AI Implications for Business Strategy

3. Machine Learning - Classification

Classification

- The model predicts which of n discrete categories a give input should be classified into
  - We will stick with binary classifications for most of this course
  - For example:
    - Loan risk for a bank customer is high or low
    - Medical test results indicate cancer or no cancer
- When the labeled data is linearly separable (ie the two categories can be separated into two disjoint categories with no errors)
  - Then the problem can be solved with a perceptron
  - We will deal with perceptrons in the Deep Learning section
- When there uncertainty about the classification, it is a *stochastic* classification

Separability

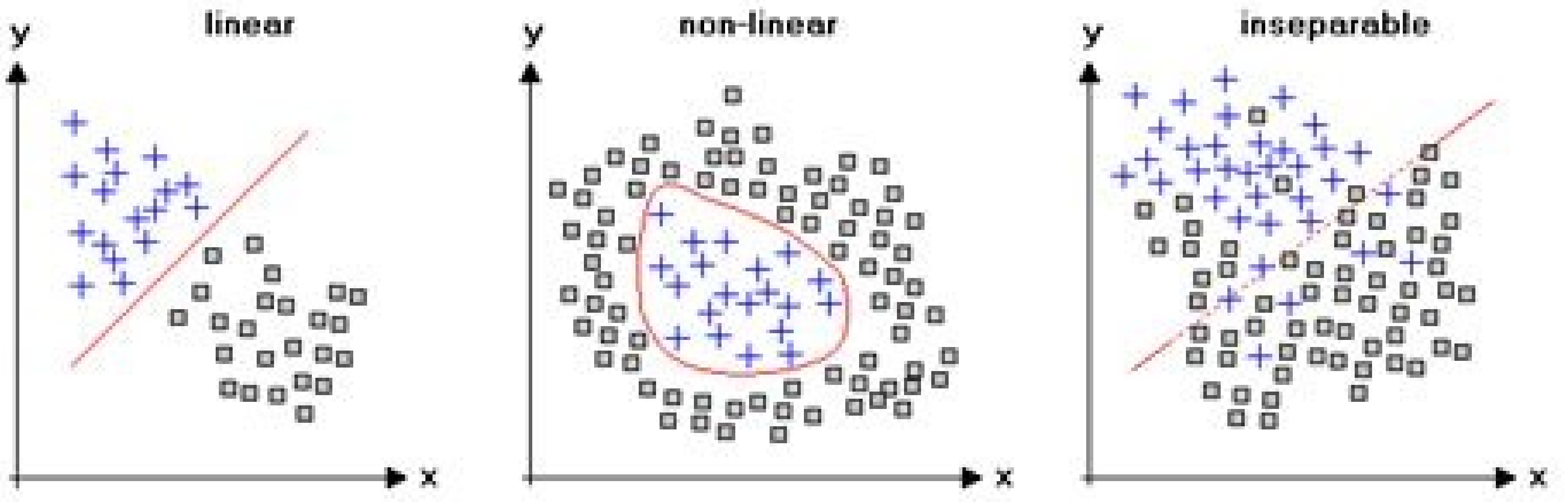
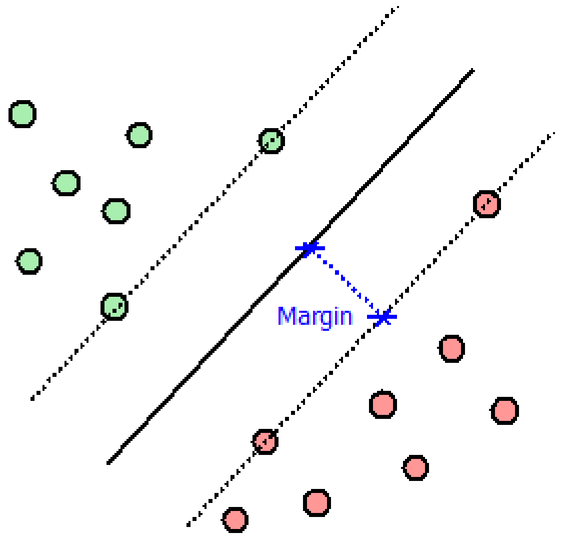


image Credit: http://www.statistics4u.com/fundstat_eng/cc_data_structure.html

The Margin

- We are not interested in the non-linear problem yet
- Margins
  - The margin is the minimum distance from the data points to the separation boundary
  - The margin is negative if there are miss-classified points
  - We use stochastic classification to deal with the cases close to the margin
- We want to assign labels of 1 or 0 to the correctly classified points
  - And a probability of how likely a point is to be correctly classified when it is close to the margin



Fuzzy and Crisp Sets

- In crisp sets, membership is binary – classical set theory
  - There is as clear boundary between what is in the set and what is outside
- For fuzzy sets, membership is a gradient
  - Usually determined by how much an data point resembles or is like a prototype
  - This prototype is often called an exemplar or best example of that class
  - An element may have partial membership is several different fuzzy sets
- For Stochastic regression
  - We try to compute a likely hood of a data point belonging to a fuzzy set
  - Done by comparing by developing a probabilistic classification based on features

Color Terms

- There are best examples of the primary colors
- But different languages assign variants to different categories
  - Depends on the amount a similarity to the prototypes
  - The classification rules are language specific
  - The features could be primary colors, hue, saturation, brightness etc.

