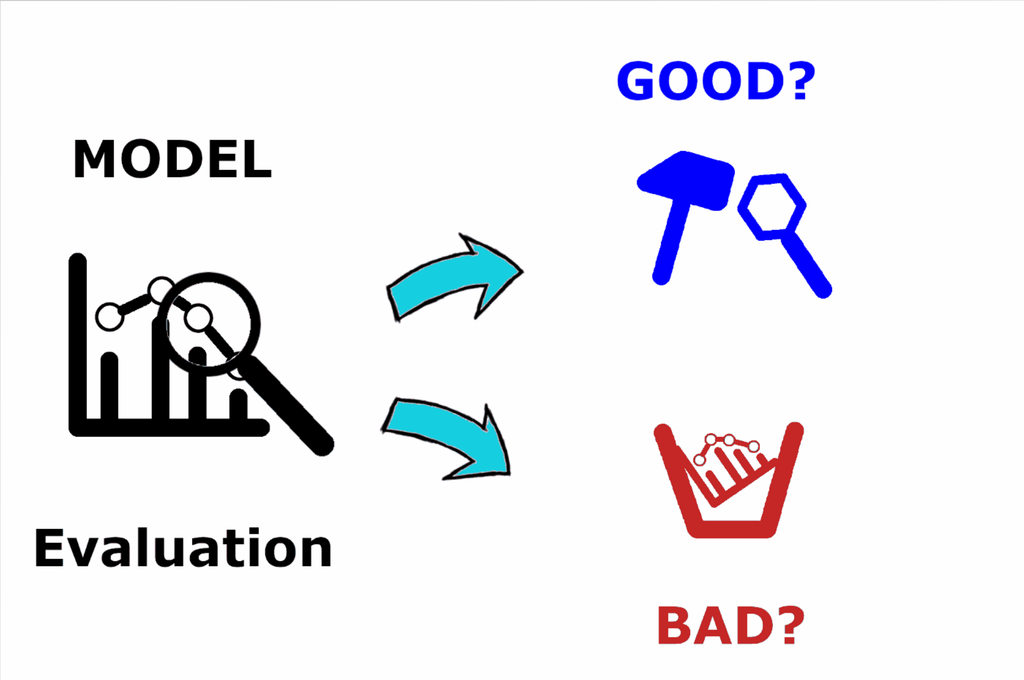
5.6 Model Performance Assessment

Dr. Sultan Alfarhood

Model Performance Assessment

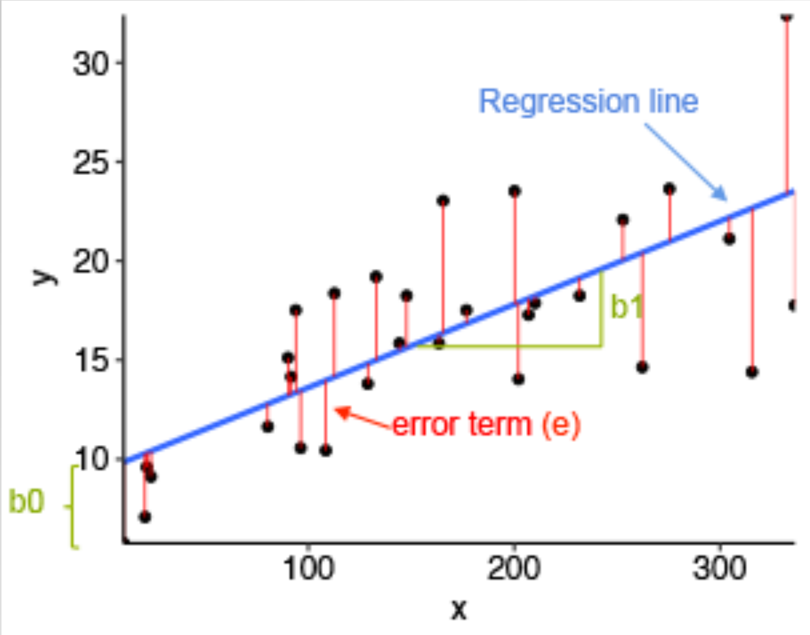
- Once you have a model which our learning algorithm has built using the training set, how can you say how good the model is?



Regression Evaluation

Regression Models Assessment

- For regression, the assessment of the model is quite simple A well-fitting regression model results in predicted values close to the observed data values.
- The **mean model**, which always predicts the average of the labels in the training data, generally would be used if there were no informative features
- The fit of a regression model being assessed should, therefore, be better than the fit of the mean model.



$y = b_1 x + b_0$ is another notation for $y = wx + b$

Regression Evaluation

- The most common metrics for evaluating regression learning problem predictions are:
  - Mean Absolute Error (MAE)
  - Mean Squared Error (MSE)
  - Root Mean Squared Error (RMSE)
  - Mean Absolute Percentage Error (MAPE)
  - R^2

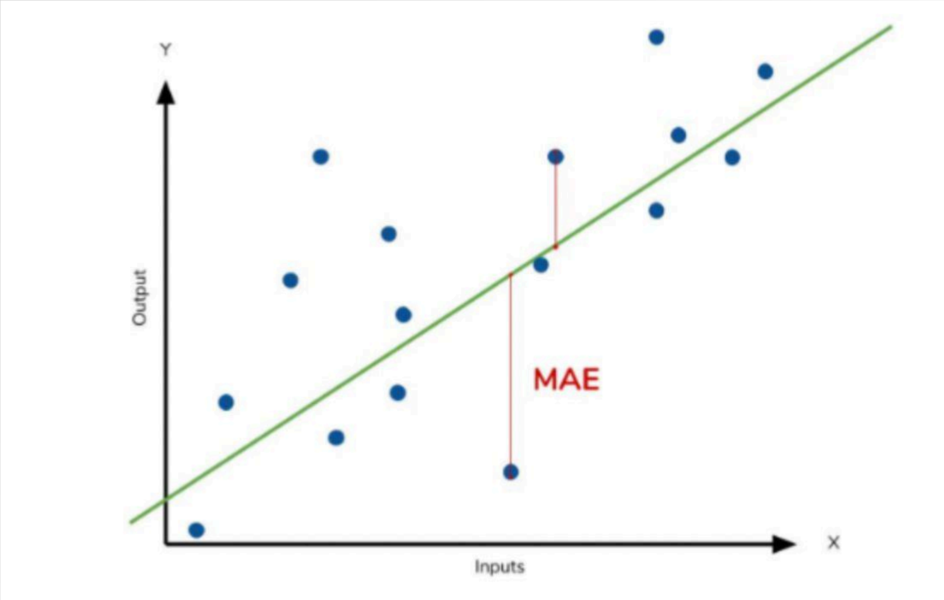
Day	Actual Temp	Predicted Temp	Error	Absolute Error	Squared Error	Percentage Error
1	20	22	-2	2	4	0.1
2	19	17	2	2	4	0.11
3	18	21	-3	3	9	0.17
4	19	18	1	1	1	0.05
5	18	18	0	0	0	0
6	20	18	2	2	4	0.1
7	21	21	0	0	0	0
8	19	18	1	1	1	0.05
9	20	23	-3	3	9	0.15
10	21	19	2	2	4	0.1
Total			**0**	**16**	**36**	**0.82**
Average			**0**	**1.6**	**3.6**	**0.08**

