

Preliminaries - ML



Some Terminology....

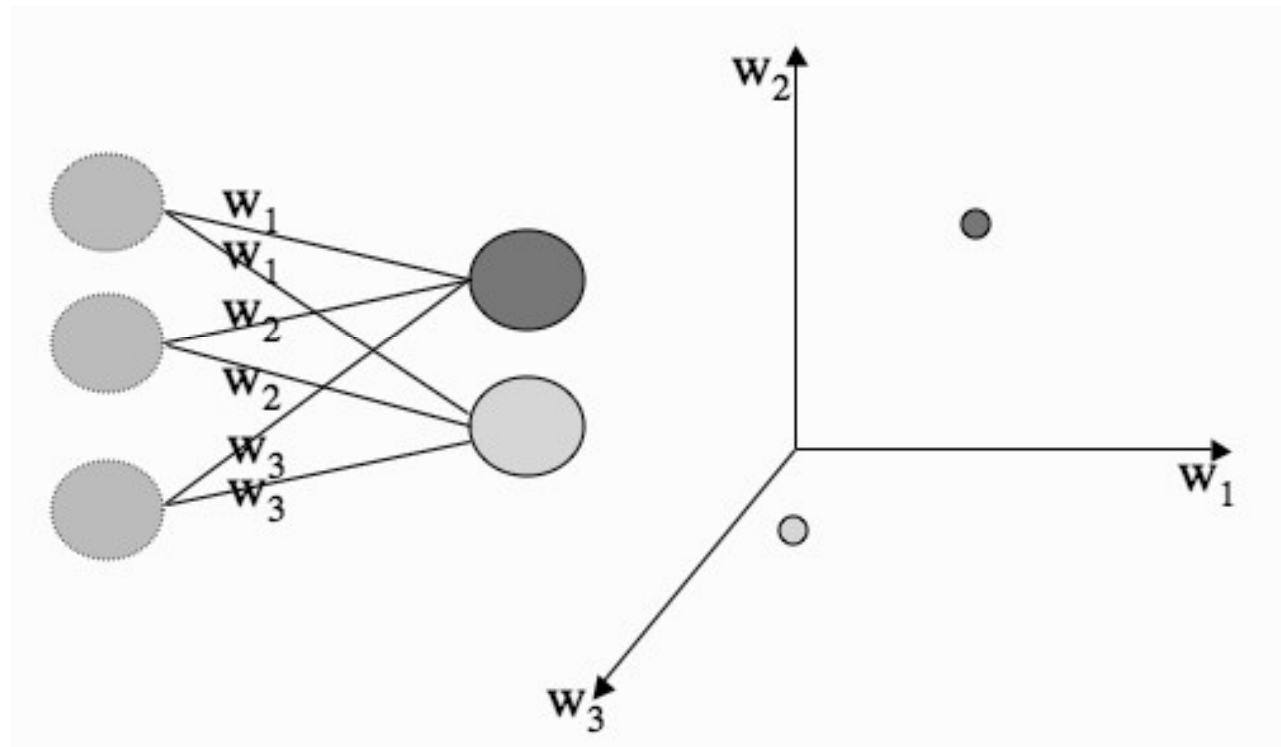
- Inputs - An input vector is the data given as one input to the algorithm. Written as x , with elements x_i , where i runs from 1 to the number of input dimensions, m .
- Weights - w_{ij} - weighted connections between nodes i and j . For neural networks these weights are analogous to the synapses in the brain. They are arranged into a matrix W .
- Outputs - The output vector is y , with elements y_j , where j runs from 1 to the number of output dimensions, n . We can write $y(x, W)$ to remind ourselves that the output depends on the inputs to the algorithm and the current set of weights of the network.

Some Terminology....

- **Targets** - The target vector t , with elements t_j , where j runs from 1 to the number of output dimensions, n , are the extra data that we need for supervised learning, since they provide the 'correct' answers that the algorithm is learning about.
- **Activation Function** - For neural networks, $g(\cdot)$ is a mathematical function that describes the firing of the neuron as a response to the weighted inputs, such as the threshold function.
- **Error E** - a function that computes the inaccuracies of the network as a function of the outputs y and targets t .

Weight space

- Representation 2D or 3D



Weight space

- Distances between inputs and neurons are computed by
 - Euclidean distance - in two dimensions can be written as:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.$$

- If the neuron is close to the input in this sense then it should fire, and if it is not close then it shouldn't fire.

