

Evaluation Metrics



Classification models

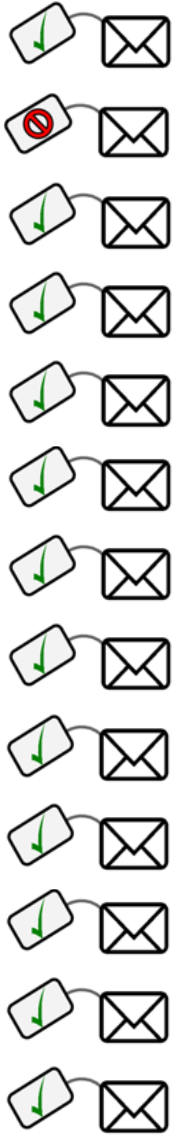
Evaluation Metrics: Accuracy

- How do we evaluate classification models?
- One possible measure: Accuracy
 - The fraction of predictions we got correct.
- In many cases, accuracy is a poor or misleading metric
 - Most often when different kinds of mistakes have different costs
 - Typical case includes class imbalance, when positives or negatives are extremely rare

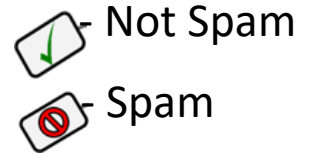
Evaluation Metrics: Accuracy

True labels

 Not Spam
 Spam



Evaluation Metrics: Accuracy




True labels





- Objective: To build a classification model that can predict if an email is a spam.


Evaluation Metrics: Accuracy


True labels






















































































































Predicted labels








































































































































































































































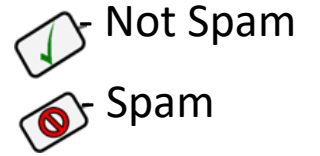
 Not Spam

 Spam

- Objective: To build a classification model that can predict if an email is a spam.
- We build a simple model that predicts every email it sees as a regular email, i.e., Not Spam.

- Objective: To build a classification model that can predict if an email is a spam.
- We build a simple model that predicts every email it sees as a regular email, i.e., Not Spam.

Evaluation Metrics: Accuracy



True labels

Predicted labels



- Objective: To build a classification model that can predict if an email is a spam.
- We build a simple model that predicts every email it sees as a regular email, i.e., Not Spam.

Total # of emails : 39

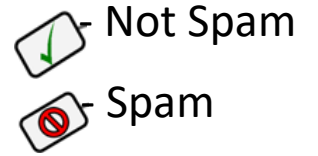
of Spam emails : 6

of Not Spam emails : 33

of incorrectly classified emails: 6

of correctly classified emails: 33

Evaluation Metrics: Accuracy



True labels

Predicted labels



- Objective: To build a classification model that can predict if an email is a spam.
- We build a simple model that predicts every email it sees as a regular email, i.e., Not Spam.

Total # of emails : 39

of Spam emails : 6

of Not Spam emails : 33

of incorrectly classified emails: 6

of correctly classified emails: 33











































































































Accuracy: fraction of correctly classified email.