Mean Absolute Error (MAE)

- The Mean Absolute Error (MAE) is the average of the absolute differences **between predictions and actual values**
- It gives an idea of **how wrong the predictions were**
- The measure gives an idea of the magnitude of the error
  - But no idea of the direction (e.g., over or under predicting)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- y is the actual value
- $\hat{y}$ is the predicted value
- n is the number of data points



Mean Squared Error (MSE)

- If the MSE of the model on the test data is substantially higher than the MSE obtained on the training data, this is a sign of overfitting.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

  - y is the actual value
  - $\hat{y}$ is the predicted value
  - n is the number of data points

$$\mathrm{MSE} = \overbrace{\boxed{\frac{1}{n}\sum_{i=1}^{n}}}^{\textbf{Mean}}\Big(\overbrace{\boxed{Y_i - \hat{Y}_i}}^{\textbf{Error}}\Big)^{\overbrace{\boxed{2}}^{\textbf{Squared}}}$$

Root Mean Squared Error (RMSE)

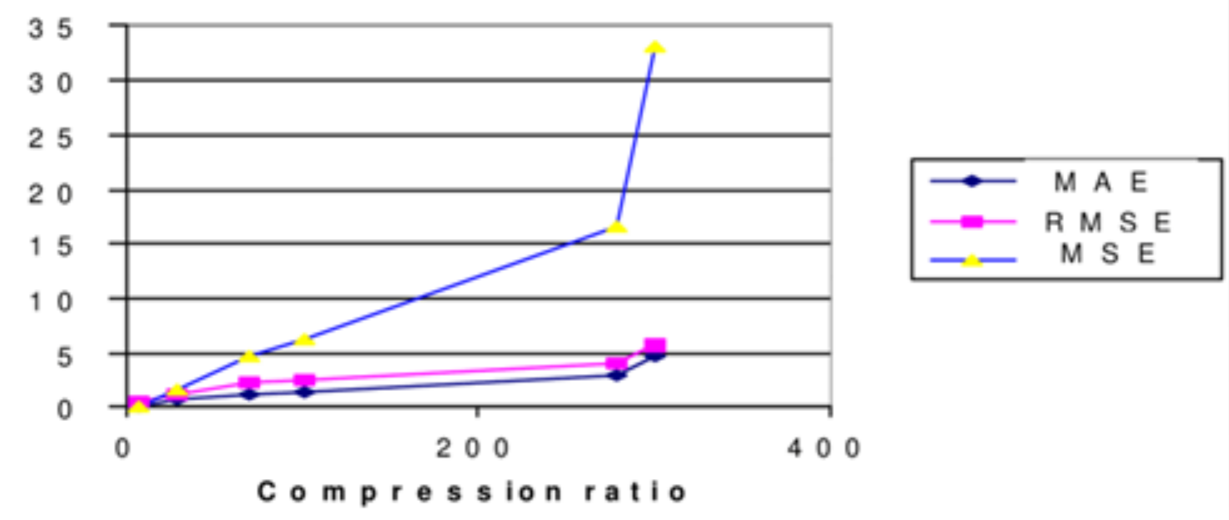
- The Root Mean Squared Error (RMSE) is much like the mean absolute error in that it provides a gross idea of the magnitude of the error
- Since the errors are squared before they are averaged, the RMSE gives a relatively **high weight to large errors**
  - This means the RMSE should be more useful when large errors are particularly undesirable

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

  - y is the actual value
  - $\hat{y}$ is the predicted value
  - n is the number of data points

MAE vs RSME vs MSE

MAE	MSE	RMSE
MAE is less biased for higher values	MSE is highly biased for higher values	RMSE reflects performance when dealing with large error values
MAE doesn't necessarily penalize large errors	MSE penalize large errors	RMSE penalize large errors



Mean Absolute Percentage Error (MAPE)

- The mean absolute percentage error (MAPE) is the mean or average of the absolute percentage errors of forecasts.

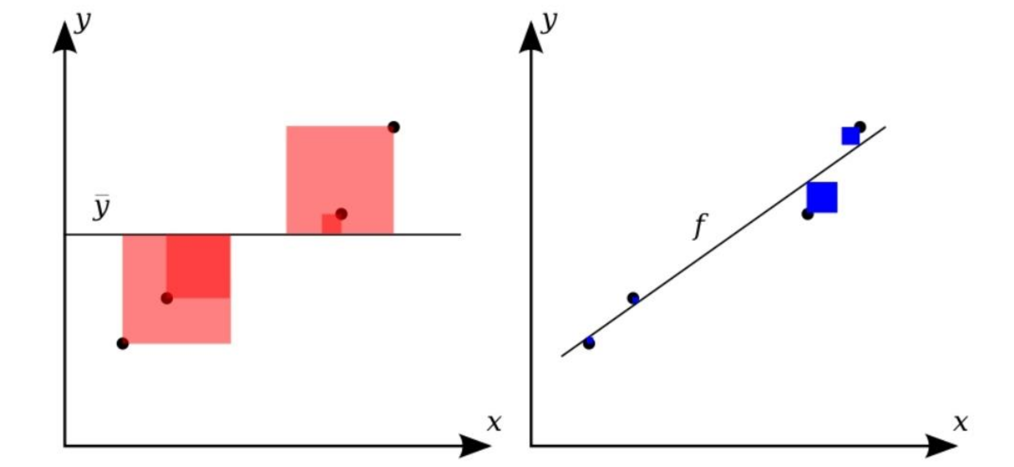
$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i}$$