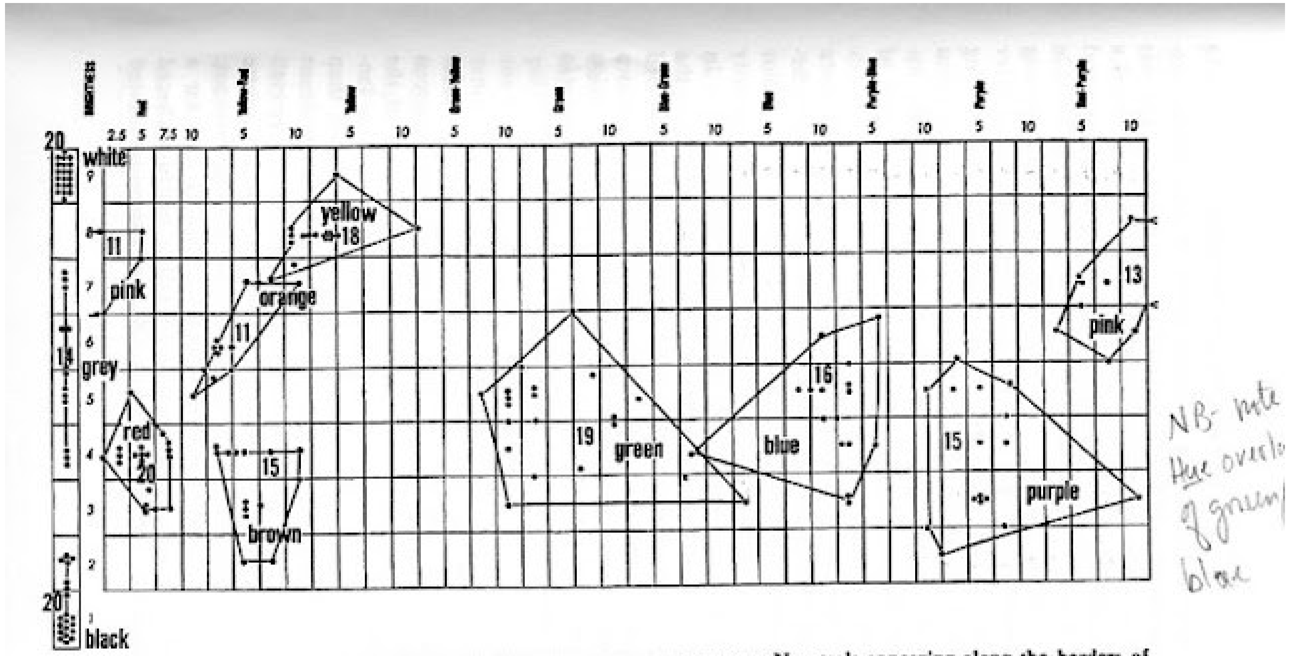
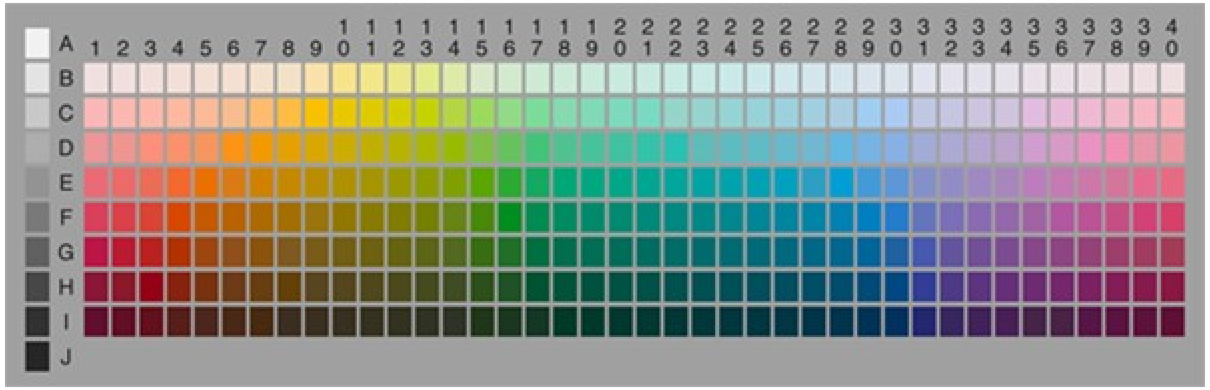


FIGURE 5. NORMALIZED FOCI OF BASIC COLOR TERMS IN TWENTY LANGUAGES. NOTE: Numerals appearing along the borders of the chart refer to the Munsell system of color notation. Numerals appearing on the body of the chart refer to the number of languages in the sample of twenty which encode the corresponding color category. The smallest possible number of lines are used to enclose each color area.



The Sigmoid Function

- Sigmoid function maps the real number line to the interval [0,1]

