Machine Learning

Performance Evaluation of Classifiers

Dr. Pratyay Kuila

Dept. of Computer Science & Engineering
NIT Sikkim, Ravangla-737139

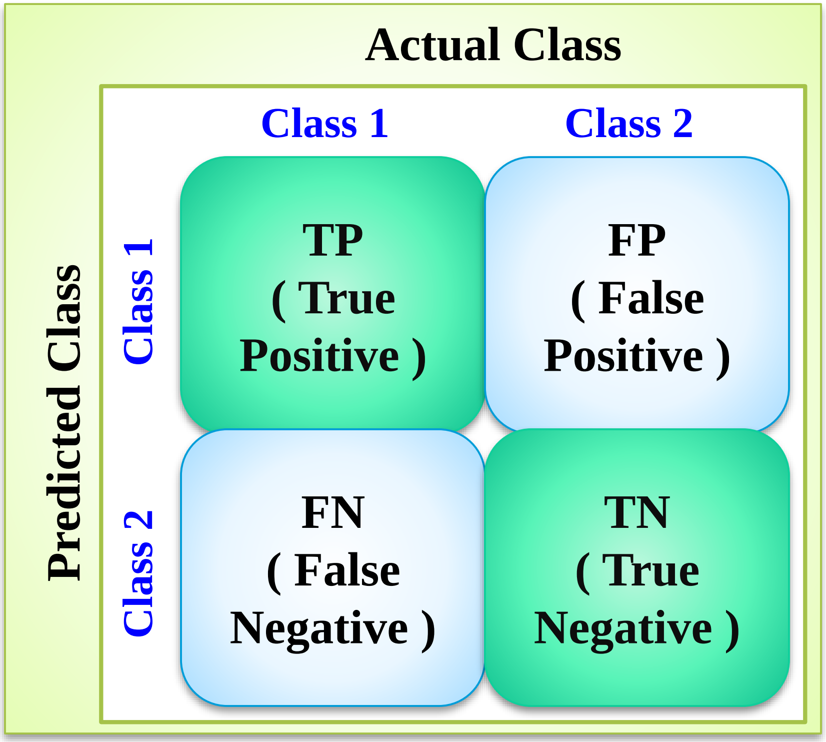
Performance Evaluation of Classifiers

- Performance indicators are very useful when the aim is to evaluate and compare different classification models or machine learning techniques.
- There are many metrics that come in handy to test the ability of any multi-class classifier.
- Many metrics are based on the Confusion Matrix, since it encloses all the relevant information about the algorithm and classification rule performance.
- The Confusion Matrix is a kind of error matrix. It visualizes the predictions for a classification task.

Confusion Matrix

- Let us consider a binary classification task. A confusion matrix that summarizes the number of instances predicted correctly or incorrectly by a classification model is shown.

- The terminologies are used :
  - True positive (TP): Corresponds to the number of positive examples correctly predicted by the classification model.
  - True negative (TN): Corresponds to the number of negative examples correctly predicted by the classification model.
  - False positive (FP): Corresponds to the number of negative examples wrongly predicted as positive by the classification model.



		Actual Class	
		Class 1	Class 2
Predicted Class	Class 1	TP (True Positive)	FP (False Positive)
	Class 2	FN (False Negative)	TN (True Negative)

  - False negative (FN): Corresponds to the number of positive examples wrongly predicted as negative by the classification model.

Confusion Matrix

- ***Example*:** Weather Forecasting: Let us consider a machine learning model that should predict if the weather will be rainy at the next day. So there are two classes.

Class 1:	**Rain**
Class 2:	**No Rain**

These are the possible results.

Predicted	Actual	Result
Rain	Rain	True Positive (TP)
Rain	No Rain	False Positive (FP)
No Rain	Rain	False Negative (FN)
No Rain	No Rain	True Negative (TN)

Confusion Matrix

- ***Example*****:** Weather Forecasting

In 20 examples the model predicted rain and it was correct

In 4 examples the model predicted rain and it was wrong

20	**4**
6	**10**

In 6 examples the model predicted no rain and it was wrong

In 10 examples the model predicted no rain and it was Correct

Confusion Matrix

- The counts in a confusion matrix can also be expressed in terms of percentages.
- The ***true positive rate*** (*TPR*) or ***sensitivity*** is defined as the fraction of positive examples predicted correctly by the model, i.e.,

$$\text{Sensitivity or } TPR = \frac{TP}{TP + FN}$$

- Similarly, the ***true negative rate*** (*TNR*) or ***specificity*** is defined as the fraction of negative examples predicted correctly by the model, i.e.,

$$\text{Specificity or } TNR = \frac{TN}{TN + FP}$$

- The ***false positive rate*** (*FPR*) is the fraction of negative examples predicted as a positive class, i.e.,

$$FPR = \frac{FP}{TN + FP}$$

- The ***false negative rate*** (*FNR*) is the fraction of positive examples predicted as a negative class, i.e.,

$$FNR = \frac{FN}{TP + FN}$$

Confusion Matrix

- ***Example***: The confusion matrix of a model applied to a set of 200 observations is given.

$$\text{Sensitivity}= \frac{TP}{TP+FN} = \frac{95}{95+7} = 0.93$$

$$\text{Specificity}= \frac{TN}{TN+FP} = \frac{94}{94+4} = 0.96$$

	Actual Class 1	**Actual Class 2**
Predicted Class 1	95	4
Predicted Class 2	7	94

- A *sensitivity* (true positive rate) of 1.0 implies that the model predicts all positive observation in a correct manner, i.e., the approach fails to make any false negative errors.