Curse of Dimensionality

- The curse of dimensionality refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces that do not occur in low-dimensional space. The expression was coined by Richard E. Bellman when considering problems in dynamic programming.
- Dimensionally cursed phenomena occur in domains such as numerical analysis, sampling, combinatorics, machine learning, data mining and databases.
- The essence of the curse is the realization that as the number of dimensions increases, the volume of the unit hypersphere does not increase with it.
- The popular aspects of curse of dimensionality;
 - ‘data sparsity’
 - ‘distance concentration’

Data Sparsity

- When number of features/dimension increases, the samples are not increasing as per the requirement
- An effective way to build a generalized model is to capture different possible combinations of the values of predictor variables and the corresponding targets.

Distance Concentration

- The unit hypersphere is the region we get if we start at the origin (the centre of our coordinate system) and draw all the points that are distance 1 away from the origin.
- 2D – circle (0, 0)
- 3D - sphere around (0, 0, 0)
- In higher dimensions, the sphere becomes a hypersphere.
- number of input dimensions gets larger, need more data to enable the algorithm for generalisation
- understand something about the data in advance

Testing ML algorithms – Overfitting

- train for too long, then overfitting occurs
- learnt about the noise and inaccuracies in the data as well as the actual function.
- the learned model will be much complicated, and won't generalise in nature
- Cross validation
- Training, testing and validation

Overfitting - Generalization vs memorization

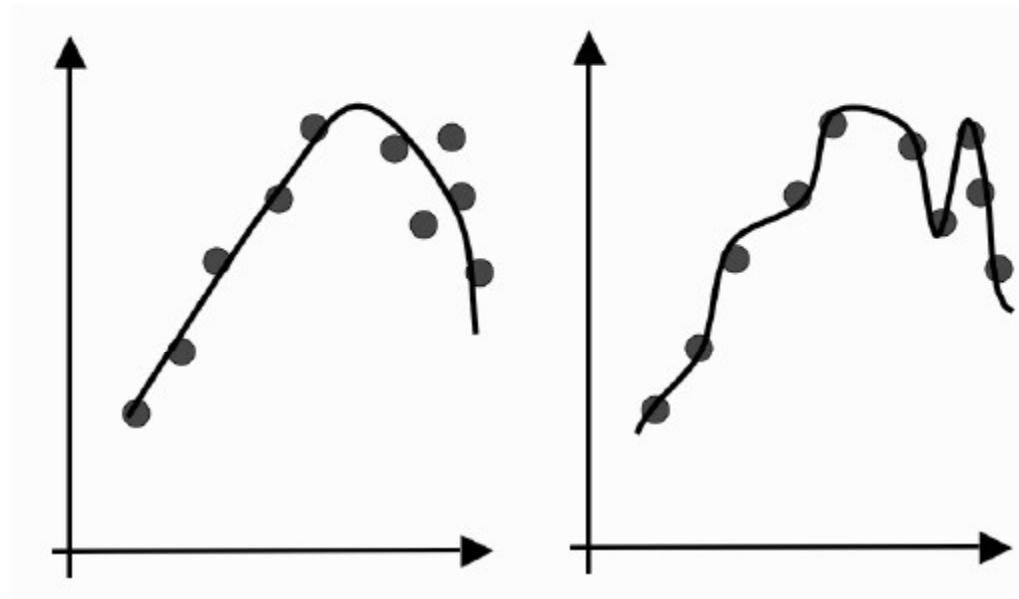
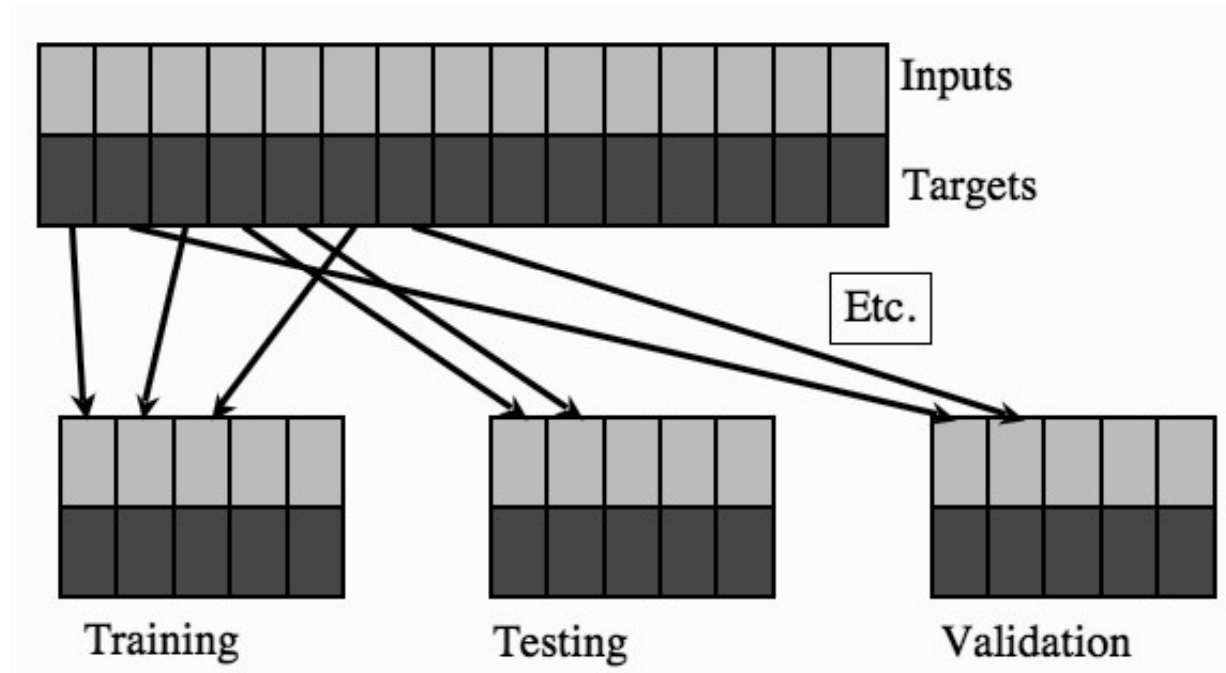


