random Forest, Gradient Boosting, Adaboost etc. give probability outputs. Converting probability outputs to class output is just a matter of creating a threshold probability.

3. Kolomogorov Smirnov chart

K-S or Kolmogorov-Smirnov chart measures performance of classification models. More accurately, K-S is a measure of the degree of separation between the positive and negative distributions. The K-S is 100, if the scores partition the population into two separate groups in which one group contains all the positives and the other all the negatives.

On the other hand, If the model cannot differentiate between positives and negatives, then it is as if the model selects cases randomly from the population. The K-S would be 0. In most classification models the K-S will fall between 0 and 100, and that the higher the value the better the model is at separating the positive from negative cases.

For the case in hand, following is the table :

						Cummulative		
Lift/Gain	Columr			%Rights	%Wrongs	Cum %Rig	Cum %Wrong	K-S
Row La	0	1	Grand Tot	0%	0%	0%	0%	0%
1		543	543	14%	0%	14%	0%	14%
2	2	542	544	14%	0%	28%	0%	28%
3	7	537	544	14%	0%	42%	1%	42%
4	15	529	544	14%	1%	56%	2%	54%
5	20	524	544	14%	1%	69%	3%	67%
6	42	502	544	13%	3%	83%	5%	77%
7	104	440	544	11%	7%	94%	12%	82% K-S
8	345	199	544	5%	22%	99%	34%	65%
9	515	29	544	1%	32%	100%	66%	34%
10	540	5	545	0%	34%	100%	100%	0%
Grand Tot	1590	3850	5440					

condition positive (P)

the number of real positive cases in the data

condition negative (N)

the number of real negative cases in the data

---

true positive (TP)

eqv. with hit

true negative (TN)

eqv. with correct rejection

false positive (FP)

eqv. with false alarm, Type I error

false negative (FN)

eqv. with miss, Type II error

---

sensitivity, recall, hit rate, or true positive rate (TPR)

$$\mathrm{TPR} = \frac{\mathrm{TP}}{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

specificity or true negative rate (TNR)

$$\mathrm{TNR} = \frac{\mathrm{TN}}{N} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

precision or positive predictive value (PPV)

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

negative predictive value (NPV)

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}}$$

miss rate or false negative rate (FNR)

$$\mathrm{FNR} = \frac{\mathrm{FN}}{P} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}} = 1 - \mathrm{TPR}$$

fall-out or false positive rate (FPR)

$$\mathrm{FPR} = \frac{\mathrm{FP}}{N} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = 1 - \mathrm{TNR}$$

false discovery rate (FDR)

$$\mathrm{FDR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TP}} = 1 - \mathrm{PPV}$$

false omission rate (FOR)

$$\mathrm{FOR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TN}} = 1 - \mathrm{NPV}$$

accuracy (ACC)

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{P + N} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

F1 score

is the harmonic mean of precision and sensitivity

$$\mathrm{PPV} \cdot \mathrm{TPR} \qquad 2\mathrm{TP}$$

Matthews correlation coefficient (MCC)

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}$$

Informedness or Bookmaker Informedness (BM)

$$\mathrm{BM} = \mathrm{TPR} + \mathrm{TNR} - 1$$

Markedness (MK)

$$\mathrm{MK} = \mathrm{PPV} + \mathrm{NPV} - 1$$