- A *specificity* of 1.0 or false positive rate of 0 indicates that the model predicts all negative observations accurately, i.e., the model does not predicts any false positive predictions.

- A model is considered good if it has a high true positive rate and low false positive rate at the same time.

Accuracy

- The accuracy can be defined as the percentage of the overall number of cases that were correctly predicted.

$$\text{Accuracy} = \frac{\boxed{TP} + \boxed{TN}}{\boxed{TP} + \boxed{FP} + \boxed{FN} + \boxed{TN}}$$

- In the confusion matrix (slide 5), the model should predict if the weather will be rainy or not at the next day. Accuracy measures how exactly our model makes predictions for both classes. We want to know the percentage out of all examples for which the weather prediction was correct.

- We divide the number of correct weather prediction by the total number of predictions:

$$\text{Accuracy} = \frac{\boxed{20} + \boxed{10}}{\boxed{20} + \boxed{4} + \boxed{6} + \boxed{10}} = 75\%$$

- Out of all 40 examples, in 30 examples the prediction of the model was correct. In this case 75% were correctly predicted, so our accuracy is 75%.

Precision

- *Precision* determines the fraction of instances that actually turns out to be positive in the group of test data instances those are declared as a positive class.
- The higher the precision is, the lower the number of false positive errors committed by the classifier.

$$\text{Precision, } p = \frac{TP}{TP + FP}$$

- The precision measures the percentage of all predicted rainy days that were correctly predicted as rainy.
- In our example (slide 5), we divide the number of correctly predicted rainy days by the number of all predicted rainy days.

$$\text{Precision, } p = \frac{20}{20 + 4} = 83\%$$

- Out of all examples that were predicted as rainy, 83% were correctly predicted.
- Therefore, when the model predicts rainy, there is a 83% chance that the prediction is correct.

Recall

- ***Recall*** measures the fraction of positive instances correctly predicted by the classifier.
- Classifiers with large recall have very few positive examples misclassified as the negative class. In fact, the value of recall is equivalent to the true positive rate.

$$\text{Recall, } r = \frac{TP}{TP + FN}$$

- The recall measures the percentage of all actual rainy days that were correctly predicted as rainy.
- In our example (slide 5), we divide the number of correctly predicted rainy days by the number of all actually rainy days.

$$\text{Recall, } r = \frac{20}{20 + 6} = 77\%$$

- Out of all examples that are actually labeled as rainy, 77% were correctly predicted.
- Therefore the model predicts 77% of all actually rainy days correctly as rainy.

Precision vs. Recall

$$\text{Precision, } p = \frac{TP}{TP + FP}$$

$$\text{Recall, } r = \frac{TP}{TP + FN}$$

- ***Precision*** measures the fraction of positive predictions correctly predicted by the classifier. ***Recall*** measures the fraction of positive examples from of the test data instances those are correctly predicted by the classifier.

- It is often possible to construct models that maximize one metric but not the other. Building a model that maximizes both *precision* and *recall* is challenging.

- *A model that declares every record to be the positive class will have a perfect recall, but very poor precision.*

- Conversely, *a model that declares positive class to few of the actual positive test records and declares negative class for all remaining records in the test set has very high precision, but low recall.*

Precision vs. Recall

$$\text{Precision}, p = \frac{TP}{TP + FP}$$

$$\text{Recall}, r = \frac{TP}{TP + FN}$$

- Consider the following cases:
  - ***Case* I:** Actual test data instances 100 with 60 positive and 40 negative class. All 100 instances are predicted as positive. Then, TP = 60, TN = 0, FP = 40 and FN = 0. Therefore, *Precision* = 60% and *Recall* = 100%.
  - ***Case* II:** Actual test data instances 100 with 60 positive and 40 negative class. Only 30 instances are predicted as positive and all are correctly done. Remaining 70 instances are predicted as negative class. Then, TP = 30, TN = 40, FP = 0 and FN = 60-30 = 30. Therefore, *Precision* = 100% and *Recall* = 50%.

- *Precision* and *recall* can be summarized into another metric known as the F_1-*Score*.

F_β-Score

- Combining *Precision* and *Recall* in F_β as:

$$F_\beta = \frac{(\beta^2+1)\times Precision \times Recall}{\beta^2 \times Precision + Recall} = \frac{(\beta^2+1)\times p \times r}{\beta^2 \times p + r}$$

- The parameter, β ∈ [0, ∞), enables the user to weigh the relative importance of the two criteria.