FIGURE 2.5 The effect of overfitting is that rather than finding the generating function (as shown on the left), the neural network matches the inputs perfectly, including the noise in them (on the right). This reduces the generalisation capabilities of the network.

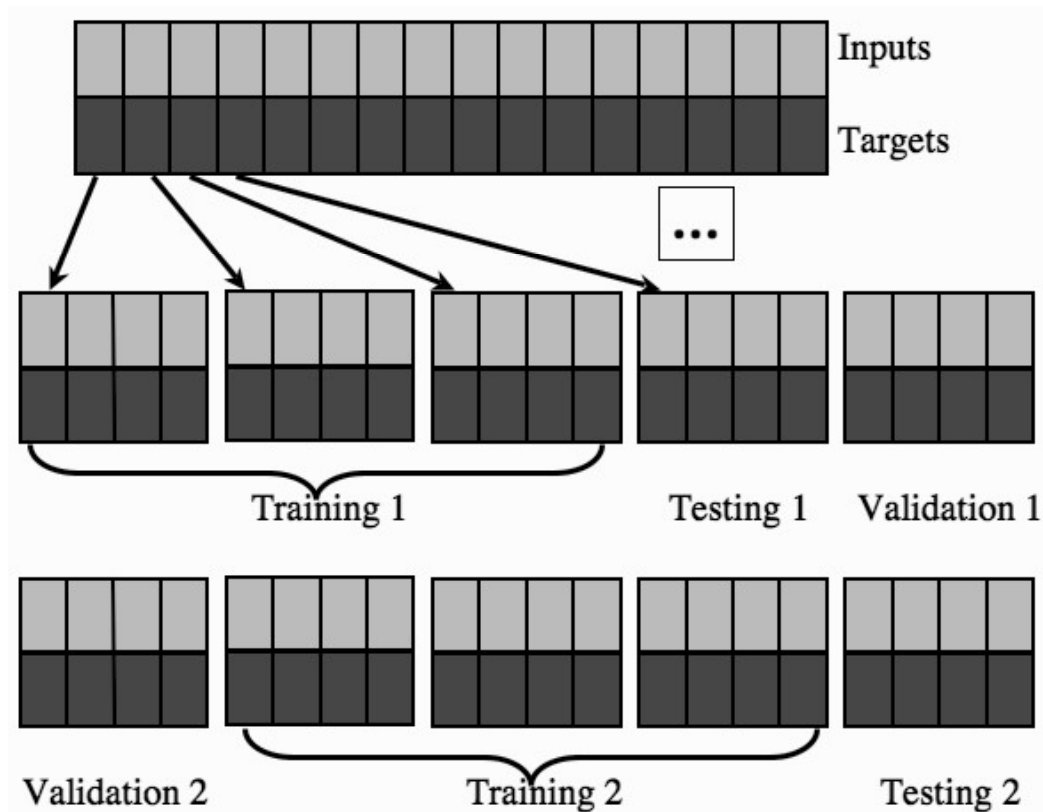
Training, Testing and Validation sets

- 50:25:25 or 60:20:20
- Randomly assign samples



Training, Testing and Validation sets

- short of training data - leave-some-out, multi-fold cross-validation



Classification – Performance metrics

- Accuracy
- Sensitivity
- Specificity
- Precision
- Recall
- F1 measure
- True positive rate (TPR)
- False negative rate (FNR)
- ROC
- AUC

Confusion Matrix

- Square matrix that contains all the possible classes in both the horizontal and vertical directions and list the classes along the top of a table as the predicted outputs, and then down the left-hand side as the targets.
- Example :

Outputs			
	C_1	C_2	C_3
C_1	5	1	0
C_2	1	4	1
C_3	2	0	4

- Accuracy?

Confusion matrix – Binary class

- Confusion matrix of binary class

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error
	Negative	False Positive (FP) Type I Error	True Negative (TN)

- Accuracy ???

Confusion matrix – Binary class with metrics

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

Sensitivity and Specificity

- Sensitivity (also known as the true positive rate) is the ratio of the number of correct positive examples to the number classified as positive
- Specificity is the same ratio for negative examples.

$$\text{Sensitivity} = \frac{\#TP}{\#TP + \#FN}$$

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

Precision and Recall

- Precision is the ratio of correct positive examples to the number of actual positive examples
- **Recall** is the ratio of the number of correct positive examples out of those that were classified as positive, which is the same as **sensitivity**

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

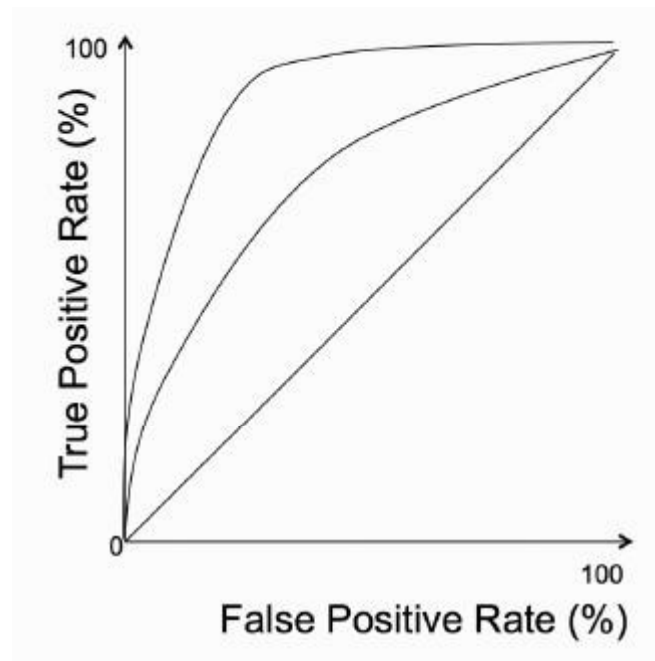
$$\text{Recall} = \frac{\#TP}{\#TP + \#FN}$$

