Machine Learning

|ROC Curve

Prof. Dr. Mostafa Elhosseini

https://youtube.com/@ProfElhosseiniTechVerse

Agenda

- True Positive Rate (TPR) – Sensitivity - Recall
- False Positive Rate (FPR) - Fall-out
- True Negative Rate (TNR) – Specificity
- The ROC Curve
- Origins - Signal detection theory

True Positive Rate (TPR) – Sensitivity - Recall

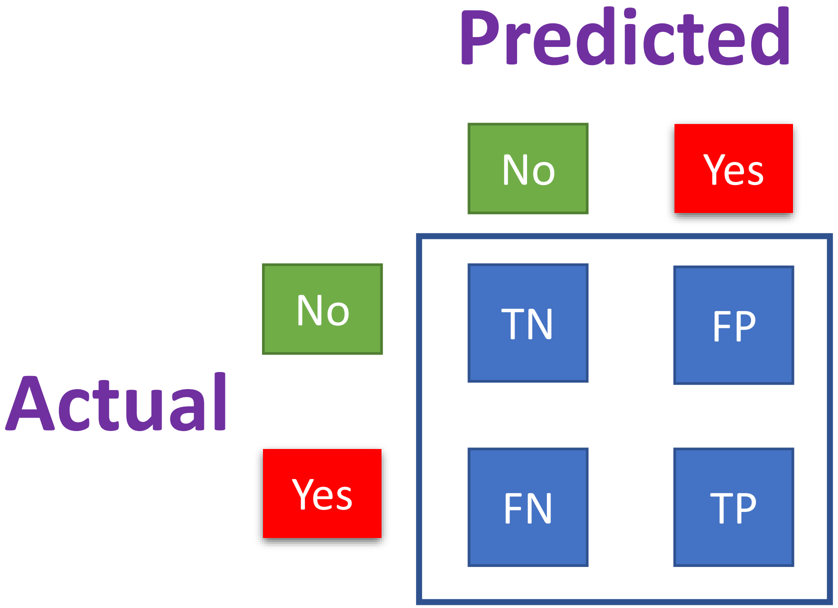
- which is the probability that an **actual positive will test positive**
- Recall tells us how many positive samples were detected out of all actual positive samples

- $$\boldsymbol{TPR} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}}$$

		Predicted	
		No	Yes
Actual	No	TN	FP
	Yes	FN	TP

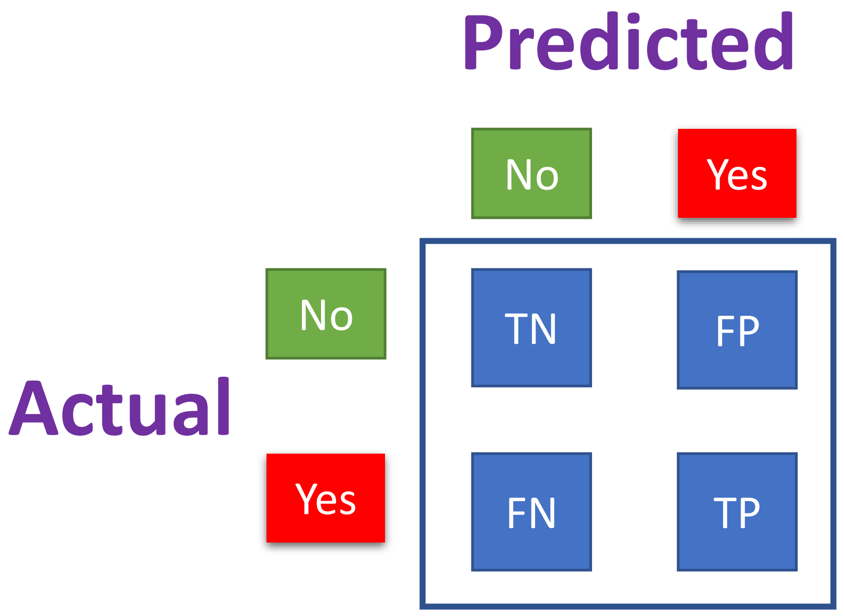
True Negative Rate (TNR) – Specificity

- which is the probability that an **actual negative will test negative**
- The ratio of negative instances that are correctly classified as negative
- $TNR = \frac{TN}{TN+FP}$