- If β > 1, then more weight is given to *recall*. If β < 1, then more weight is apportioned to *precision*.

- It would be easy to show that F_β converges to *recall* when β → ∞, and to *precision* when β = 0.

- Quite often, the engineer does not really know which of the two, *precision* or *recall*, is more important, and by how much. In that event, she prefers to work with the neutral value of the parameter, F_1.

F_β-Score

- The values of *precision* and *recall*, respectively, as $P = 0.40$ and $R = 0.29$.

- Using these numbers, we will calculate F_β for the following concrete settings of the parameter: $\beta = 0.2$, $\beta = 1$, and $\beta = 5$.

$$F_{0.2} = \frac{(0.2^2 + 1) \times 0.4 \times 0.29}{0.2^2 \times 0.4 + 0.29} = \frac{0.121}{0.306} = 0.39$$

$$F_1 = \frac{2 \times 0.4 \times 0.29}{0.4 + 0.29} = \frac{0.232}{0.69} = 0.336$$

$$F_5 = \frac{(5^2 + 1) \times 0.4 \times 0.29}{5^2 \times 0.4 + 0.29} = \frac{3.02}{10.29} = 0.29$$

F_1-Score

- F_1 Score is a Precision-Recall measure to evaluate the accuracy of a model.
- It is calculated as follows:

$$F_1 = 2.\frac{Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{2 \times TP + FP + FN}$$

- In principle, F_1 represents a harmonic mean between recall and precision, i.e.,

$$F_1 = \frac{2}{\frac{1}{r} + \frac{1}{p}}$$

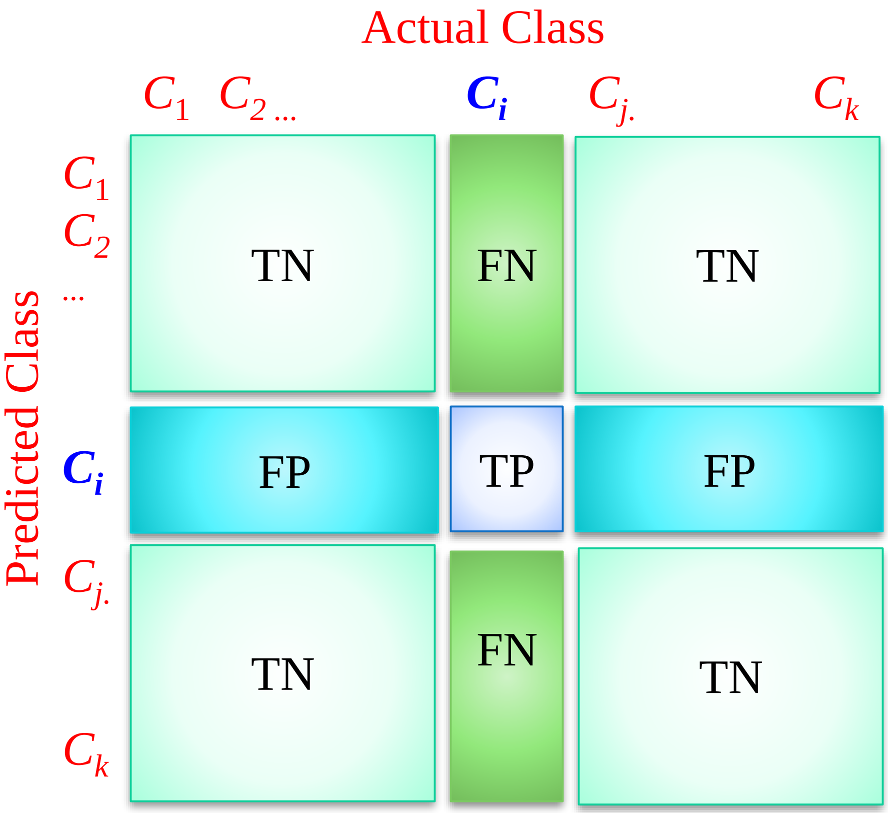
- The F_1 Score has a range from 0 to 1.
- In our previous example (slide 5), *Precision*, p = 0.83 and *Recall*, r = 0.77.
- In this case we would get the following result for the F_1 Score.

$$F_1 = 2.\frac{0.83 \times 0.77}{0.83 + 0.77} = 0.799$$

- Therefore, the F_1 score is 79.9%.
- The harmonic mean of two numbers x and y tends to be closer to the smaller of the two numbers.
- Hence, a high value of F_1-score ensures that both *precision* and *recall* are reasonably high.