- y is the actual value
- $\hat{y}$ is the predicted value
- n is the number of data points

R^2 Metric

- works only with linear models
- other models (RF - kNN - ...) get simulated (R^2) results

- The R^2 (or R Squared) metric indicates the goodness of fit of a set of predictions to the actual values
- A value between **0 and 1 for no-fit and perfect fit** respectively



$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

Squared sum error of regression line

Squared sum error of mean line

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$$

- y is the actual value
- $\hat{y}$ is the predicted value
- $\bar{y}$ is the mean of the y values
- n is the number of data points

Classification Evaluation

Classification Evaluation

- Many metrics can be used to evaluate the predictions for these problems
- Here are some:
    1. Classification Accuracy
    2. Confusion Matrix
    3. Precision, Recall, and F_1 score
    4. Area Under ROC Curve (AUC)

Classification Accuracy

– can be tricky when the dataset is unbalanced (unreliable)

- It is the number of **correct predictions made over all predictions made**
- The most common evaluation metric for classification problems

$$\text{Accuracy} = \frac{\sum \text{True positive} + \sum \text{True negative}}{\sum \text{Actual positive} + \sum \text{Actual negative}}$$

- This is only suitable when there is an equal number of observations in each class (**balanced dataset**) and all predictions and prediction errors are of equal importance

Confusion Matrix

- Confusion matrix allows visualization of the **performance** of an algorithm
- The name stems from the fact that it makes it easy to see if the system is confusing two classes
- Example:
  - Let's say, the model predicts two classes:
    - "spam"
    - "not_spam"

		PREDICTED	
		Positive	Negative
ACTUAL	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

	spam (predicted)	not_spam (predicted)
spam (actual)	23 (TP)	1 (FN)
not_spam (actual)	12 (FP)	556 (TN)

Confusion Matrix

TP stands for True Positive which indicates the number of positive examples classified accurately.

FN stands for False Negative which is the number of actual positive examples classified as negative.

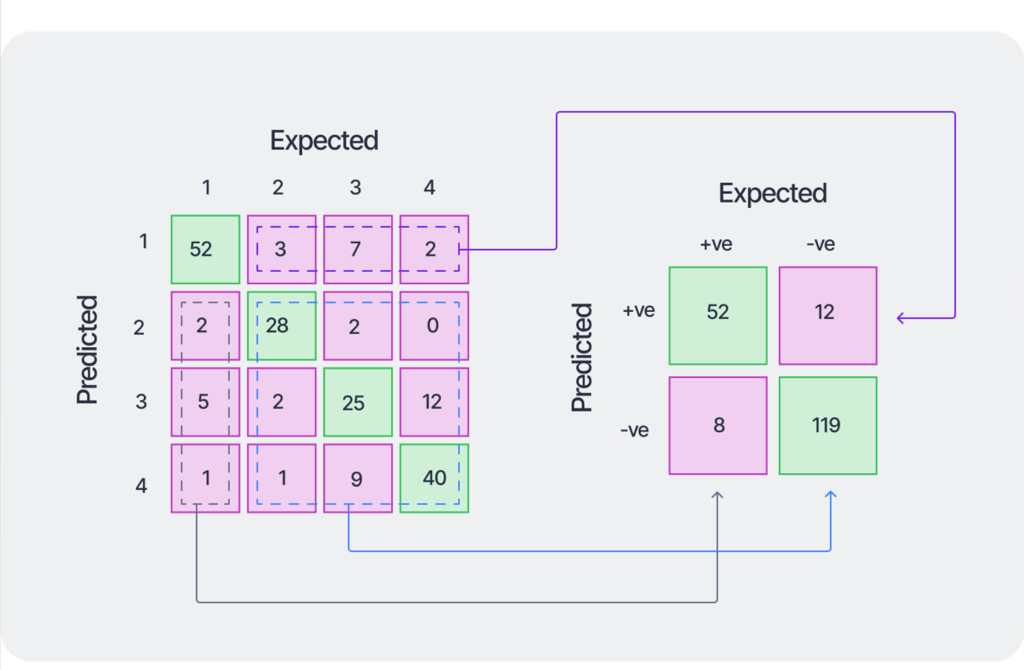
		PREDICTED	
		Positive	Negative
ACTUAL	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

FP stands for False Positive which is the number of actual negative examples classified as positive.

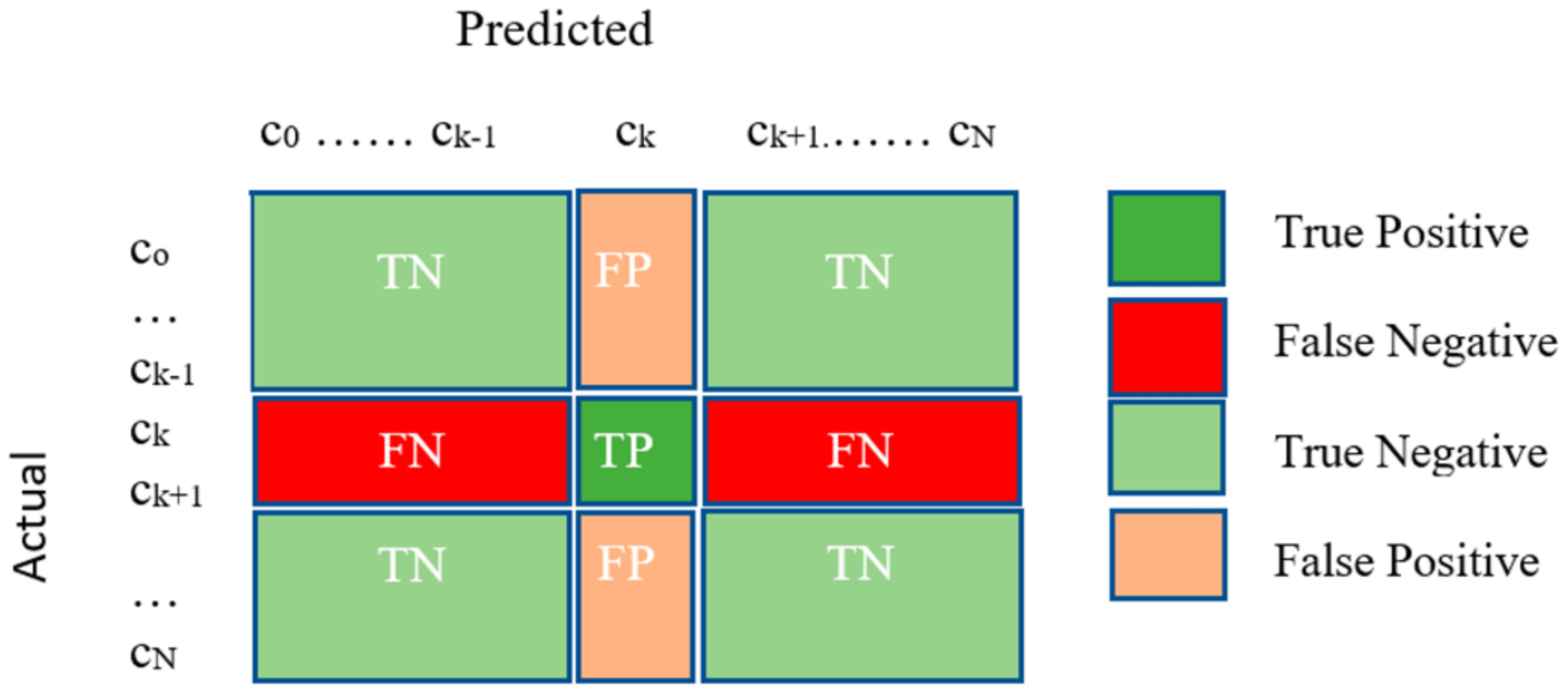
TN stands for True Negative which shows the number of negative examples classified accurately.