= # of correctly classified emails / Total # of emails

= 33 / 39 = **0.85**

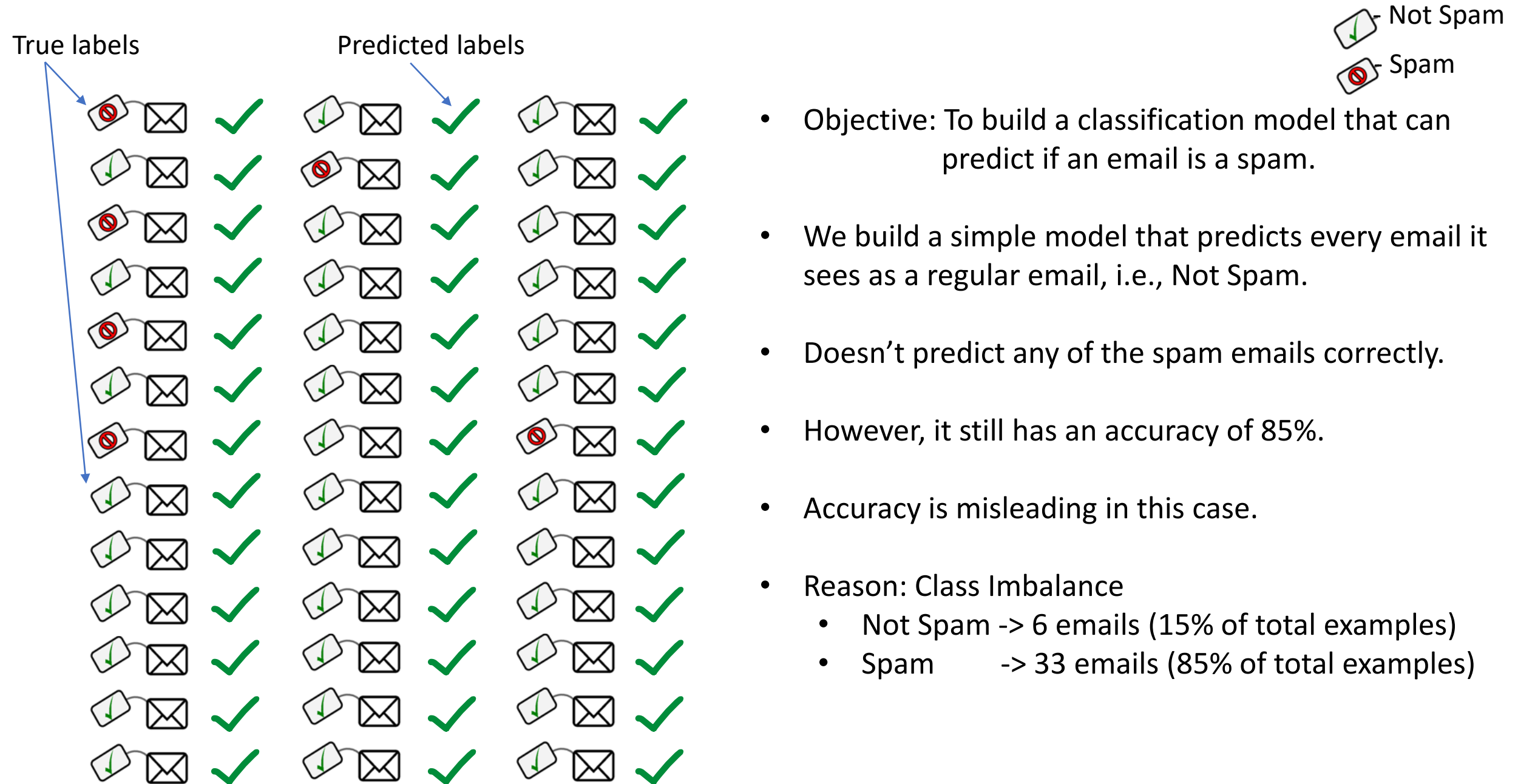
85% accuracy!!! -> Is this a good model?