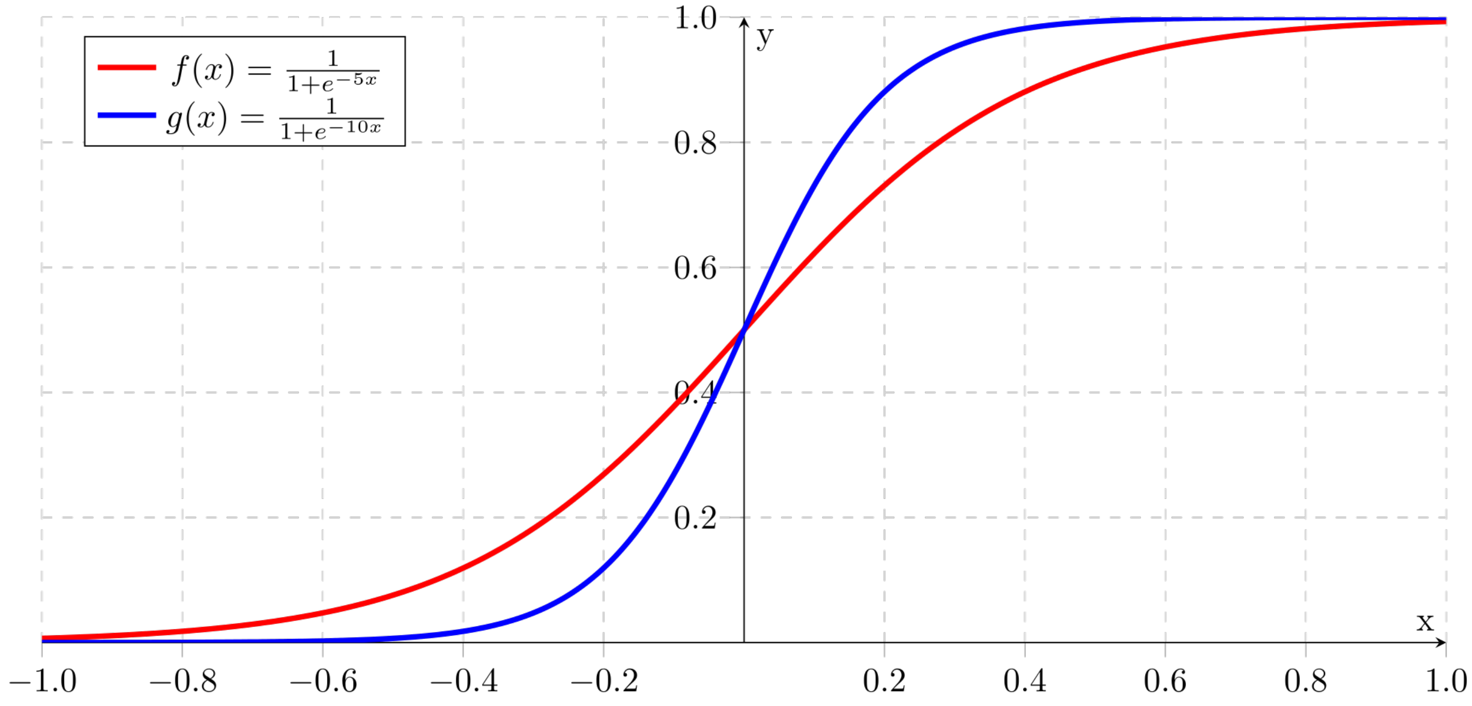
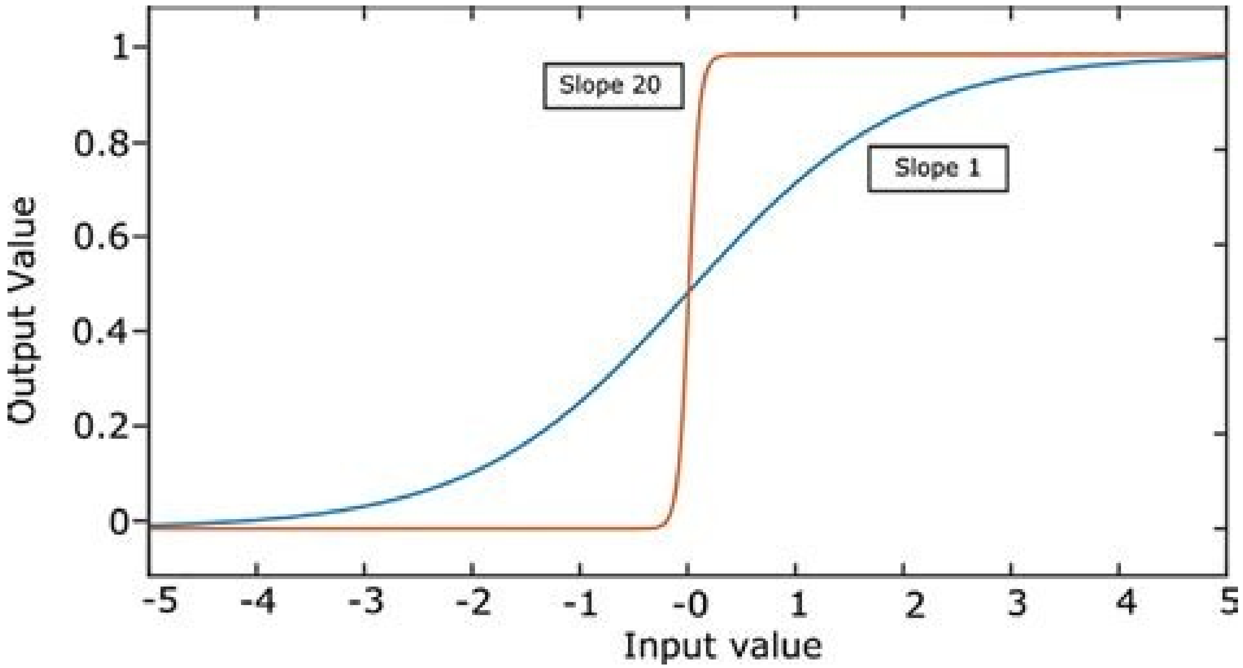


Image Credit: https://math.stackexchange.com/questions/2437708/sigmoid-circ-sigmoid-sigmoid

Stochastic Regression

- Finding the right slope of the sigmoid function to describe the probability that data is classified correctly



Sigmoid Alternative

- The hyperbolic tan function is often used to map into the range [-1,1]