Confusion Matrix for Multiclass Classifier

- ➢ The previous examples are for a binary classification with only 2 outputs so we got a 2 × 2 matrix.
- ➢ What if the outputs are greater than 2 classes i.e., Multi-class classification? How to calculate TP, FN, FP, TN?
  - ▪ TP: the actual value and predicted value are the same.
  - ▪ FN: the sum of values of corresponding columns except for the TP.
  - ▪ FP: the sum of values of the corresponding rows except for the TP.
  - ▪ TN: the sum of values of all columns and rows except the values of that class that we are calculating the values for.



Actual Class

C_1 C_2 ... C_i $C_{j.}$ C_k

Predicted Class

C_1 C_2 ... C_i $C_{j.}$ C_k

TN	FN	TN
FP	TP	FP
TN	FN	TN

Confusion Matrix for Multiclass Classifier

- Consider a classifier for the dataset which has 3 flowers as outputs or classes as *Versicolor*, *Virginia*, *Setosa*.
- As, the dataset has 3 classes hence we get a 3 × 3 confusion matrix.

		Actual		
		Setosa	***Versicolor***	***Virginia***
Predicted	***Setosa***	16 (Cell 1)	0 (Cell 2)	0 (Cell 3)
	Versicolor	0 (Cell 4)	17 (Cell 5)	1 (Cell 6)
	Virginia	0 (Cell 7)	0 (Cell 8)	11 (Cell 9)

Confusion Matrix for Multiclass Classifier

- Let us calculate the TP, TN, FP, FN values for the class ***Setosa*.**
  - TP: the actual value and predicted value are the same.

    Value of cell 1 only. TP = 16.
  - FN: the sum of values of corresponding columns except for the TP.

    FN = (cell 4 + cell 7) = 0.
  - FP: the sum of values of the corresponding rows except for the TP. FP = (cell 2 + cell 3) = 0.
  - TN: the sum of values of all columns and rows except the values of that class that we are calculating the values for. TN = (cell 5 + cell 6 + cell 8 + cell 9) = 17 + 1 +0 + 11 = 29

		Actual		
		Setosa	***Versicolor***	***Virginia***
Predicted	***Setosa***	16 (Cell 1)	0 (Cell 2)	0 (Cell 3)
	Versicolor	0 (Cell 4)	17 (Cell 5)	1 (Cell 6)
	Virginia	0 (Cell 7)	0 (Cell 8)	11 (Cell 9)

Confusion Matrix for Multiclass Classifier

- Let us calculate the TP, TN, FP, FN values for the class ***Versicolor*.**
  - TP: the actual value and predicted value are the same.
    Value of cell 5 only. TP = 17.
  - FN: the sum of values of corresponding columns except for the TP.
    FN = (cell 2 + cell 8) = 0.
  - FP: the sum of values of the corresponding rows except for the TP.
    FP = (cell 4 + cell 6) = 1.
  - TN: the sum of values of all columns and rows except the values of that class that we are calculating the values for. TN = (cell 1 + cell 3 + cell 7 + cell 9) = 16 + 0 +0 + 11 = 27

		Actual		
		Setosa	***Versicolor***	***Virginia***
Predicted	***Setosa***	16 (Cell 1)	0 (Cell 2)	0 (Cell 3)
	Versicolor	0 (Cell 4)	17 (Cell 5)	1 (Cell 6)
	Virginia	0 (Cell 7)	0 (Cell 8)	11 (Cell 9)

Confusion Matrix for Multiclass Classifier

- Consider the following confusion matrix for a three class classifier.
- If we take class *Apple*, then following are the values of the metrics from the confusion matrix.

TP = 7 **TN = (2+3+2+1) = 8** **FP = (8+9) = 17** **FN = (1+3) = 4**

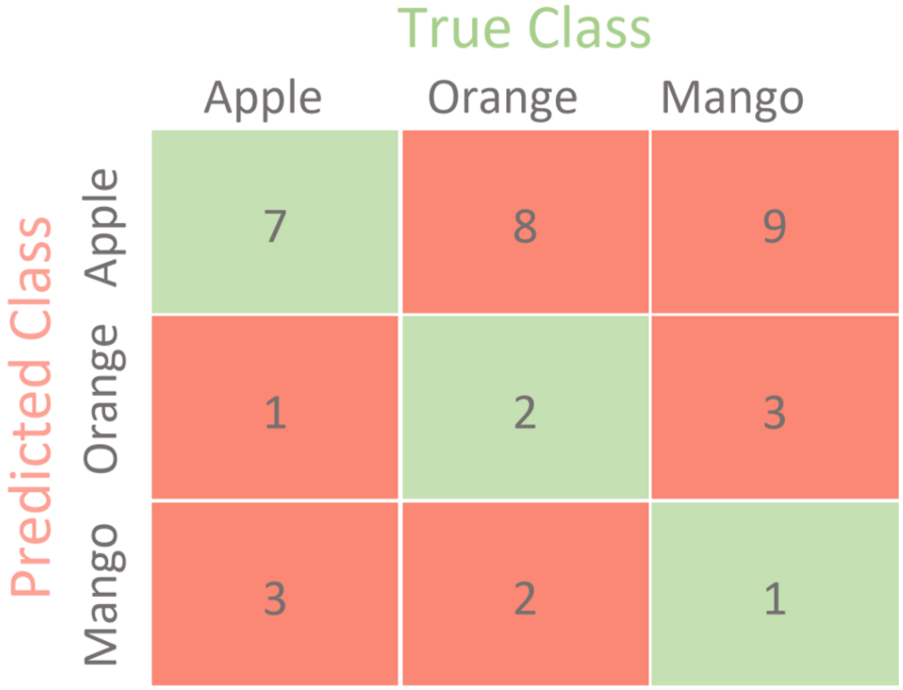
- Now we can calculate the performance measures for class Apple.