False Positive Rate (FPR)

- FPR is the **probability that a false alarm will be raised**
- FPR is the ratio of negative instances that are incorrectly classified as positive
- FPR aka **fall-out**
- $FPR = \frac{FP}{TN+FP}$
- FPR = 1-TNR



The ROC Curve

- The Receiver Operating Characteristic (ROC) curve is another common tool used with binary classifiers.
- ROC curve plots the true positive rate (another name for recall) against the false positive rate FPR
- The ROC curve plots **sensitivity** (recall) versus **1 – specificity**

The ROC Curve

- Signal detection theory
- World War II
- Analysis of radar signals.
- ROC curve was used to characterize the performance of a radar receiver operator in distinguishing between the presence or absence of an enemy target.
- The operator had to decide whether a blip on the radar screen represented an enemy target (signal) or noise (no signal).

The ROC Curve

- The ROC curve in signal detection theory **plots the probability of detecting a true signal** (True Positive Rate) **against the probability of a false alarm** (False Positive Rate) for different threshold settings.
- The concepts from signal detection theory are analogous to the performance of a binary classifier

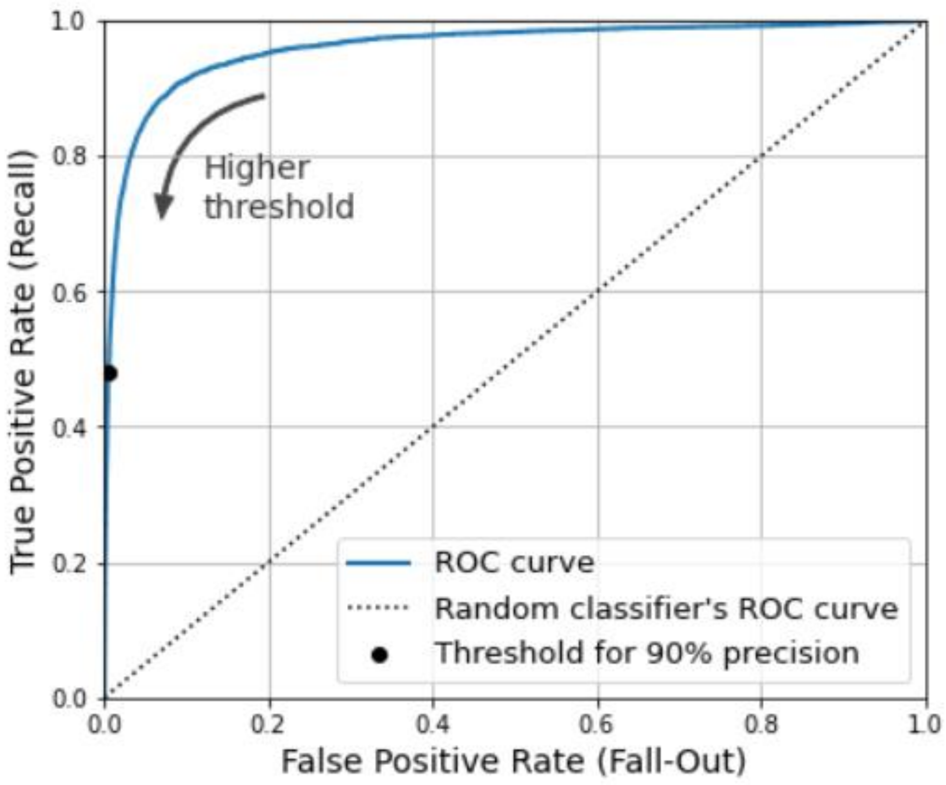
The ROC Curve

- By plotting the TPR against the FPR for different classification thresholds, the ROC curve provides a visual representation of the classifier's performance.
- A perfect classifier would have an ROC curve that passes through the **top-left corner**, indicating a TPR of 1 and an FPR of 0.
- A **random** classifier would have an ROC curve that follows the **diagonal line** from the bottom-left to the top-right corner.