F1 measure

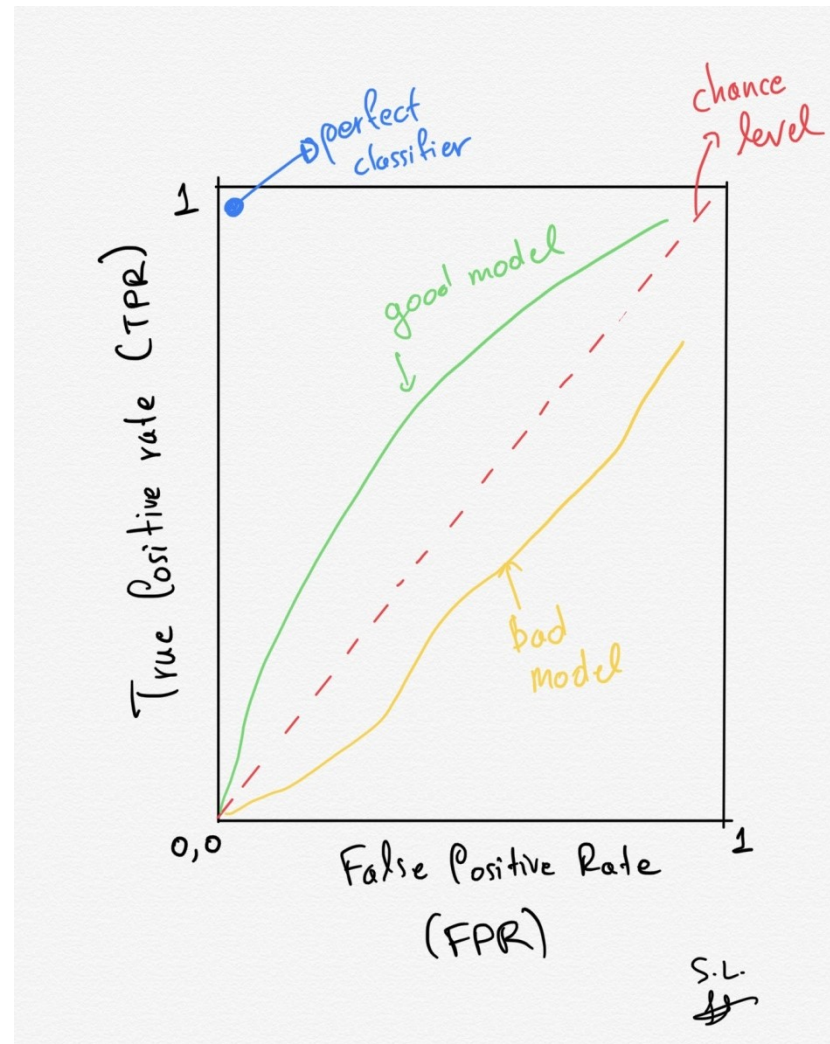
- Precision and recall are to some extent inversely related, in that if the number of false positives increases, then the number of false negatives often decreases, and vice versa.
- They can be combined to give a single measure, the *F1 measure*, which can be written in terms of precision and recall as:

$$F_1 = 2 \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Receiver Operator Characteristic (ROC) Curve



Receiver Operator Characteristic (ROC) Curve



ROC - AUC

- TPR - Sensitivity
- FPR - (1-specificity)
- With imbalanced datasets, the Area Under the Curve (AUC) score is calculated from ROC and is a very useful metric in imbalanced datasets.
- The AUC (area under the curve) indicates if the curve is above or below the diagonal (chance level). AUC ranges in value from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0 and one whose predictions are 100% correct has an AUC of 1.0.

Performance metrics – overall

		True condition			
Total population		Condition positive	Condition negative	$Prevalence = \frac{\Sigma \text{Condition positive}}{\Sigma \text{Total population}}$	$Accuracy (ACC) = \frac{\Sigma \text{True positive} + \Sigma \text{True negative}}{\Sigma \text{Total population}}$
Predicted condition	Predicted condition positive	True positive	False positive, Type I error	$Positive \text{ predictive value (PPV), Precision} = \frac{\Sigma \text{True positive}}{\Sigma \text{Predicted condition positive}}$	$False \text{ discovery rate (FDR)} = \frac{\Sigma \text{False positive}}{\Sigma \text{Predicted condition positive}}$
	Predicted condition negative	False negative, Type II error	True negative	$False \text{ omission rate (FOR)} = \frac{\Sigma \text{False negative}}{\Sigma \text{Predicted condition negative}}$	$Negative \text{ predictive value (NPV)} = \frac{\Sigma \text{True negative}}{\Sigma \text{Predicted condition negative}}$
		$True \text{ positive rate (TPR), Recall, Sensitivity, probability of detection, Power} = \frac{\Sigma \text{True positive}}{\Sigma \text{Condition positive}}$	$False \text{ positive rate (FPR), Fall-out, probability of false alarm} = \frac{\Sigma \text{False positive}}{\Sigma \text{Condition negative}}$	$Positive \text{ likelihood ratio (LR+)} = \frac{TPR}{FPR}$	$Diagnostic \text{ odds ratio (DOR)} = \frac{LR+}{LR-}$
	$False \text{ negative rate (FNR), Miss rate} = \frac{\Sigma \text{False negative}}{\Sigma \text{Condition positive}}$	$Specificity (SPC), Selectivity, True negative rate (TNR)} = \frac{\Sigma \text{True negative}}{\Sigma \text{Condition negative}}$	$Negative \text{ likelihood ratio (LR-)} = \frac{FNR}{TNR}$	$F_1 \text{ score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$	