  Precision = 7/(7+17) = 0.29

  Recall = 7/(7+4) = 0.64

  F1-score = 0.40

- Similarly, calculate for the other classes.



		True Class		
		Apple	Orange	Mango
Predicted Class	Apple	7	8	9
	Orange	1	2	3
	Mango	3	2	1

Confusion Matrix for Multiclass Classifier

		Actual			
		A	***B***	***C***	***D***
Predicted	***A***	52	3	7	2
	B	2	28	2	0
	C	5	2	25	12
	D	1	1	9	40

Class	Precision (%)	Recall (%)	F1-Score (%)
A	81.25	86.67	83.87
B	87.50	82.35	84.85
C	56.82	58.14	57.47
D	78.43	74.07	76.19

- The confusion matrix and its performance measures are given.
- The *unweighted means* of the measures are:

  Macro Precision = 76.00%

  Macro Recall = 75.31%

  Macro F1-Score = 75.60%

Confusion Matrix for Multiclass Classifier

		Actual			
		A	***B***	***C***	***D***
Predicted	***A***	52	3	7	2
	B	2	28	2	0
	C	5	2	25	12
	D	1	1	9	40

Class	Precision (%)	Recall (%)	F1-Score (%)
A	81.25	86.67	83.87
B	87.50	82.35	84.85
C	56.82	58.14	57.47
D	78.43	74.07	76.19

- Weighted Precision =

$$\frac{81.25 \times 60 + 87.50 \times 34 + 56.82 \times 43 + 78.43 \times 54}{60 + 34 + 43 + 54}\% = 76.07\%$$

- Weighted Recall =

$$\frac{86.67 \times 60 + 82.35 \times 34 + 58.14 \times 43 + 74.07 \times 54}{60 + 34 + 43 + 54}\% = 75.92\%$$

- Weighted F1-Score =

$$\frac{83.87 \times 60 + 84.85 \times 34 + 57.47 \times 43 + 76.19 \times 54}{60 + 34 + 43 + 54}\% = 75.93\%$$

The Receiver Operator Characteristic (ROC) Curve

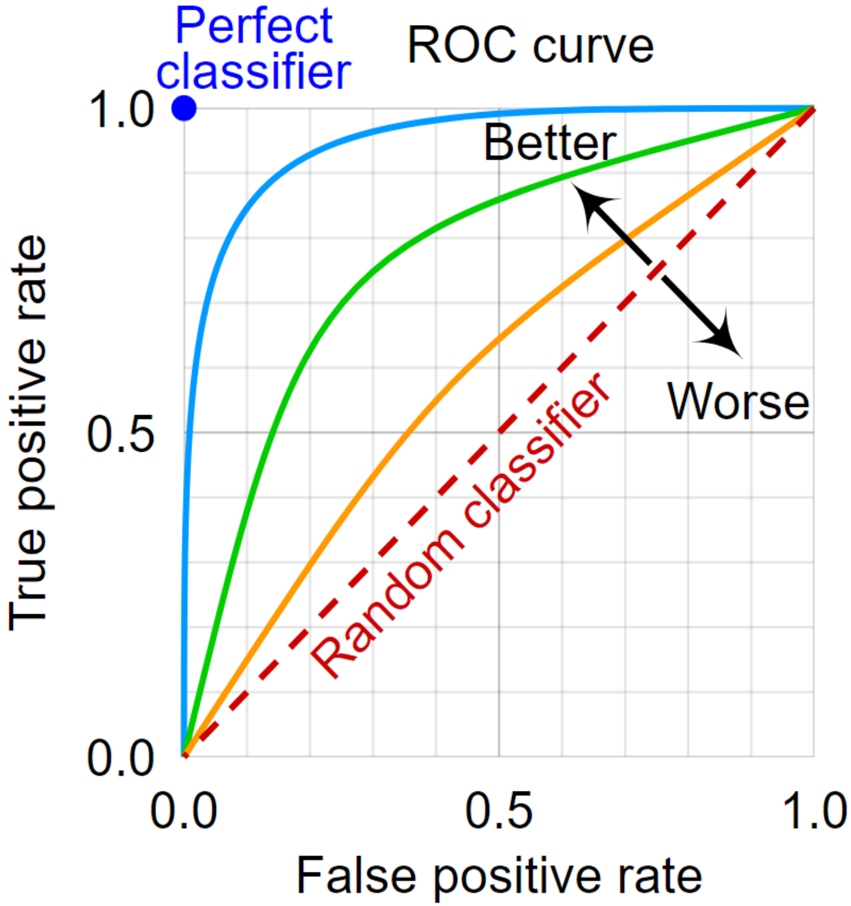
- A receiver operator characteristic (ROC) curve is a graphical approach for displaying the tradeoff between *true positive rate* and *false positive rate* of a classifier.