The ROC Curve

- To plot the ROC curve, you first use the roc_curve() function to compute the TPR and FPR for various threshold values

```
from sklearn.metrics import roc_curve

fpr, tpr, thresholds = roc_curve(y_train_5, y_scores)
```

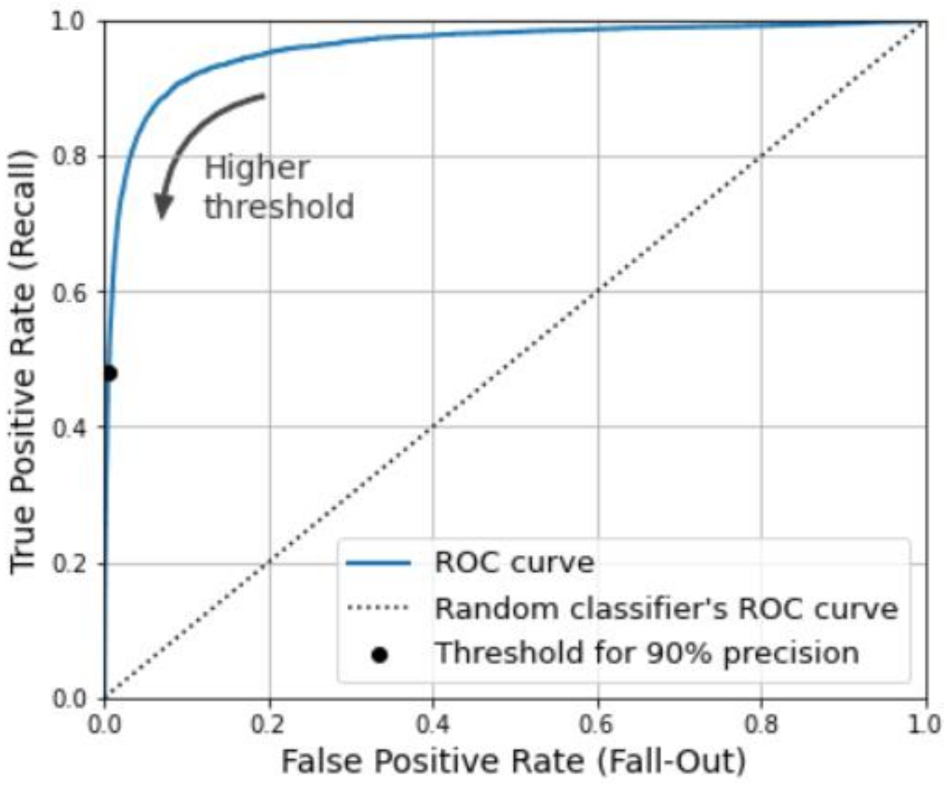


Note:

- Threshold ↑: FP↓ Threshold ↑: FN ↑
- Threshold ↑: Precision ↑
- Threshold ↑: Recall ↓

The ROC Curve

- There is a tradeoff: the higher the recall (TPR), the more false positives (FPR) the classifier produces.
- A **good classifier** stays as far **away** from that line as possible (toward the **top-left** corner).