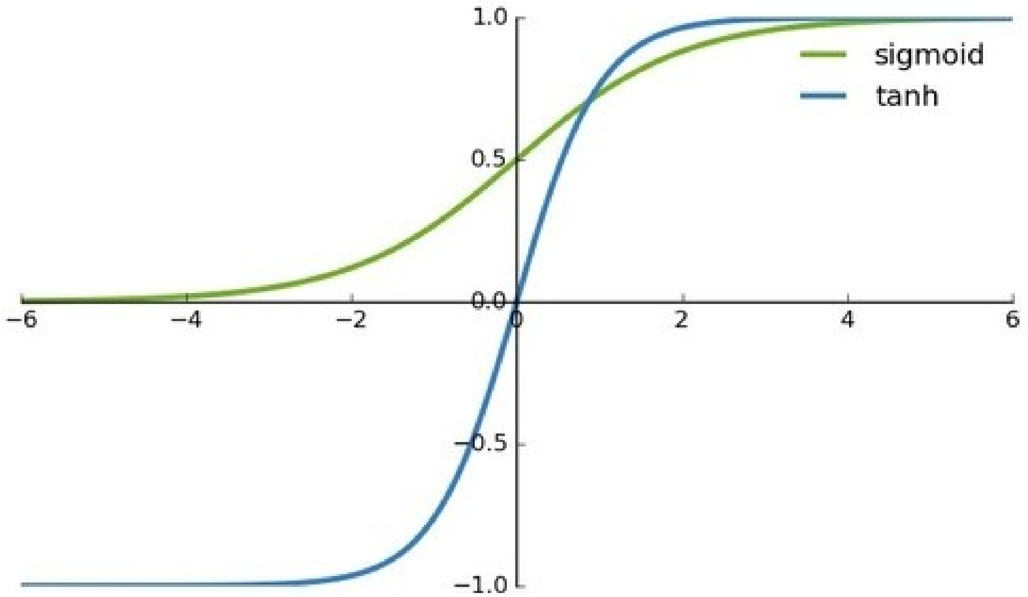


Image Credit:https://www.quora.com/What-are-the-benefits-of-a-tanh-activation-function-over-a-standard-sigmoid-activation-function-for-artificial-neural-nets-and-vice-versa

Other Functions

- Any function could theoretically be used as a classifier
  - Sigmoid and tanh work well with the gradient descent and ML algorithms

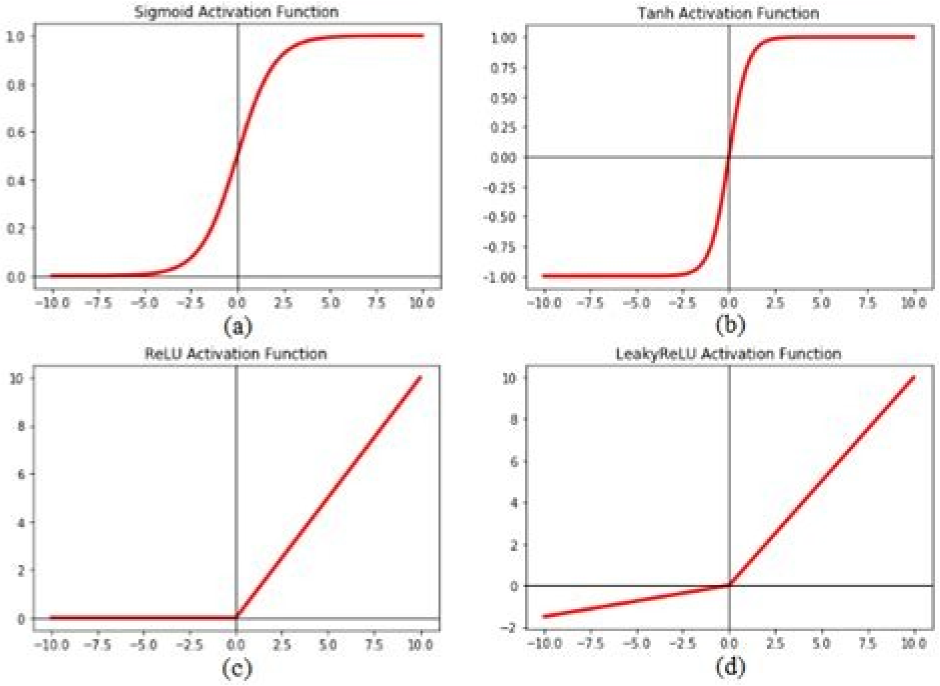


Image Credit:https://www.researchgate.net/figure/Plot-of-different-activation-functions-a-Sigmoid-activation-function-b-Tanh_fig4_339991922