Multi-Class Confusion Matrix

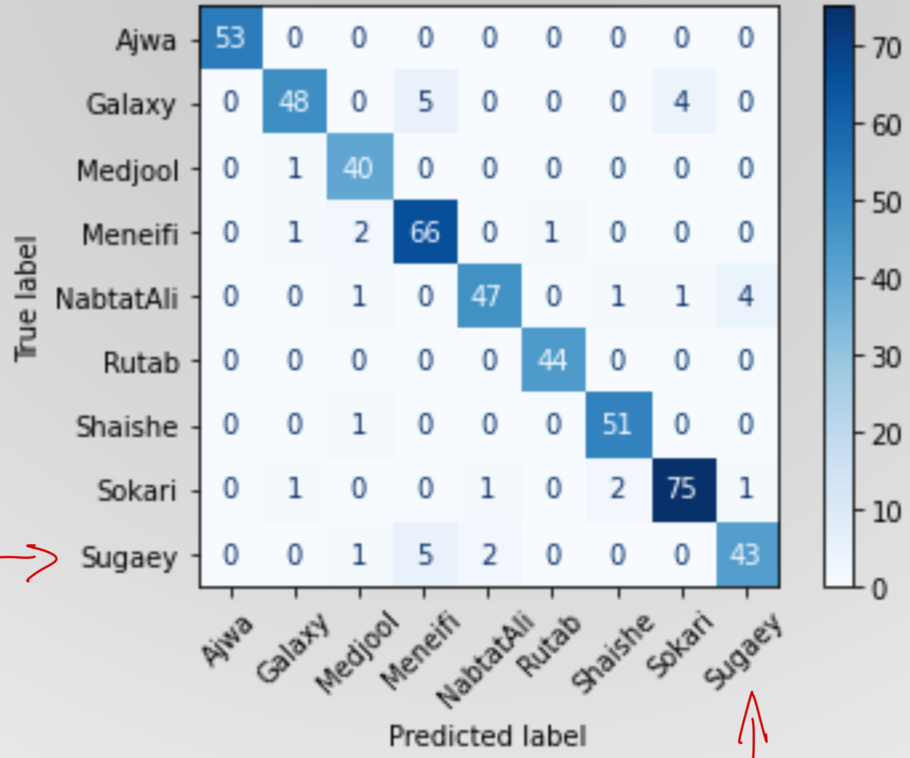
- The confusion matrix can be converted into a one-vs-all type matrix (binary-class confusion matrix) for calculating class-wise metrics like accuracy, precision, recall, etc.



Multi-Class Confusion Matrix



Multi-Class Confusion Matrix Example



Recall

Percision

Precision

- **Precision** , also called Positive predictive value (PPV), is the ratio of correctly predicted positive observations to the total predicted positive observations:

$$\text{Precision} = \frac{\sum \text{True positive}}{\sum \text{Predicted positive}} = \frac{TP}{TP + FP}$$

	حامل للجوال(predicted)	غير حامل للجوال(predicted)
حامل للجوال(Actual)	30(TP)	8(FN)
غير حامل للجوال(Actual)	7(FP)	17(TN)