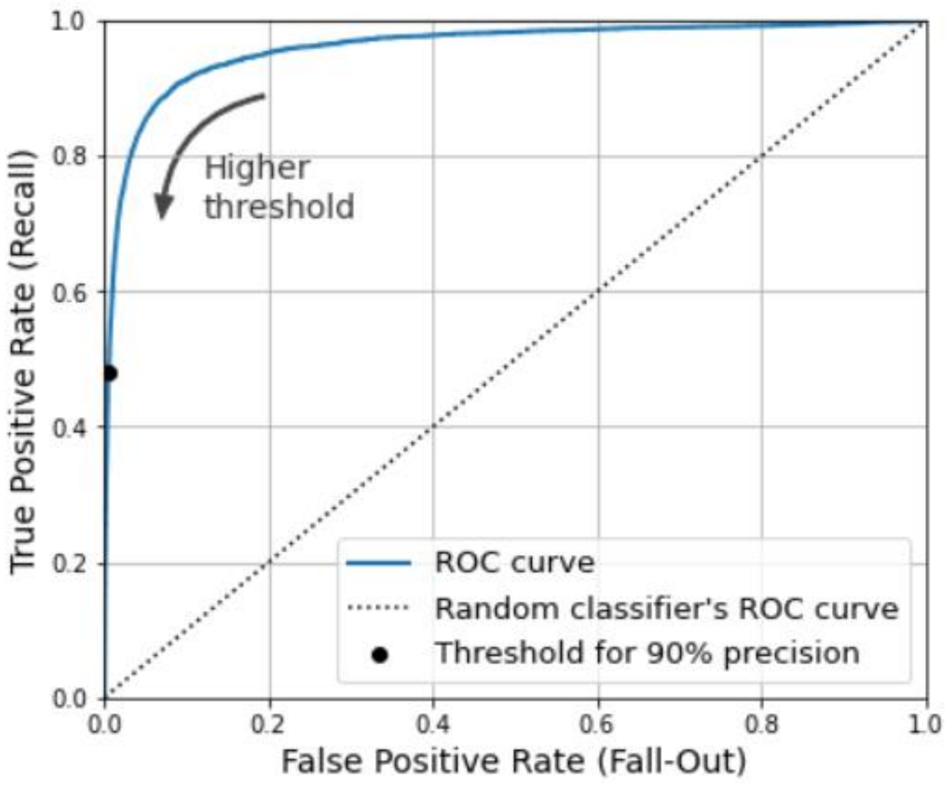


The ROC Curve

- One way to **compare** classifiers is to measure the **area under the curve** (AUC). A perfect classifier will have a ROC AUC equal to **1**, whereas a purely random classifier will have a ROC AUC equal to 0.5

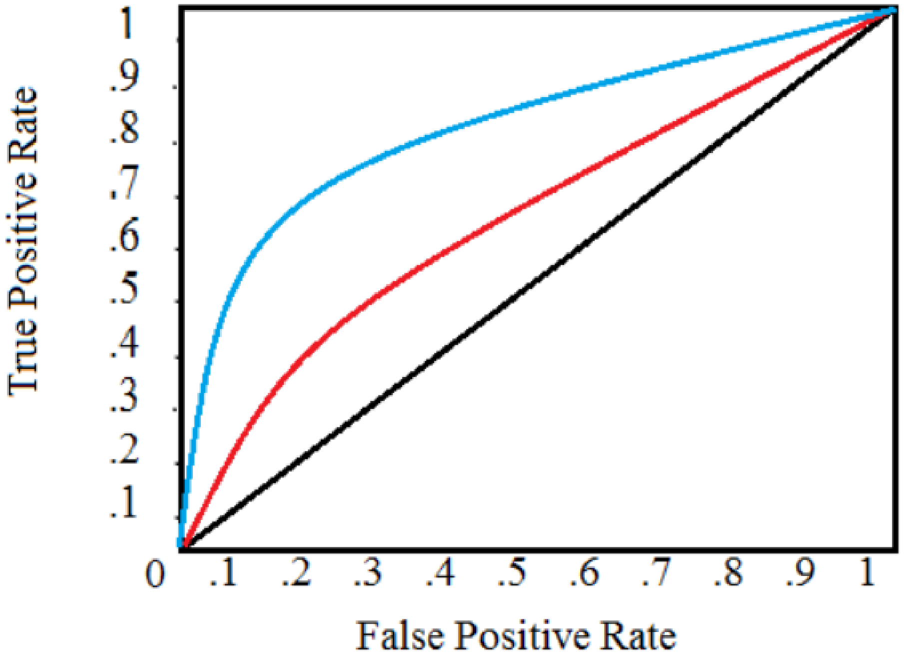
```
from sklearn.metrics import roc_auc_score

roc_auc_score(y_train_5, y_scores)
```

```
0.9604938554008616
```