- In an ROC curve, the *true positive rate* (*TPR*) is plotted along the *y* axis and the *false positive rate* (*FPR*) is plotted along the *x* axis.

  $TPR = TP/(TP + FN).$

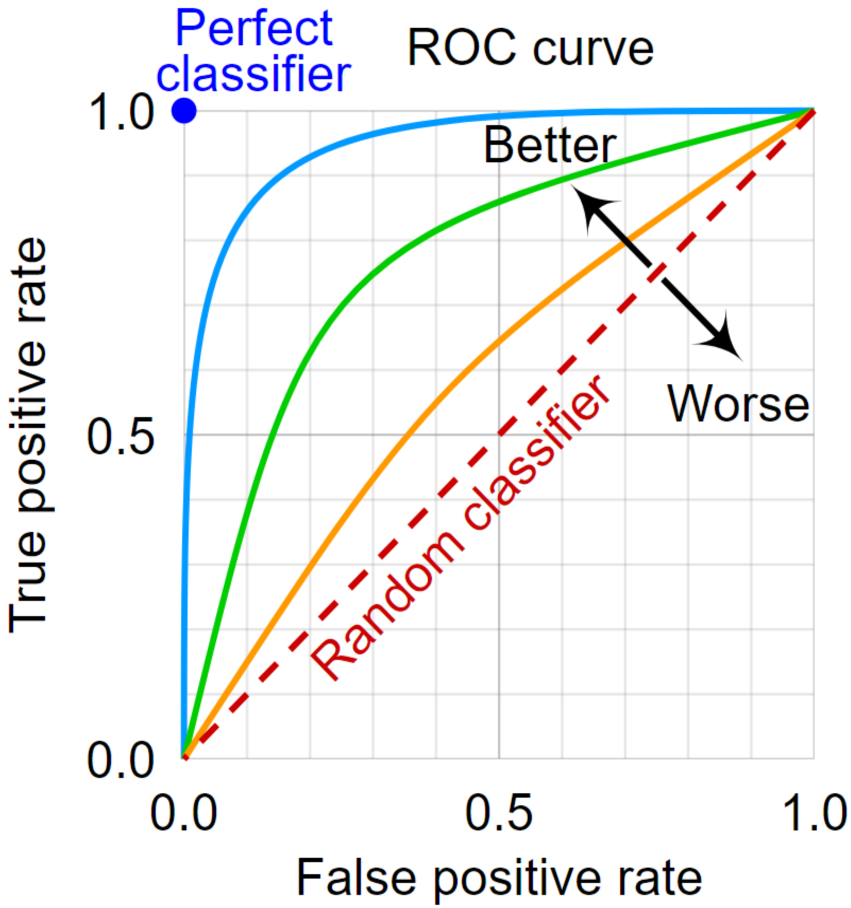
  $FPR = FP/(TN + FP)$

- Each point along the curve corresponds to one of the models induced by the classifier.



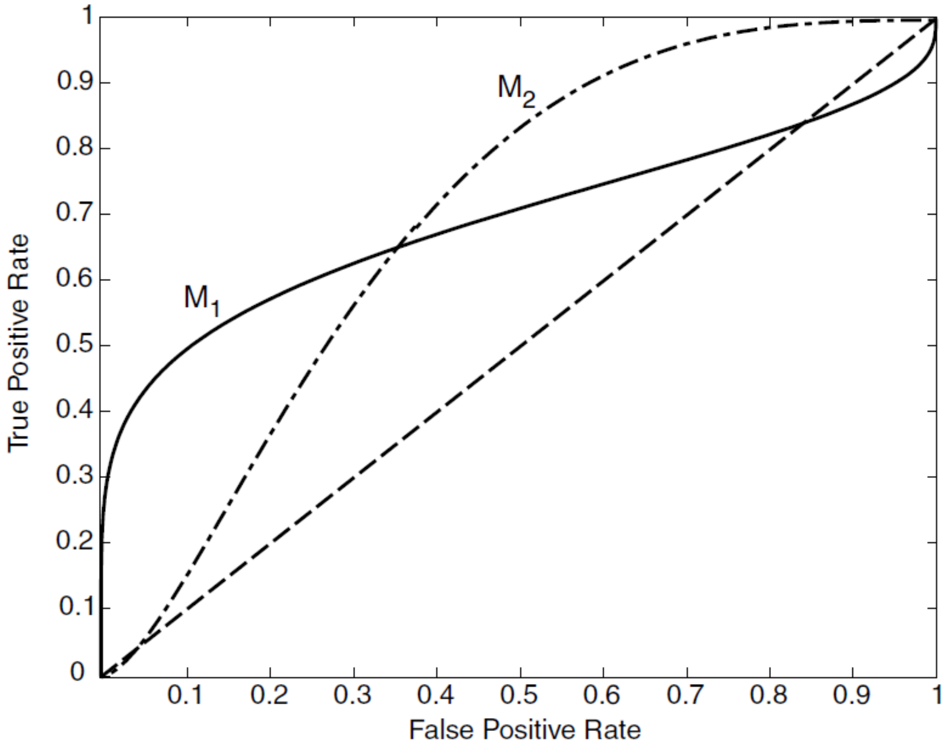
The Receiver Operator Characteristic (ROC) Curve