https://en.wikipedia.org/wiki/Sensitivity_and_specificity

Accuracy Metrics – overall - Example

		Patients with bowel cancer (as confirmed on endoscopy)			
		Condition positive	Condition negative	Prevalence = (TP + FN) / Total_Population = (20 + 10) / 2030 ≈ 1.48%	Accuracy (ACC) = (TP + TN) / Total_Population = (20 + 1820) / 2030 ≈ 90.64%
Fecal occult blood screen test outcome	Test outcome positive	True positive (TP) = 20 (2030 × 1.48% × 67%)	False positive (FP) = 180 (2030 × (100 – 1.48%) × (100 – 91%))	Positive predictive value (PPV), Precision = TP / (TP + FP) = 20 / (20 + 180) = 10%	False discovery rate (FDR) = FP / (TP + FP) = 180 / (20 + 180) = 90.0%
	Test outcome negative	False negative (FN) = 10 (2030 × 1.48% × (100 – 67%))	True negative (TN) = 1820 (2030 × (100 – 1.48%) × 91%)	False omission rate (FOR) = FN / (FN + TN) = 10 / (10 + 1820) ≈ 0.55%	Negative predictive value (NPV) = TN / (FN + TN) = 1820 / (10 + 1820) ≈ 99.45%
		TPR, Recall, Sensitivity = TP / (TP + FN) = 20 / (20 + 10) ≈ 66.7%	False positive rate (FPR), Fall-out, probability of false alarm = FP / (FP + TN) = 180 / (180 + 1820) = 9.0%	Positive likelihood ratio (LR+) = $\frac{TPR}{FPR}$ = (20 / 30) / (180 / 2000) ≈ 7.41	Diagnostic odds ratio (DOR) = $\frac{LR+}{LR-}$ ≈ 20.2
		False negative rate (FNR), Miss rate = FN / (TP + FN) = 10 / (20 + 10) ≈ 33.3%	Specificity, Selectivity, True negative rate (TNR) = TN / (FP + TN) = 1820 / (180 + 1820) = 91%	Negative likelihood ratio (LR-) = $\frac{FNR}{TNR}$ = (10 / 30) / (1820 / 2000) ≈ 0.366	
					F ₁ score = 2 × $\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$ ≈ 0.174

https://en.wikipedia.org/wiki/Sensitivity_and_specificity

Check your understanding...

	Predicted class POSITIVE (spam )	Predicted class NEGATIVE (normal )
Actual class POSITIVE (spam )	TRUE POSITIVE (TP)   320	FALSE NEGATIVE (FN)   43
Actual class NEGATIVE (normal )	FALSE POSITIVE (FP)   20	TRUE NEGATIVE (TN)   538

Multiclass Classification – metrics

	Predicted class POSITIVE (spam 📧)	Predicted class NEGATIVE (ad 📧)	Predicted class NEGATIVE (normal 📧)
Actual class POSITIVE (spam 📧)	TRUE POSITIVES 📧 📧 27	FALSE NEGATIVES 📧 📧 286 📧 📧 40	
Actual class NEGATIVE (ad 📧)	📧 📧 1 FALSE POSITIVES	📧 📧 37 TRUE NEGATIVES	📧 📧 9 TRUE NEGATIVES
Actual class NEGATIVE (normal 📧)	📧 📧 5 FALSE POSITIVES	📧 📧 16 TRUE NEGATIVES	📧 📧 500 TRUE NEGATIVES

Multiclass Classification – metrics

- **True Positives**, i.e. where the actual and predicted class is spam
- **False Negatives**, where the actual class is spam, and the predicted class is normal or ad
- **False Positives**, where the actual class is normal or ad, and the predicted class is spam
- **True Negatives**, where the actual class is ad or normal, and the predicted class is ad or normal. An incorrect prediction inside the negative class is still considered as a true negative

Source: <https://towardsdatascience.com/confusion-matrix-and-class-statistics-68b79f4f510b>

Check your understanding...

Find the value of following performance metrics

- What is contingency table?
- What is error matrix?
- Accuracy
- Sensitivity - True positive rate (TPR) - Recall
- Specificity
- Precision
- F1 measure
- False negative rate (FNR) – Miss rate