The ROC Curve

- The AUC for the blue curve will be greater than that for the red curve, meaning the blue model is better at achieving a blend of precision and recall. A random classifier (the black line) achieves an AUC of 0.5

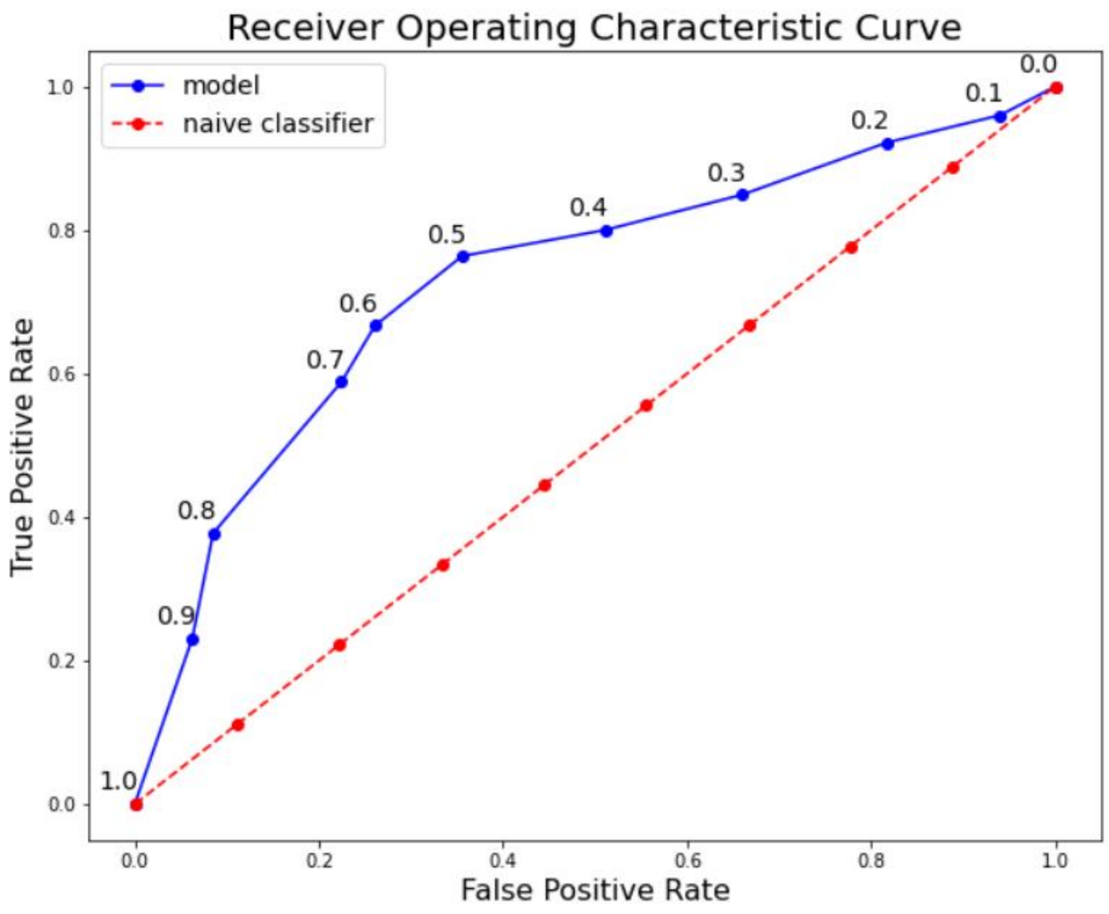


The ROC Curve

	Threshold	Recall	Precision	F1	TPR	FPR
0	0.0	1.0	0.5	0.666667	1.0	1.0
1	0.1	0.96	0.505263	0.662069	0.96	0.94
2	0.2	0.921569	0.54023	0.681159	0.921569	0.816327
3	0.3	0.849057	0.592105	0.697674	0.849057	0.659574
4	0.4	0.8	0.656716	0.721311	0.8	0.511111
5	0.5	0.763636	0.724138	0.743363	0.763636	0.355556
6	0.6	0.666667	0.75	0.705882	0.666667	0.26087
7	0.7	0.588235	0.731707	0.652174	0.588235	0.22449
8	0.8	0.377358	0.833333	0.519481	0.377358	0.085106
9	0.9	0.230769	0.8	0.358209	0.230769	0.0625
10	1.0	0.0	0	0	0	0.0

The ROC Curve

	Threshold	Recall	Precision	F1	TPR	FPR
0	0.0	1.0	0.5	0.666667	1.0	1.0
1	0.1	0.96	0.505263	0.662069	0.96	0.94
2	0.2	0.921569	0.54023	0.681159	0.921569	0.816327
3	0.3	0.849057	0.592105	0.697674	0.849057	0.659574
4	0.4	0.8	0.656716	0.721311	0.8	0.511111
5	0.5	0.763636	0.724138	0.743363	0.763636	0.355556
6	0.6	0.666667	0.75	0.705882	0.666667	0.26087
7	0.7	0.588235	0.731707	0.652174	0.588235	0.22449
8	0.8	0.377358	0.833333	0.519481	0.377358	0.085106
