Types of Errors

Types of Predicted Values

- **Categorical** like Yes/No , Purchased/Not Purchased and also other type of categorical values, not necessarily only binary. We use Classification Confusion Matrix for evaluation
- **Numeric** like Sales, Cost, Profit, Scores
  - We use RMSE, RMSPE, MAPE for evaluation

Categorical: Example

- Suppose that we have predicted a categorical variable named defaulter which has values as Y (Defaulter) and N (Not a Defaulter) on the validation dataset using a model built on training dataset
- Here, we term defaulter(Y) as positive class and non-defaulter(N) as negative class
- Say, the validation set has got some 30 values as

Y N Y N N N N Y Y N N N Y N N N N N N Y Y N N Y N N N N N Y Y

Diagnosis

- In the following cases, we won't have errors:
  - We predict a defaulter as defaulter
  - We predict a non-defaulter as non-defaulter
- In the following cases we have errors:
  - We predict a defaulter as non-defaulter
  - We predict a non-defaulter as defaulter

Indicators Tabulated

	Actually a Defaulter (+ve class)	Actually a Non-Defaulter (-ve class)
Predicted as Defaulter	True +ve	False +ve
Predicted as Non-Defaulter	False –ve	True -ve

The Matrix shown above is called **Classification Confusion Matrix**

Basic quantitative quality indicators

- **TP – True Positive** : Correctly assigned observations to the positive class.
- **TN – True Negative** : Correctly assigned observations to the negative class.
- **FP – False Positive** : Wrongly assigned observations to the positive class. (Which actually belong to the negative class)
- **FN – False Negative** : Wrongly assigned observations to the negative class. (Which actually belong to the positive class)

Classification Confusion Matrix

Sensitivity = TP / (TP + FN)

False Positive Rate = FP / (TN + FP)

	Actually a Defaulter (+ve class)	Actually a Non-Defaulter (-ve class)
Predicted as Defaulter	TP (Defaulter diagnosed as Defaulter)	FP (Non-Defaulter diagnosed as Defaulter)
Predicted as Non-Defaulter	FN (Defaulter diagnosed as Non-Defaulter)	TN (Non-Defaulter diagnosed as Non-Defaulter)

False Negative Rate = FN / (TP + FN)

Specificity= TN / (TN + FP)

Overall Prediction Correctness

ACC (Total Accuracy)

= P(correct prediction)

= number of correct decision/ total number of decisions

$$ACC = (TP + TN) / (TP + TN + FP + FN)$$

	Actually a Defaulter (+ve class)	Actually a Non-Defaulter (-ve class)
Predicted as Defaulter	TP (Defaulter diagnosed as Defaulter)	FP (Non-Defaulter diagnosed as Defaulter)
Predicted as Non-Defaulter	FN (Defaulter diagnosed as Non-Defaulter)	TN (Non-Defaulter diagnosed as Non-Defaulter)

Example

```
> Defaulter
 [1] Y N Y N N N N Y Y N N N Y N N N N N Y Y N N Y N N N N N Y Y
Levels: Y N
> Predicted
 [1] N N N N Y Y N Y Y N N N Y N N N Y N Y Y Y N Y N N N N N N Y
Levels: Y N
> table(Predicted,Defaulter)
         Defaulter
Predicted  Y  N
        Y  7  4
        N  3 16
```