Evaluation Metrics: Accuracy

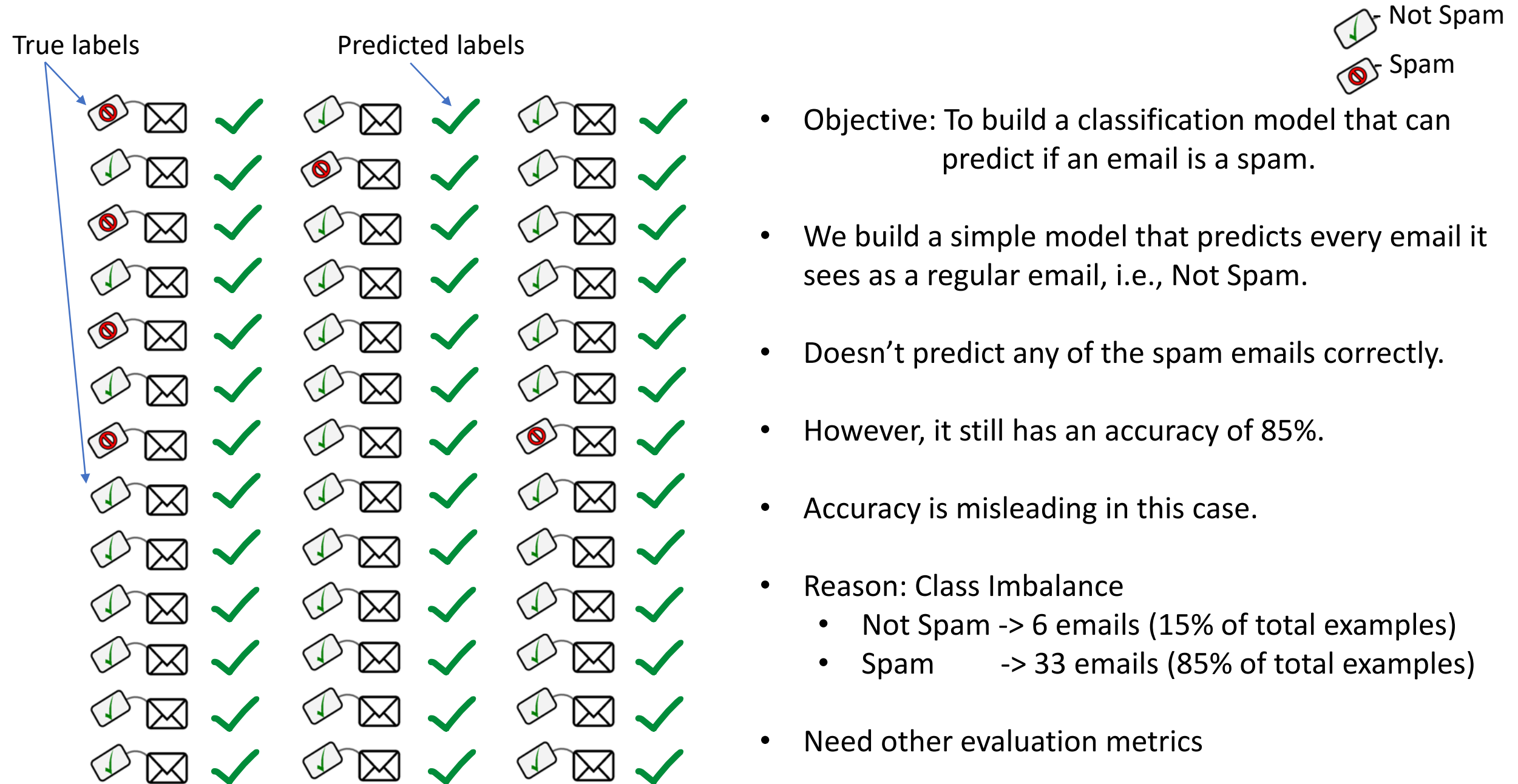
True labels			Predicted labels			 Not Spam  Spam	
							
							
							
							
							
							
							
							
							
							
							
							
							

- Objective: To build a classification model that can predict if an email is a spam.
- We build a simple model that predicts every email it sees as a regular email, i.e., Not Spam.

Evaluation Metrics: Accuracy



Evaluation Metrics: Accuracy



Classification: Threshold

- Logistic regression returns a probability.
- Returned probability can be used "as is" (e.g., the probability that the user will click on this ad is 0.00023) or can be converted to a binary value (e.g., this email is spam).
- A logistic regression model that returns 0.9995 for an email is predicting that it is very likely to be spam. Conversely, another email with a prediction score of 0.0003 on that same logistic regression model is very likely not spam.
- However, what about an email with a prediction score of 0.6?
- In order to map a logistic regression value to a binary category, you must define a **classification threshold** (also called the **decision threshold**).
- A value above that threshold indicates "spam"; a value below indicates "not spam."
- It is tempting to assume that the classification threshold should always be 0.5, but thresholds are problem-dependent, and are therefore should be optimized.

Classification: Threshold

- Part of choosing a threshold is assessing how much you'll suffer for making a mistake, or what is your tolerance level for mistakes.