9	0.9	0.230769	0.8	0.358209	0.230769	0.0625
10	1.0	0.0	0	0	0	0.0



ROC Use cases

- Balanced Datasets
- Equal Costs of Misclassification: when the costs of false positives and false negatives are approximately equal
- Comparative Performance: Comparing the performance of multiple classifiers on the same test data, as they provide a clear and intuitive graphical representation of different classifiers' trade-offs between sensitivity and specificity.

Precision/Recall Curve

- Imbalanced Classes
- Focus on Positive Class
- High Cost of False Negatives

Example 01 – Balanced Dataset

	Predicted Positive	Predicted Negative
Actual Positive	40	10
Actual Negative	10	40

- **True Positives (TP)**: 40
- **False Positives (FP)**: 10
- **True Negatives (TN)**: 40
- **False Negatives (FN)**: 10

- **Precision**:
  $$\text{Precision} = \frac{TP}{TP+FP} = \frac{40}{40+10} = \frac{40}{50} = 0.8$$
- **Recall (Sensitivity or TPR)**:
  $$\text{Recall} = \frac{TP}{TP+FN} = \frac{40}{40+10} = \frac{40}{50} = 0.8$$
- **False Positive Rate (FPR)**:
  $$\text{FPR} = \frac{FP}{FP+TN} = \frac{10}{10+40} = \frac{10}{50} = 0.2$$

- Both Precision/Recall and ROC curves would provide similar insights into the model's performance.