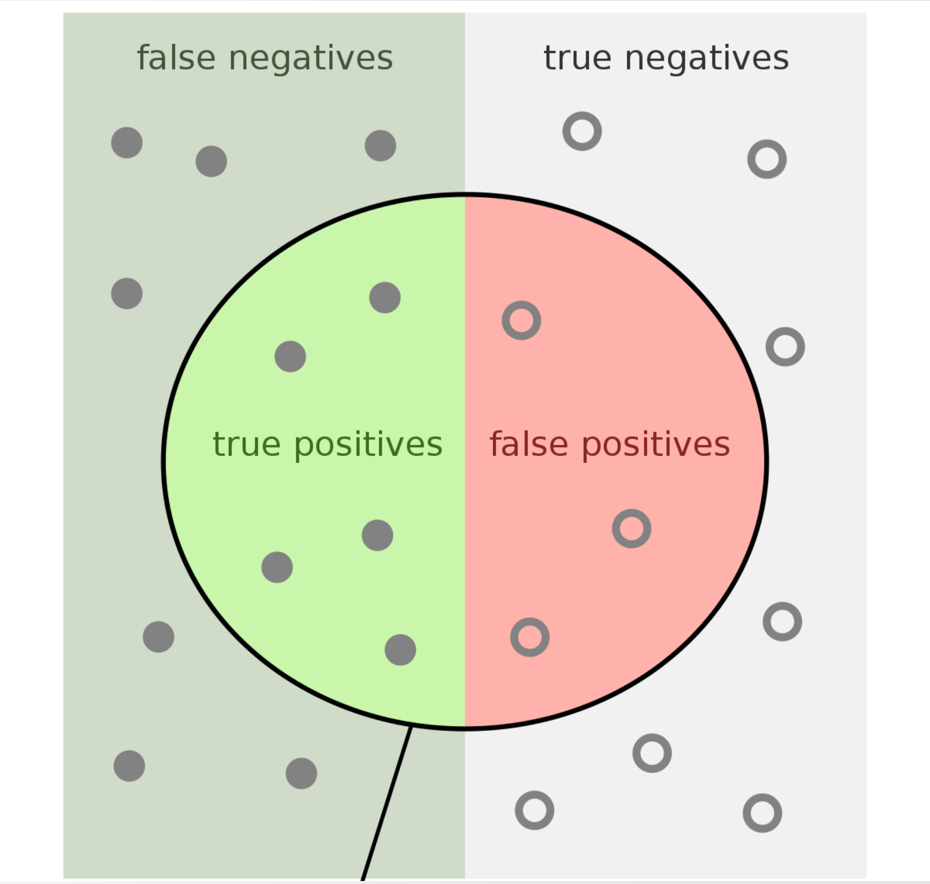
Recall

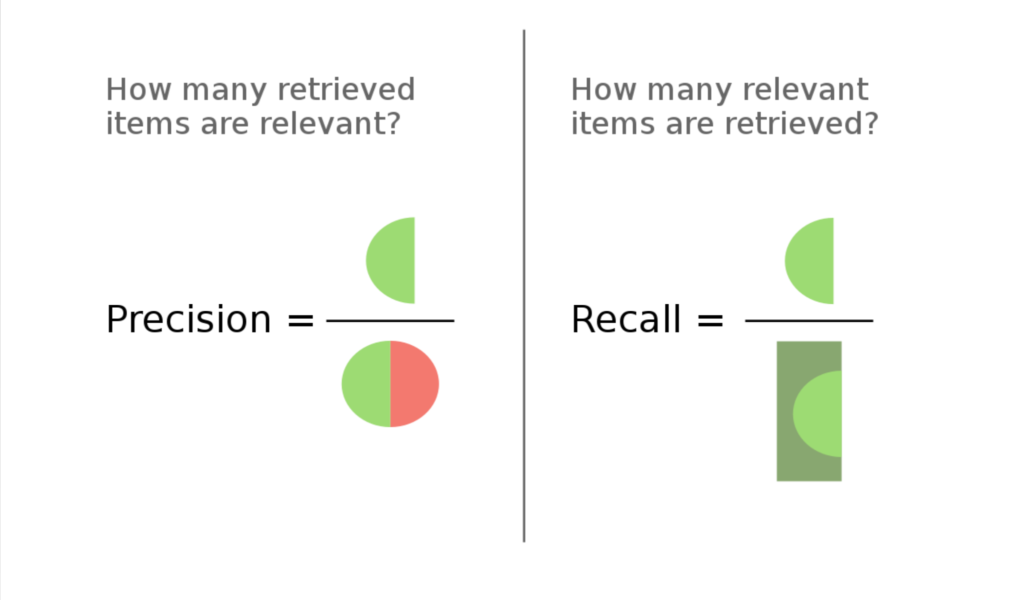
- **Recall** , also called Sensitivity or True Positive Rate (TPR), is the ratio of correctly predicted positive observations to the all observations in actual class:

$$\text{Recall} = \frac{\sum \text{True positive}}{\sum \text{Actual positive}} = \frac{TP}{TP + FN}$$

	(predicted)مصاب	(predicted)غير مصاب
(Actual)مصاب	10(TP)	3(FN)
(Actual)غير مصاب	5(FP)	12(TN)

Precision vs Recall





Precision vs Recall

- Precision is more important than Recall:
  - Detecting using mobile while driving to issue citations
  - Detecting spam emails
- Recall is more important than precision:
  - Detecting tumor in X-ray images

F_1 Score

- **F_1 Score** combines the precision and recall of a classifier into a single metric by taking their harmonic mean:

$$F_1 \text{ score } = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

- The harmonic mean tends strongly toward the least value

		PREDICTED	
		Positive	Negative
ACTUAL	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

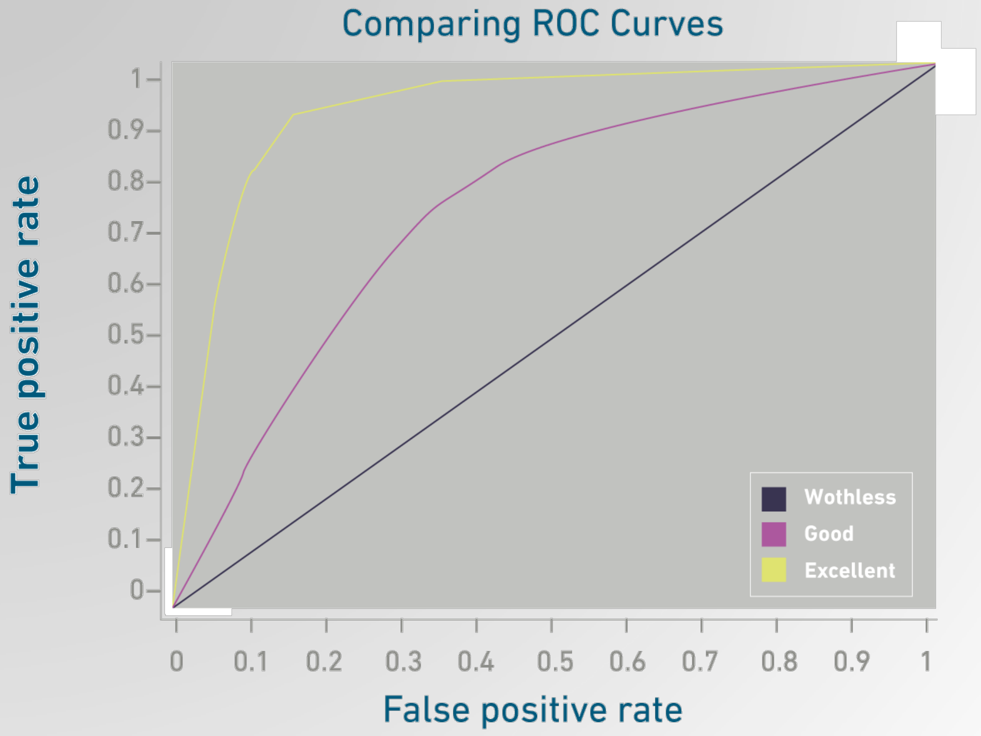
Classification Report in sklearn

	precision	recall	f1-score	support
Ajwa	1.00	1.00	1.00	53
Galaxy	0.94	0.84	0.89	57
Medjool	0.89	0.98	0.93	41
Meneifi	0.87	0.94	0.90	70
NabtatAli	0.94	0.87	0.90	54
Rutab	0.98	1.00	0.99	44
Shaishe	0.94	0.98	0.96	52
Sokari	0.94	0.94	0.94	80
Sugaey	0.90	0.84	0.87	51
accuracy			0.93	502
macro avg	0.93	0.93	0.93	502
weighted avg	0.93	0.93	0.93	502