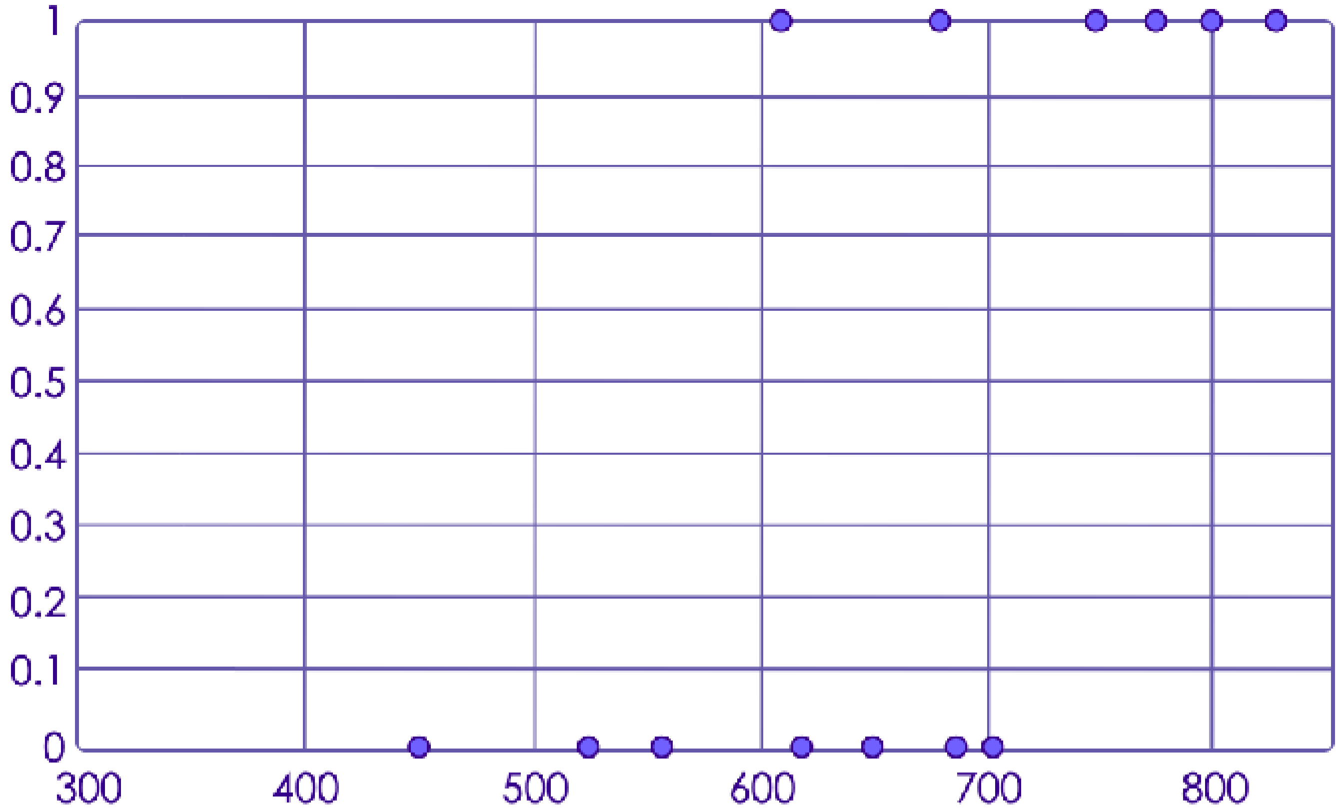
Example: Credit Approval Data

- Given this historical data on credit score and credit card approvals
  - What is the chance some one with score of 700 getting a credit card approved?

Credit Score	Approved?
560	No
750	Yes
680	Yes
650	No
450	No
800	Yes
775	Yes
525	No
620	No
830	Yes
610	Yes