Low Threshold better

- Situations where more false positives have relatively lower negative consequences than that from more false negatives.
- Better for cases when we would like to take some preventive measures/actions.
- Examples:
 - Fraud Detection
 - Disease Detection
 - Customer Churn
 - Spam Filtering
 - Predictive Maintenance


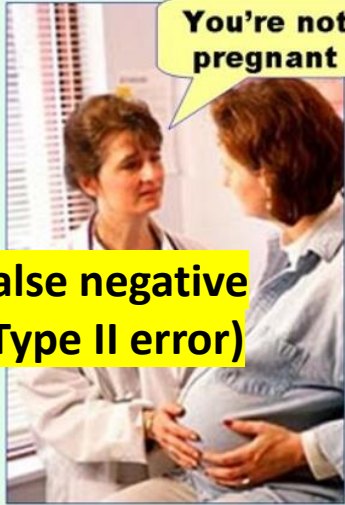


High Threshold better

- Situations where more false negatives have relatively lower negative consequences than that from more false positives.
- Better for cases when we would like to take some targeted measures/actions.
- Examples:
 - Credit Risk Assessment
 - Medical Tests
 - Job Applications
 - Legal Decisions
 - Product Defect Detection

Types of errors

Predicted values


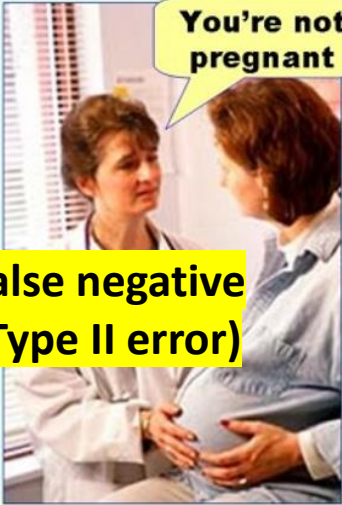


Actual Values

	1 (Pregnant)	0 (Not Pregnant)
1 (Pregnant)	 <p>True positive</p>	 <p>False negative (Type II error)</p>
0 (Not Pregnant)	 <p>False positive (Type I error)</p>	 <p>True negative</p>

Types of errors

Actual Values

Predicted values

	1 (Pregnant)	0 (Not Pregnant)
1 (Pregnant)	 True positive	 False negative (Type II error)
0 (Not Pregnant)	 False positive (Type I error)	 True negative

Predicted values

Actual Values

	1	0
1	True positive Actual: 1 (Pregnant) Predicted: 1 (Pregnant)	False negative (Type II error) Actual: 1 (Pregnant) Predicted: 0 (Not Pregnant)
0	False positive (Type I error) Actual: 0 (Not Pregnant) Predicted: 1 (Pregnant)	True negative Actual: 0 (Not Pregnant) Predicted: 0 (Not Pregnant)

Confusion Matrix

- A table that describes the performance of a classification model, i.e., gives the aggregate of total true positives, false positives, false negatives, and true negatives
- E.g.
 - We have 165 patients; 105 patients have COVID (1), 60 patients don't (0).
 - We build a classification model to predict whether the patients have COVID or not.
 - Out of 105 patients that actually have COVID, our model was able to correctly predict 100 patients as having COVID, while incorrectly predict 5 as not having COVID.
 - Out of 60 patients that did not have COVID, our model was able to correctly predict 50 as not having COVID, while incorrectly predict 10 as having COVID.

		Predicted values	
		1	0
Actual Values	1	100 TP	5 FN
	0	10 FP	50 TN

Confusion Matrix

- A table that describes the performance of a classification model, i.e., gives the aggregate of total true positives, false positives, false negatives, and true negatives
- E.g.
 - We have 165 patients; 105 patients have COVID (1), 60 patients don't (0).
 - We build a classification model to predict whether the patients have COVID or not.
 - Out of 105 patients that actually have COVID, our model was able to correctly predict 100 patients as having COVID, while incorrectly predict 5 as not having COVID.
 - Out of 60 patients that did not have COVID, our model was able to correctly predict 50 as not having COVID, while incorrectly predict 10 as having COVID.