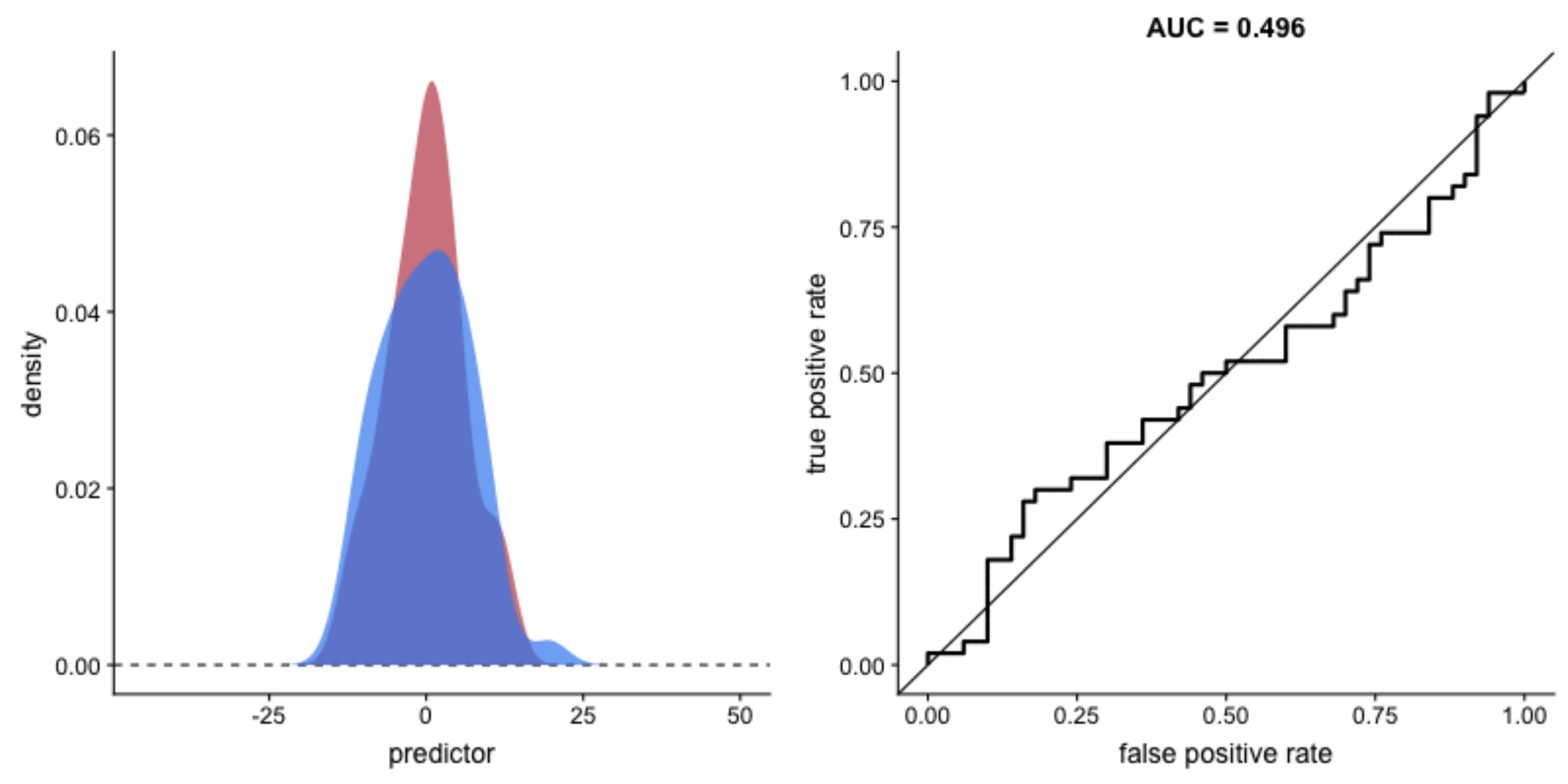
- support is the number of occurrences of each class in the test set
- Macro average: all classes equally contribute to the final averaged metric.
- Weighted average: each class's contribution to the average is weighted by its size (support).

Area Under ROC Curve (AUC)