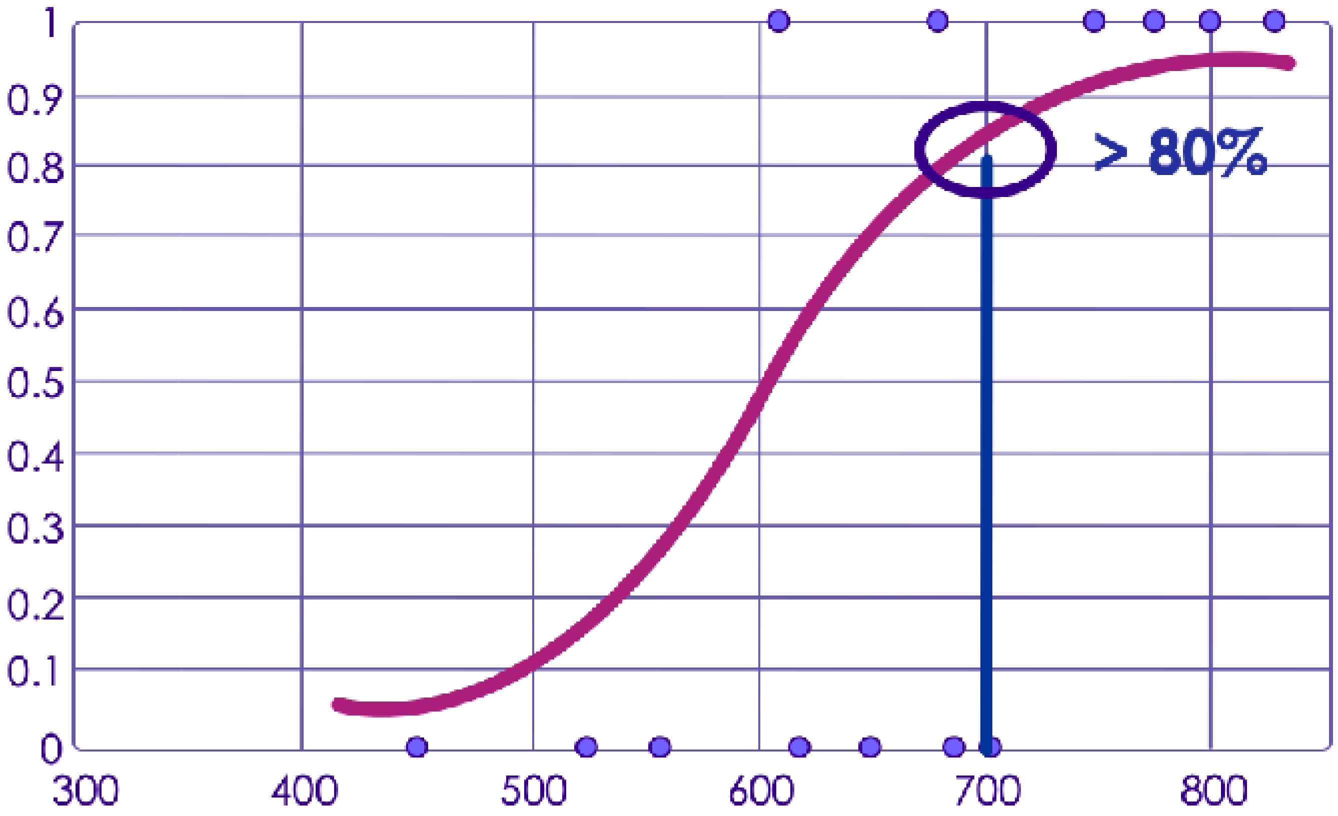
Example: Credit Approval Data

- Graphing, the data is not linearly separable with any vertical line



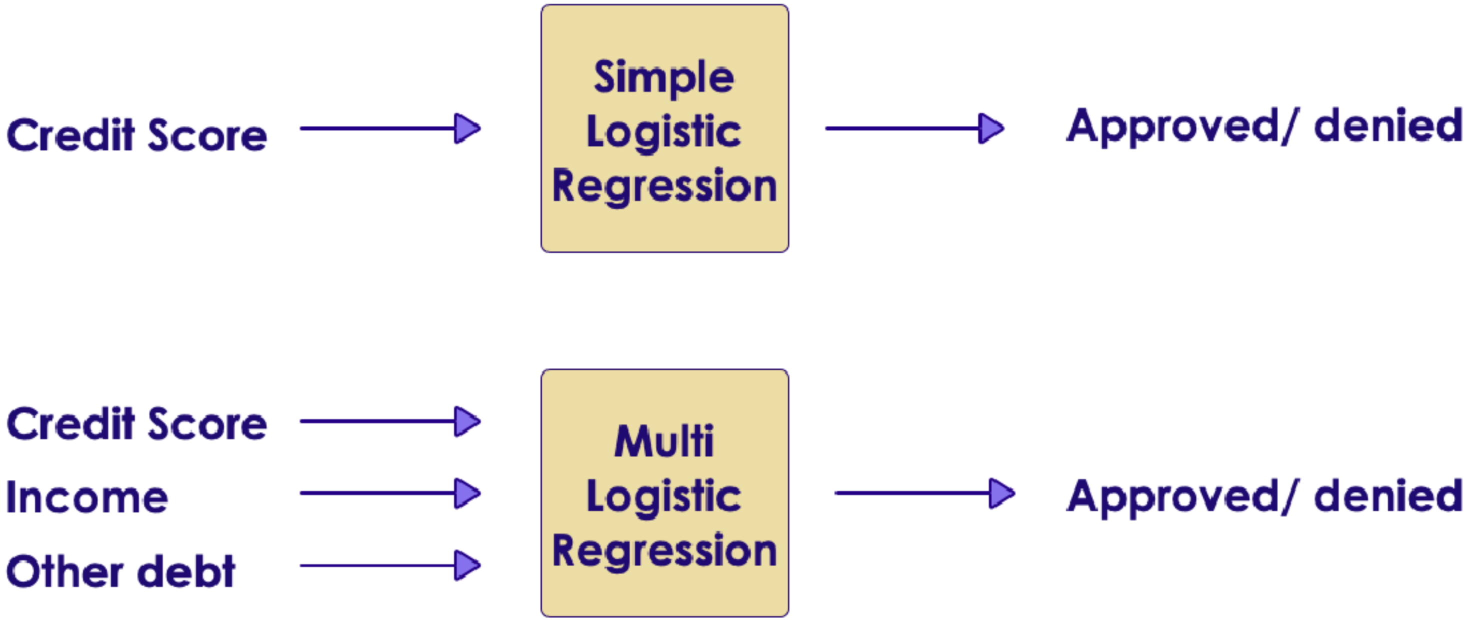
Applying Logistic Regression

- Want to find a stochastic function for data between 600-700
  - Clearly, adjusting the slope of the function will improve performance



Multiple Logistic Regression

- We can have multiple input factors (independent variables) determining a classification as
  - This is called ‘multiple logistic regression’



Multiple Logistic Regression

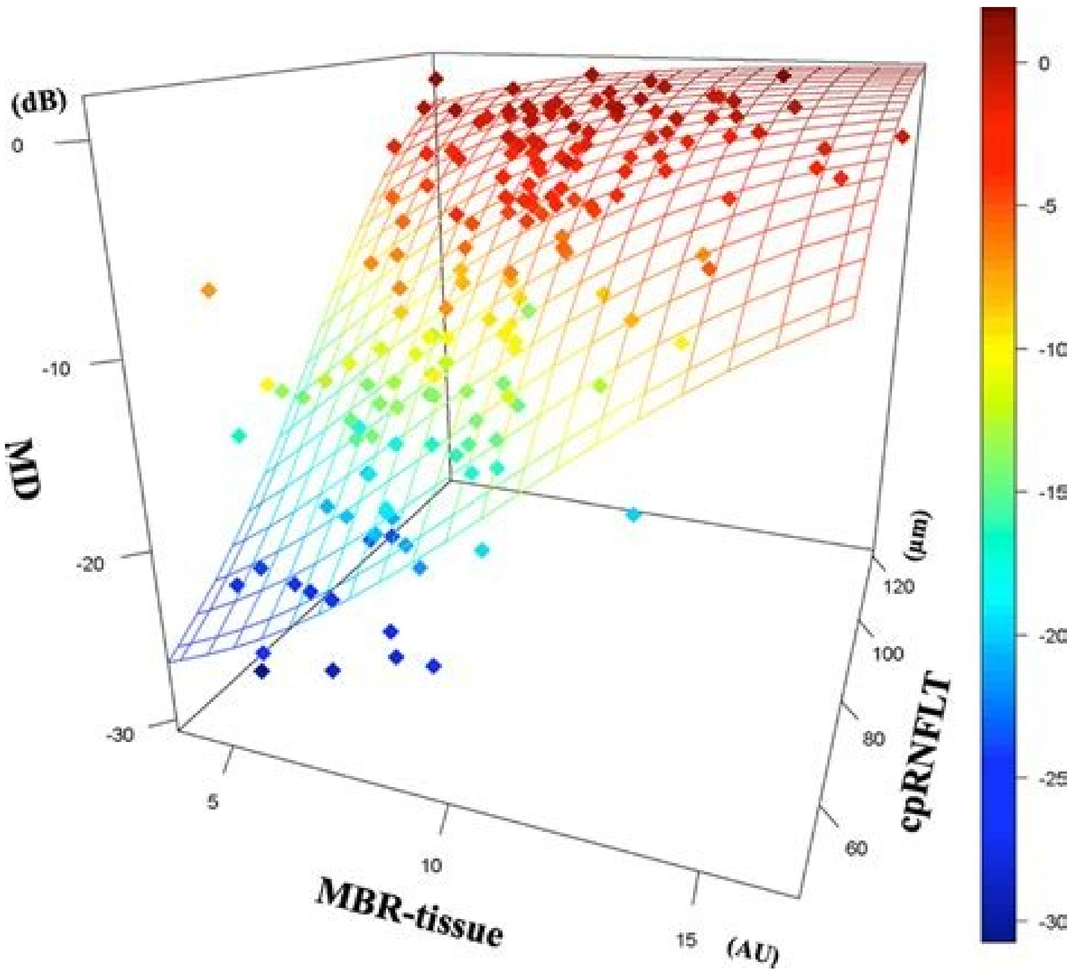


Image Credit: https://www.researchgate.net/figure/Stepwise-multiple-logistic-regression-analyses-of-the-relationships-among-the-mean_fig3_338692557