- There are several critical points along an ROC curve that have well-known interpretations:
  - (TPR=0, FPR=0): The lower left point (0,0). Model predicts every instance to be a negative class.
  - (TPR=1, FPR=1): Model predicts every instance to be a positive class.
  - (TPR=1, FPR=0): The ideal model.



The Receiver Operator Characteristic (ROC) Curve

- Figure shows the ROC curves for a pair of classifiers, M_1 and M_2.



- A good classification model should be located as close as possible to the upper left corner of the diagram.

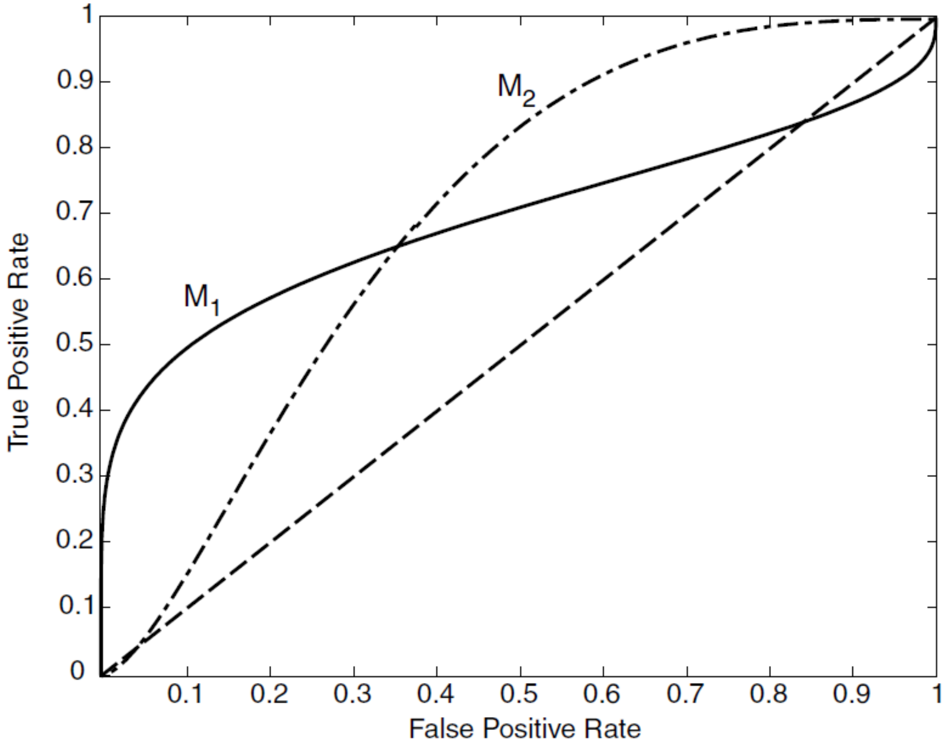
- A model that makes random guesses should reside along the main diagonal, connecting the points (TPR = 0, FPR = 0) and (TPR = 1, FPR = 1). Random guessing means that a record is classified as a positive class with a fixed probability p, irrespective of its attribute set.

Dr. Pratyay Kuila, *NIT Sikkim*

The Receiver Operator Characteristic (ROC) Curve

- For example, consider a data set that contains *n*+ positive instances and *n*− negative instances.

- The random classifier is expected to correctly classify *pn*+ of the positive instances and to misclassify *pn*− of the negative instances.