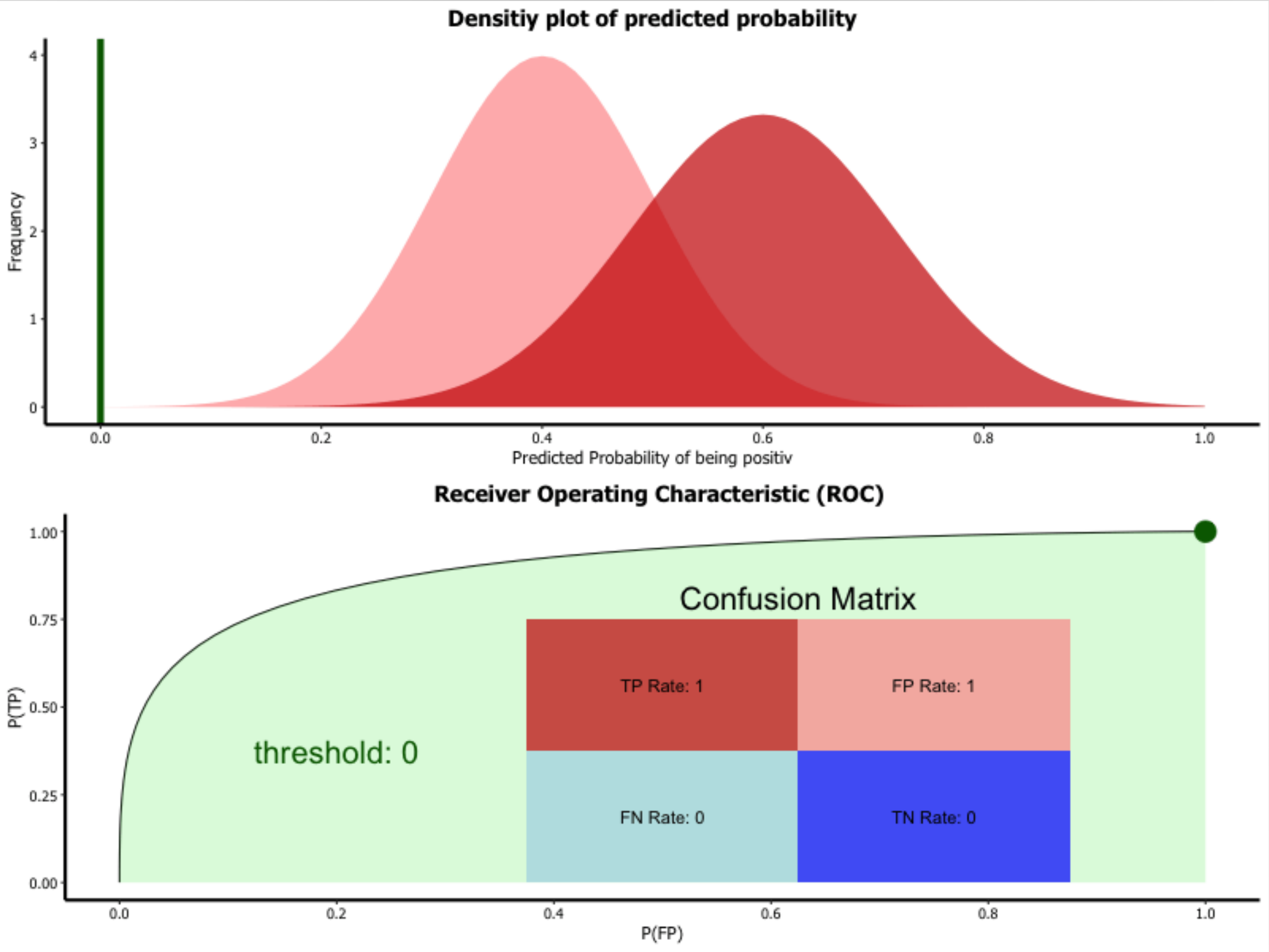
- The Receiver Operating Characteristic (ROC) curve is a graph showing the performance of a classification model at all classification thresholds
- ROC curves can **only** be used to assess classifiers that return some **confidence score** (or a probability) of prediction.
- AUC represents the capacity of the model to distinguish between positive and negative classes
  - **The 1.0 area represents a model that makes all predictions perfectly**
  - An area of 0.5 represents a random model



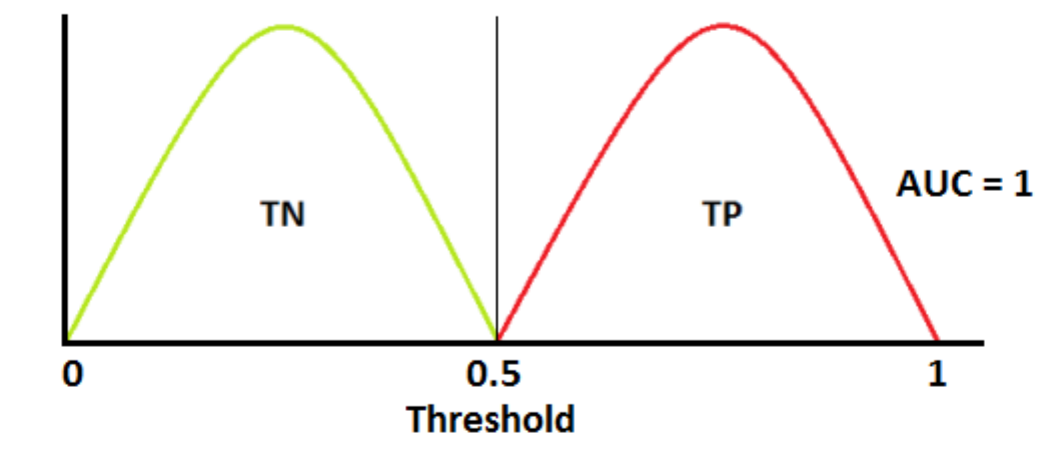
Area Under ROC Curve (AUC)

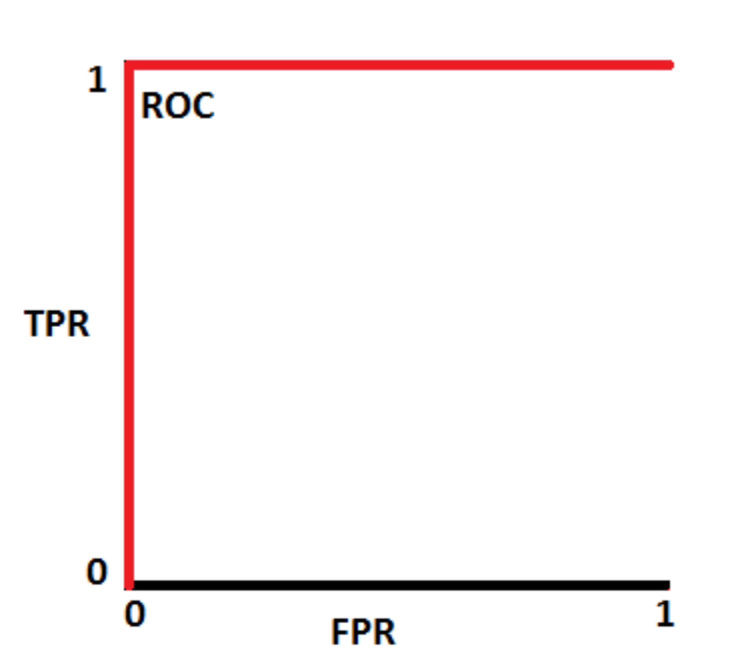


Area Under ROC Curve (AUC)