$$Accuracy = \frac{(7+16)}{(7+4+3+16)}$$

$$Sensitivity = \frac{7}{7+3}$$

$$False\ Negative\ Rate = \frac{3}{7+3}$$

$$Specificity = \frac{16}{16+4}$$

$$False\ Positive\ Rate = \frac{4}{4+16}$$

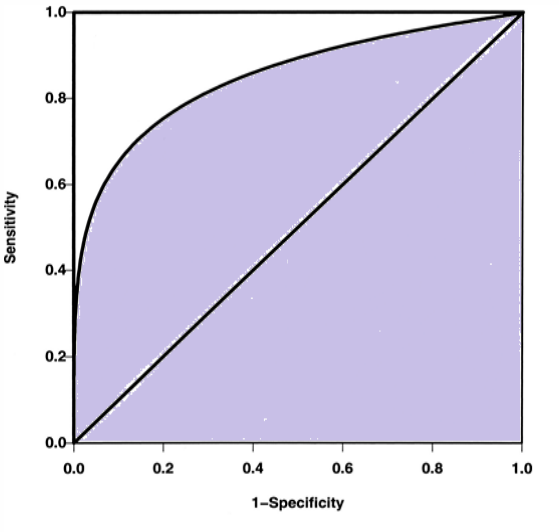
Receiver Operating Characteristic Curve

ROC Curve

What is ROC curve?

- Receiver operating characteristic (ROC), or ROC curve, is a graphical plot that illustrates the performance of a binary classifier algorithm.
- The curve is created by plotting the Sensitivity(Y axis) or true positive rate (TPR) against the (1 – Specificity) (X axis) or false positive rate (FPR) at various threshold settings.

ROC Curve



0 <= AUC <= 1
AUC = 0.5 for Random Guessing
= 1 for perfect classification
Usually,
AUC > 0.8 is considered as good

- The area is measured of lower right portion of the curve.
- That area is termed as AUC or area under the curve
- The area to be considered has been indicated by the coloured portion
- Bigger the AUC better is the model

From where did the ROC come from?

- The ROC curve was first developed by electrical engineers and radar engineers during World War II for detecting enemy objects in battlefields.
- They were building the "Chain Home" series of radar detectors to identify incoming German planes. But the radar detectors would also detect flocks of birds and other "false positive" signals.

Origin of ROC

- The term “receiver operating characteristic” came from tests of the ability of World War II radar operators to determine whether a blip on the radar screen represented an object (signal) or noise.
- The science of “signal detection theory” was later applied to diagnostic medicine and later in the other branches of research and analysis.

For Numeric / Continuous Response

- For numeric or continuous response variables we use the function postResample() which we will be covering in the subsequent sessions
- postResample() calculates Root Mean Square Error (RMSE)
- Other measures we calculate are
  – RMSPE (Root Mean Square Percentage Error)
  – MAPE (Mean Absolute Percentage Error)

Model Evaluation: RMSE

- RMSE : Root Mean Square Error

$$RMSE = \sqrt{\frac{\sum(y_i - \widehat{y_i}\ )^2}{n}}$$

where

y_i = Observed Values
$\widehat{y_i}$ = Predicted Values
n = No. of observations

Model Evaluation: MAPE

- MAPE: Mean Absolute Percentage Error

$$\text{MAPE} = \frac{\sum \left| \frac{y_i - \widehat{y_i}}{y_i} \right|}{n}$$

- Where

y_i = Observed Values

$\widehat{y_i}$ = Predicted Values

n = No. of observations

Model Evaluation: RMSPE

- RMSPE: Root Mean Square Percentage Error
  - Often used at Kaggle competitions

$$\mathrm{RMSPE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2}$$

- Where

y_i = Observed Values

$\hat{y}_i$ = Predicted Values

n = No. of observations

Example

- Suppose that we have response variable from validation dataset with length 162 as *validation$price* and we predict it based on a regression model built on a training dataset as *pred.RT*.

```
> RMSE <- function(y, yhat) {
+   sqrt(mean((y - yhat)^2))
+ }
> RMSE(validation$price, pred.RT)
[1] 18090.74
>
> MAPE <- function(y, yhat) {
+   mean(abs((y - yhat)/y))
+ }
> MAPE(validation$price , pred.RT)
[1] 0.2036191
>
> RMSPE<- function(y, yhat) {
+   sqrt(mean((y-yhat)/y)^2)
+ }
> RMSPE(validation$price , pred.RT)
[1] 0.04037694
```

Model Evaluation

- About MAPE, RMSE and RMPSE, only one criterion holds: **Smaller** their value, **Better** is the model prediction