Example 02 – Imbalanced Dataset

	Predicted Positive	Predicted Negative
Actual Positive	10	40
Actual Negative	5	945

- **True Positives (TP)**: 10
- **False Positives (FP)**: 5
- **True Negatives (TN)**: 945
- **False Negatives (FN)**: 40

- **Precision**:
  $$\text{Precision} = \frac{TP}{TP+FP} = \frac{10}{10+5} = \frac{10}{15} = 0.667$$
- **Recall (Sensitivity or TPR)**:
  $$\text{Recall} = \frac{TP}{TP+FN} = \frac{10}{10+40} = \frac{10}{50} = 0.2$$
- **False Positive Rate (FPR)**:
  $$\text{FPR} = \frac{FP}{FP+TN} = \frac{5}{5+945} = \frac{5}{950} \approx 0.0053$$

Interpretation

Precision/Recall Curve:

- Precision (0.667) and Recall (0.2) indicate that while the model has decent precision, it has low recall, meaning it misses many positive cases (actual frauds).
- This is crucial in an imbalanced dataset since it highlights the model's difficulty in identifying the minority class (frauds).

Interpretation

ROC Curve:

- TPR (0.2) and FPR (0.0053) might suggest the model performs well due to the very low FPR.
- However, in the context of imbalanced data, this can be misleading as the large number of true negatives (non-frauds) skews the interpretation.

Visualization

For balanced dataset:

- The ROC curve and Precision/Recall curve both show good performance since TPR and precision are high.

For imbalanced dataset:

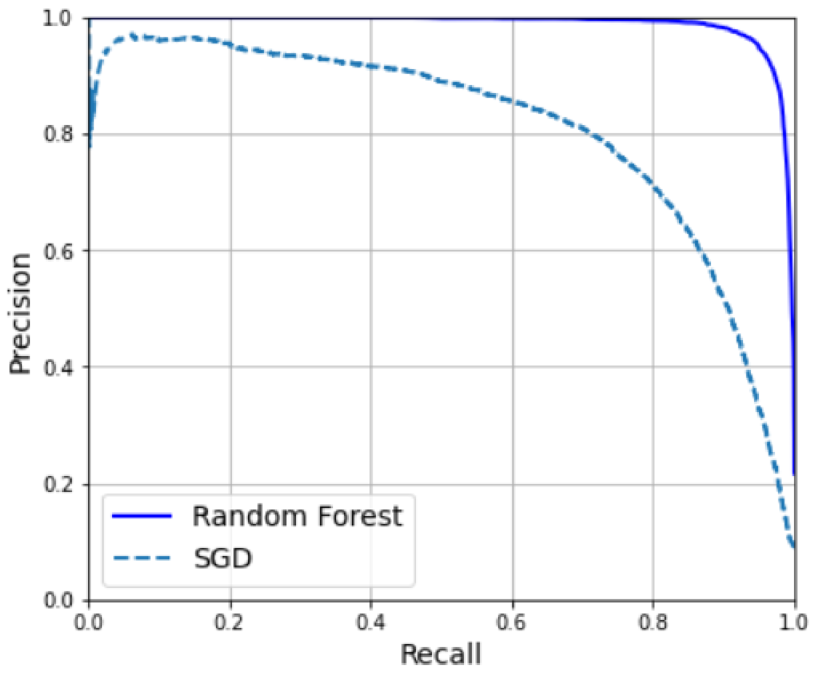
- False Positive Rate (FPR): Measures the proportion of false positives out of all actual negatives. In imbalanced datasets, the FPR can be low even if the model poorly detects the positive class.
- The ROC curve may look impressive due to low FPR, but the Precision/Recall curve reveals the low recall, indicating poor performance in identifying positives (frauds).

In conclusion

- while ROC curves can be useful in balanced datasets, the Precision/Recall curve is more informative for imbalanced datasets, especially when the positive class is the focus of the prediction task.

RandomForestClassifier & SGDClassifier

- RandomForestClassifier's PR curve looks much better than the SGDClassifier's: it comes much closer to the top-right corner.
- F_1 score and ROC AUC score are also significantly better
  - F_1 score = 0.9242275142688446
  - AUC = 0.9983436731328145
  - Precision = 99.1%
  - Recall = 86.6%



Next Lecture

- Whereas binary classifiers distinguish between two classes, multiclass classifiers (also called multinomial classifiers) can distinguish between more than two classes.
- **Multiclass Classification**