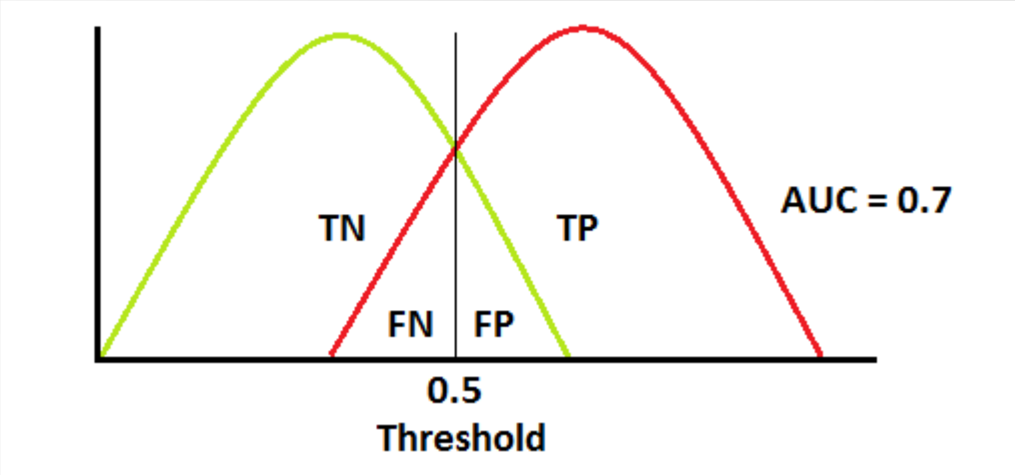
Area Under ROC Curve (AUC)

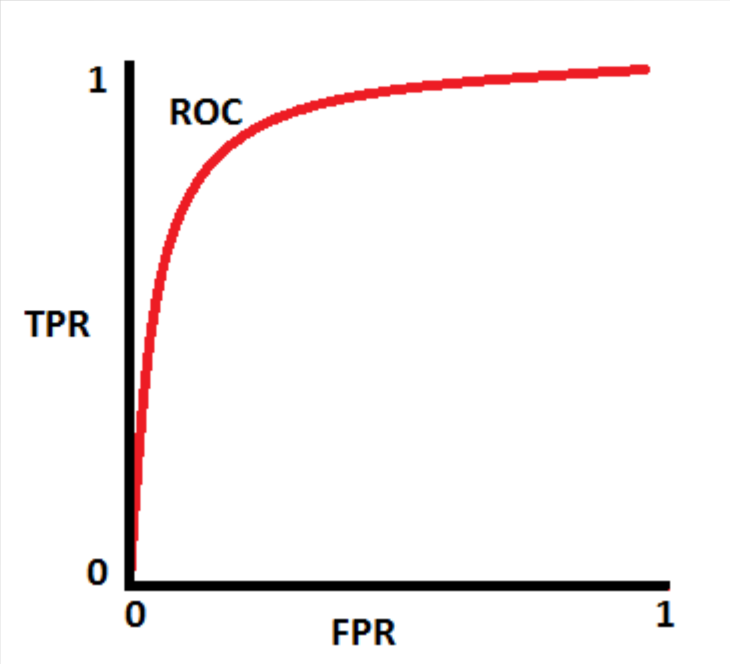




This is an ideal situation. When two curves don't overlap at all means model has an ideal measure of separability. It is perfectly able to distinguish between positive class and negative class.

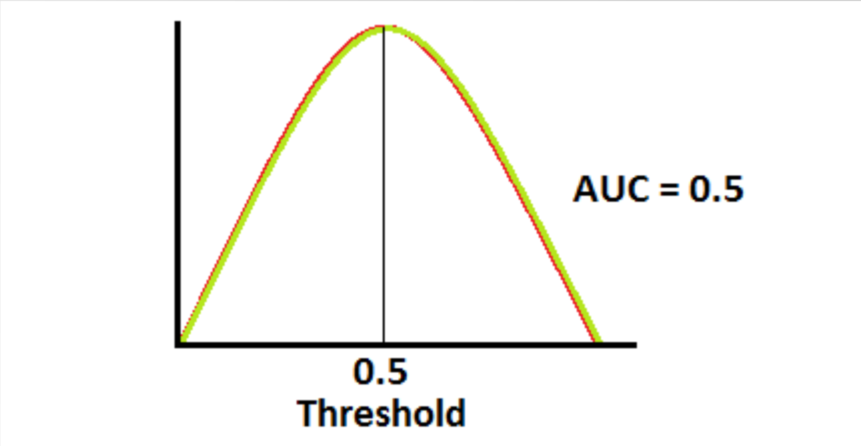
Area Under ROC Curve (AUC)

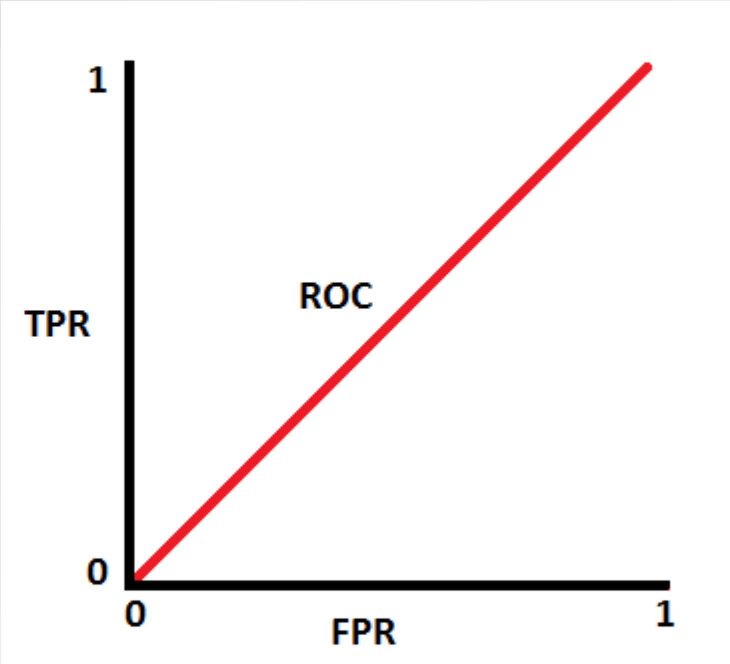




When AUC is 0.7, it means there is a 70% chance that the model will be able to distinguish between positive class and negative class.

Area Under ROC Curve (AUC)





When AUC is approximately 0.5, the model has no discrimination capacity to distinguish between positive class and negative class.