		Predicted values	
		1	0
Actual Values	1	100 TP	5 FN
	0	10 FP	50 TN

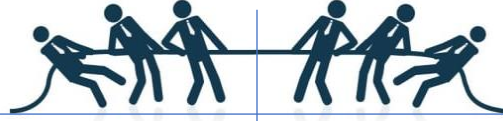
- **Accuracy:** How often is the model **correct**?

$$\frac{TP + TN}{TP + TN + FP + FN} = \frac{150}{165} = 0.91$$

- **Misclassification Rate:** How often is the model **wrong**?

$$\frac{FP + FN}{TP + TN + FP + FN} = \frac{15}{165} = 0.09$$

Evaluation Metrics



Recall

- Out of all the actual positives, how many did the model predict correctly.

Actual Values	Predicted values	
	1	0
1	100 TP	5 FN
0	10 FP	50 TN

All Actual positives

$$\text{Recall} = \frac{\text{True Positives}}{\text{All Actual Positives}}$$
$$= \frac{TP}{TP + FN} = \frac{100}{105} = 0.95$$

- Proportion of relevant things that are retrieved.
- A model with high recall might give a lot of irrelevant things, but it also returns most of the important items.
- Higher recall can be achieved by lowering the prediction threshold, as it reduces false negatives at the cost of increasing false positives.
- Also known as: **Sensitivity, True Positive Rate**

Precision

- Of all the times when the model predicted positive, how many times was it actually correct.

Actual Values	Predicted values	
	1	0
1	100 TP	5 FN
0	10 FP	50 TN

All Predicted positives

$$\text{Precision} = \frac{\text{True Positives}}{\text{All Predicted Positives}}$$
$$= \frac{TP}{TP + FP} = \frac{100}{110} = 0.91$$

- Proportion of retrieved things that are relevant.
- A model with high precision might leave some good items out, but what it returns are of high quality.
- Higher precision can be achieved by raising the prediction threshold, as it reduces false positives at the cost of increasing false negatives.

Evaluation Metrics

Recall



Precision

Supply Chain Problem: Prevent Truck Driver Accidents

Predicting truck driver accidents, we may want a **high recall** (and be ok with low precision). That is, we want a list that captures all the high risk drivers.

We can then do extra training and extra monitoring.

And, we are ok that this list may also include a lot of drivers who wouldn't have had accidents anyway.

Our money spent on training and monitoring these already good drivers is worth it if we prevent just one severe accident.

Supply Chain Problem: Predict Stock Outs

If 200 of 5,000 items will stock out next month, we may want **high precision**. That is, we would be happy if we listed with high precision 60 SKUs that most likely are going to stock out.

We expedite and take extra measures with these 60 SKUs. We will still miss 140. But, that is better than the model giving us a list of 600 SKUs.

The list of 600 has most of the 200 in there, but we will be spending time and money on 400 items where there isn't going to be a problem.

Evaluation Metrics

- **Accuracy:** Overall, how often is the model correct?

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

- **Misclassification Rate (Error Rate):** Overall, how often is the model wrong?

$$\text{Misclassification Rate} = \frac{FP + FN}{TP + TN + FP + FN}$$

- **Recall (Sensitivity) (True Positive Rate):** When actual value is positive, how often is prediction correct?

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

- **F1 Score:** When precision and recall are equally valued.

- **Precision:** Of the positive predictions, how many are actually correct?

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

$$\text{F1 Score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

- **Specificity:** When actual value is negative, how often is prediction correct?