Preparing Data for Logistic Regression

- Binary Output Variable
  - Logistic Regression predicts probability of an instance belonging to default class
  - These are mapped into 0 or 1 classification based on a threshold value
- Remove noise
  - Remove outliers from input data
- Remove highly correlated inputs to avoid overfitting
- Failure to converge
  - Highly correlated input
  - Data is very sparse (lots of zeros in data)

Multinomial Logistics

- Data output can be a set of discrete categories

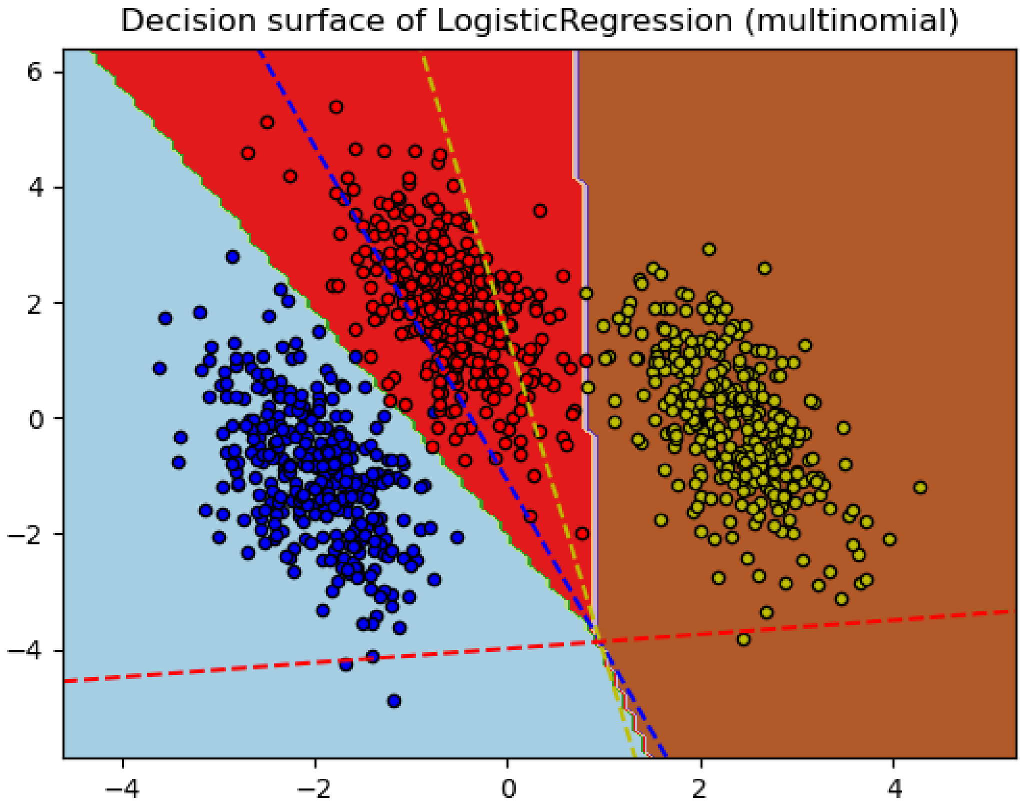


Image Credit: https://w10schools.com/posts/233818_Plot-multinomial-and-One-vs-Rest-Logistic-Regression