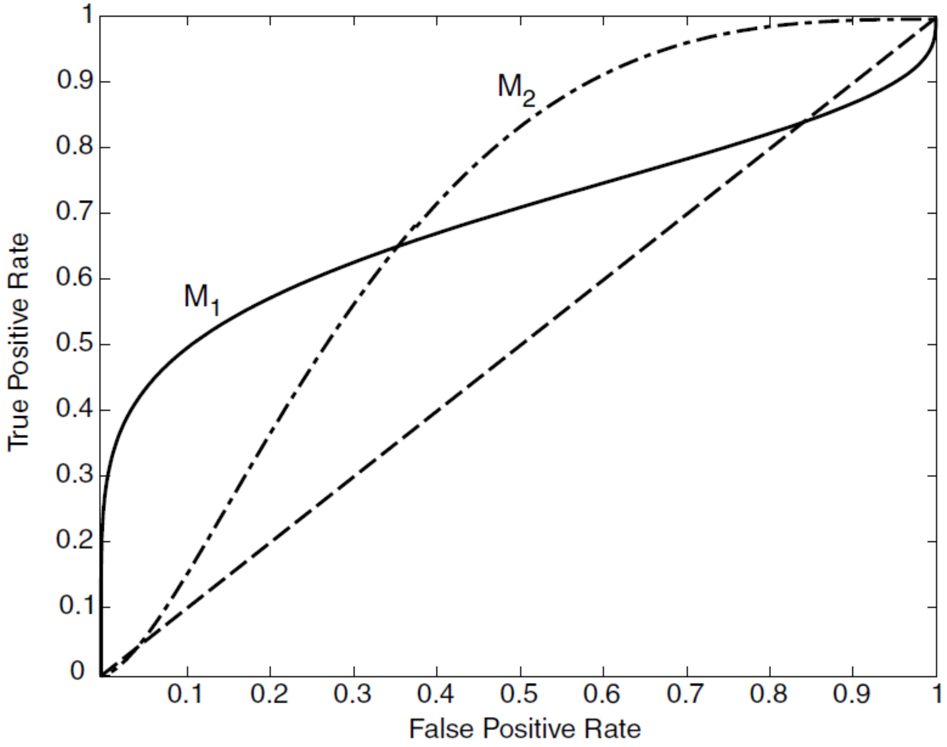


- Therefore, the *TPR* of the classifier is (*pn*+)/*n*+ = *p*, while its *FPR* is (*pn*−)/*n*− = *p*.

- Since the *TPR* and *FPR* are identical, the ROC curve for a random classifier always reside along the main diagonal.

The Receiver Operator Characteristic (ROC) Curve

- An ROC curve is useful for comparing the relative performance among different classifiers.

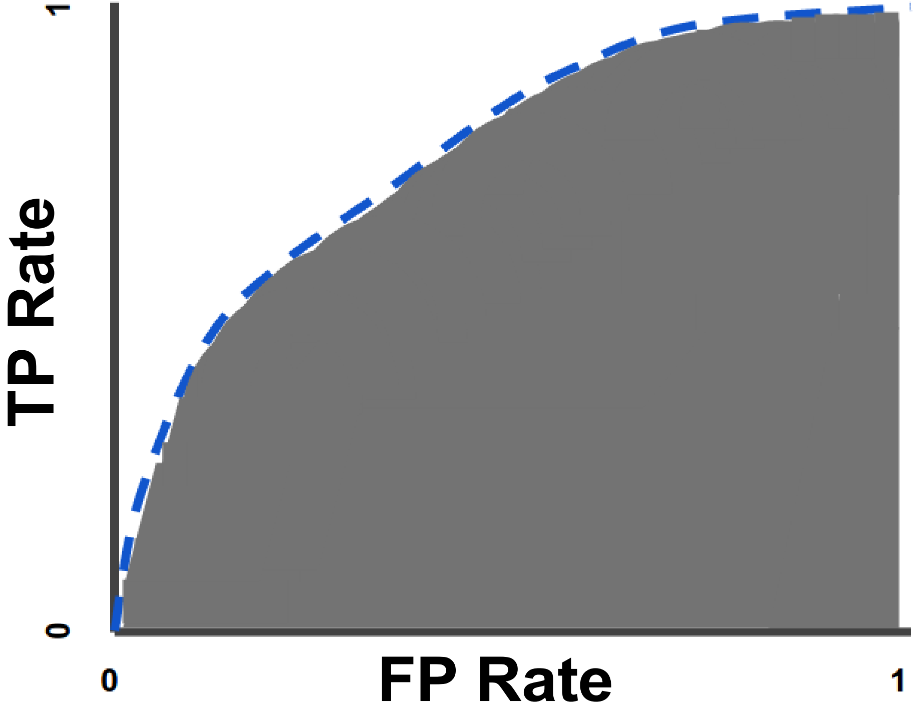


- In the Figure, M_1 is better than M_2 when *FPR* is less than 0.36, while M_2 is superior when *FPR* is greater than 0.36.
- Clearly, neither of these two classifiers dominates the other.

The Receiver Operator Characteristic (ROC) Curve

- The area under the ROC curve (AUC) provides another approach for evaluating which model is better on average.

- If the model is perfect, then its area under the ROC curve would equal 1.



- If the model simply performs random guessing, then its area under the ROC curve would equal 0.5.

- A model that is strictly better than another would have a larger area under the ROC curve.

References:

1. “Applied Machine Learning” by M.Gopal, McGraw Hill.
2. “Introduction to Data Mining” by Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson.
3. “Metrics for Multi-Class Classification: An Overview” by Margherita Grandini, Enrico Bagli, Giorgio Visani, *A White Paper*.