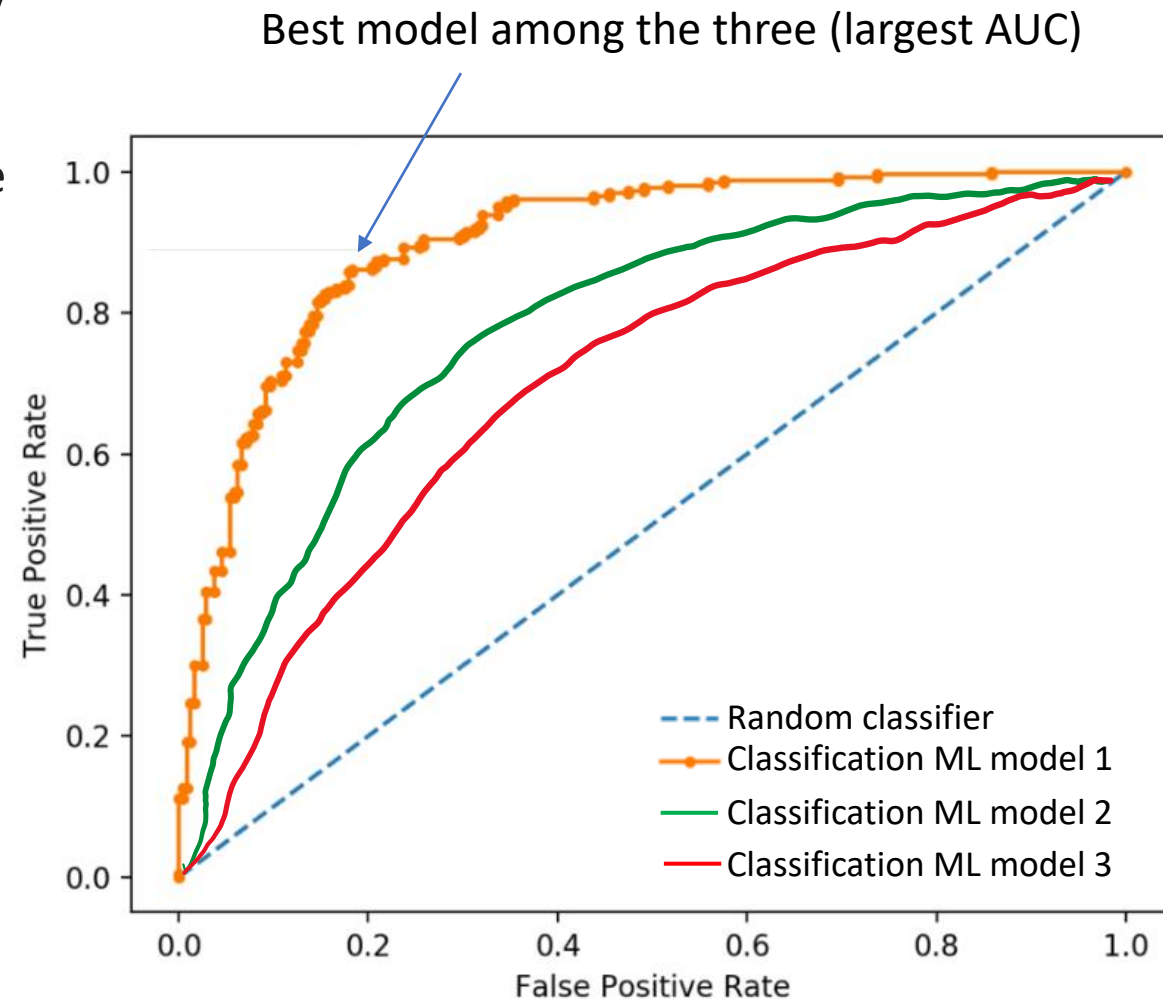
$$\text{Specificity} = \frac{TN}{TN + FP}$$

- **False Positive Rate:** When actual value is negative, how often is prediction wrong?

$$\text{False Positive Rate} = 1 - \text{Specificity} = \frac{FP}{TN + FP}$$

ROC Curve and Area under the ROC Curve (AUC)

- Graphical representation of the performance of a binary classification.
- Plots True Positive Rate (TPR) against False Positive Rate (FPR) for different classification thresholds.
- Shows how well the model can distinguish between positive and negative cases as the threshold for classification is varied.
- A perfect classifier would have ROC curve that passes through the top left corner of the plot, indicating 100% TPR and 0% FPR
- Area under the ROC curve (AUC) – measure of the area underneath the ROC curve.
- A higher AUC indicate a better classifier. A random classifier would have an AUC of 0.5, indicating no better performance than random chance.



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

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

Precision-Recall Curve

- Another graphical representation of the performance of a binary classification.
- Plots Precision against Recall for different classification thresholds.
- Precision-Recall curve is more informative than the ROC curve when evaluating binary classifiers on imbalanced dataset.
- The main reason for this is because of the use of true negatives in the FPR in the ROC curve and the careful avoidance of this rate in the Precision-Recall curve.
- Precision and Recall can be plotted as a function of classification thresholds to find the optimal threshold that balances both precision and recall. Useful in cases where both precision and recall are equally important.