Multinomial Logistics

- The final surface is a combination of the individual stochastic functions

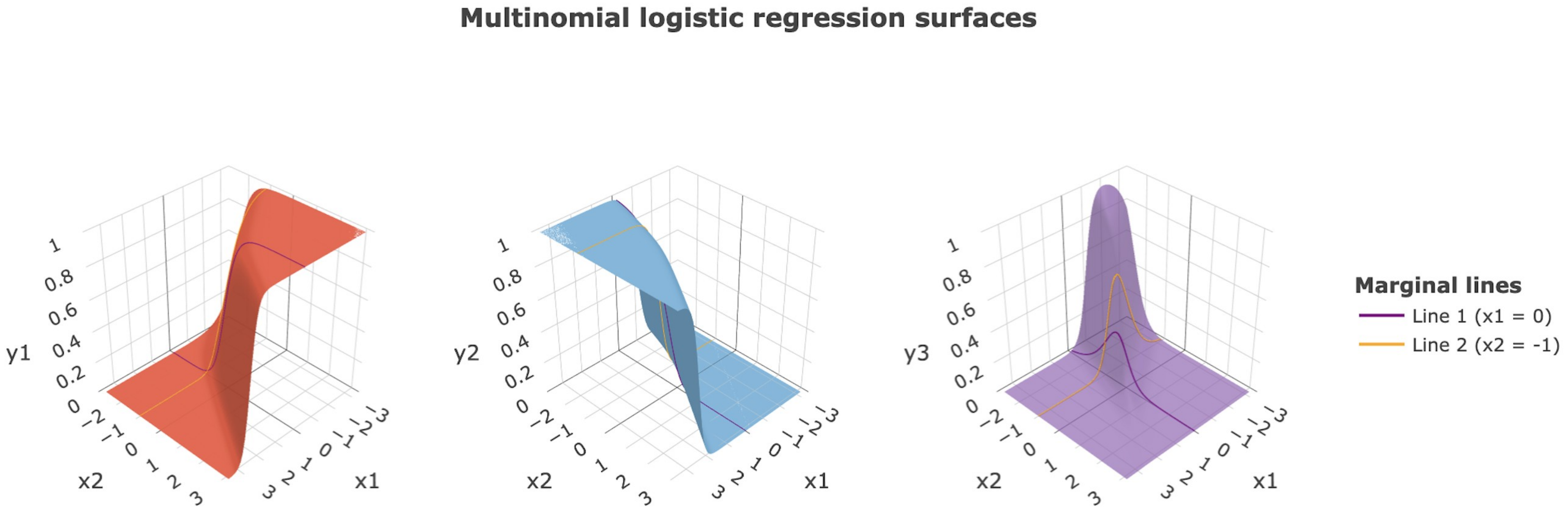


Image Credit: https://jakejing.github.io/posts/2022/09/blog-post-1/

Confusion Matrix and ROC Curve

- Confusion Matrix / Error Matrix
  - Binary classifier picks one of two outcomes (spam / not-spam)
  - Say we are classifying 10 emails (6 spam, 4 not-spam)

Columns (predicted)

Rows (actual)

Classification =>	Spam (predicted)	Not Spam (predicted)
Actual Spam (6 total)	4 count 66% accuracy Correct **True-Positive Rate (sensitivity)**	2 count 33% Incorrect **False negative rate (miss rate)**
Actual Not Spam (4 total)	1 count 25 % Incorrect **False-positive rate (fall-out)**	3 count 75% Correct **True negative rate (specificity)**

Confusion Matrix: More Than 2 Outcomes

		Predicted		
Actual		Cat	Dog	Rabbit
	Cat (8)	5	3	0
	Dog (6)	2	3	1
	Rabbit (13)	0	2	11

Interpreting Confusion Matrix (True/False Positives/Negatives)

		Predicted Condition	
		Predicted Positive	**Predicted Negative**
Actual condition (a cancer diagnostic)	**Positive (has caner)**	**True positive** - Patients who have cancer are correctly identified	**False negative** Miss rate - A cancer patient is missed - Guilty prisoner was not convicted
	Negative (doesn't have cancer)	**False Positive** Sensitivity - A healthy patient is flagged incorrectly - False alarm - 'Crying wolf' - Hiring someone who is not qualified	**True negative** - Patients who do not have cancer are correctly identified

Confusion Matrix: Accuracy / Error Rate

- Accuracy
  - Overall how accurate is the model?
  - = (TP + TN) / total
  - = (90 + 70) / 200
  - = 0.8 or 80%
- Misclassifications / Error rate
  - How wrong is the model?
  - = (FP + FN) / total
  - = (10 + 30) / 200
  - = 0.2 or 20%
  - = 1 - accuracy

		Predicted Condition	
		Predicted Positive	**Predicted Negative**
Actual condition (n = 200)	**Positive (n = 120)**	True positive (n = 90)	False negative (n = 30)
	Negative (n = 80)	False Positive (n = 10)	True negative (n = 70)

Confusion Matrix: TPR / FPR

- True Positive Rate (TPR) /Sensitivity / Hit Rate / Recall
  - How often model predicts ‘positive’ as ‘positive’ (correctly) ? – actual positive
  - = TP / (TP + FN)
  - = 90 / 120
  - = 0.75 or 75%
- False Positive Rate (FPR)
  - How often model predicts ‘negative’ as ‘positive’ (incorrectly) – actual negative = FP / (FP + TN)
  - = 10 / 80
  - = 0.125 or 12.5%

		Predicted Condition	
		Predicted Positive	**Predicted Negative**
Actual condition (n = 200)	**Positive (n = 120)**	True positive (n = 90) TPR = 75%	False negative (n = 30)
	Negative (n = 80)	False Positive (n = 10) FPR = 12.5%	True negative (n = 70)

Confusion Matrix: Specificity / Precision / Prevalence

- Specificity
  - How often model predicts negative' as negative' (correctly)? – actual no
  - = TN / (TN + FP)
  - = 70 / (70 + 10)
  - = 0.875 or 87.5 %
  - = 1 - FPR
- Precision / Positive Predictive Value (PPV)
  - When model predicts ‘positive’ how often it is right? – true / predicted positive= TP / (TP + FP)
  - = 90 / (90 + 10)
  - = 0.9 or 90%

		Predicted Condition	
		Predicted Positive	**Predicted Negative**
Actual condition (n = 200)	**Positive (n = 120)**	True positive (n = 90) TPR = 75%	False negative (n = 30)
	Negative (n = 80)	False Positive (n = 10) FPR = 12.5%	True negative (n = 70) Specificity = 87.5%

Confusion Matrix: PPV / Null Error Rate

- Prevalence
  - How often does ‘positive’ occurs in our sample
  - = actual positive / total
  - = 120 / 200
  - = 0.6 or 60%
- Null Error Rate
  - How often would the model be wrong if it always predicted the majority class?
  - Here our majority = Positive
  - If we always predicted ‘positive’ we would be wrong 80 times (negative)
  - = 80/200
  - = 40% of time

		Predicted Condition	
		Predicted Positive	Predicted Negative
Actual condition (n = 200)	**Positive (n = 120)**	True positive (n = 90) TPR = 75%	False negative (n = 30)
	Negative (n = 80)	False Positive (n = 10) FPR = 12.5%	True negative (n = 70) Specificity = 87.5%

F-Score

- Confusion Matrix: F-Score
  - While precision and recall are very important measures, looking at only one of them will not provide us with the full picture.
  - One way to summarize them is the f-score or f-measure, which is with the harmonic mean of precision and recall
  - F = 2 * (Precision * Recall) / (Precision + Recall)

Other Measures Associated with Confusion Matrix

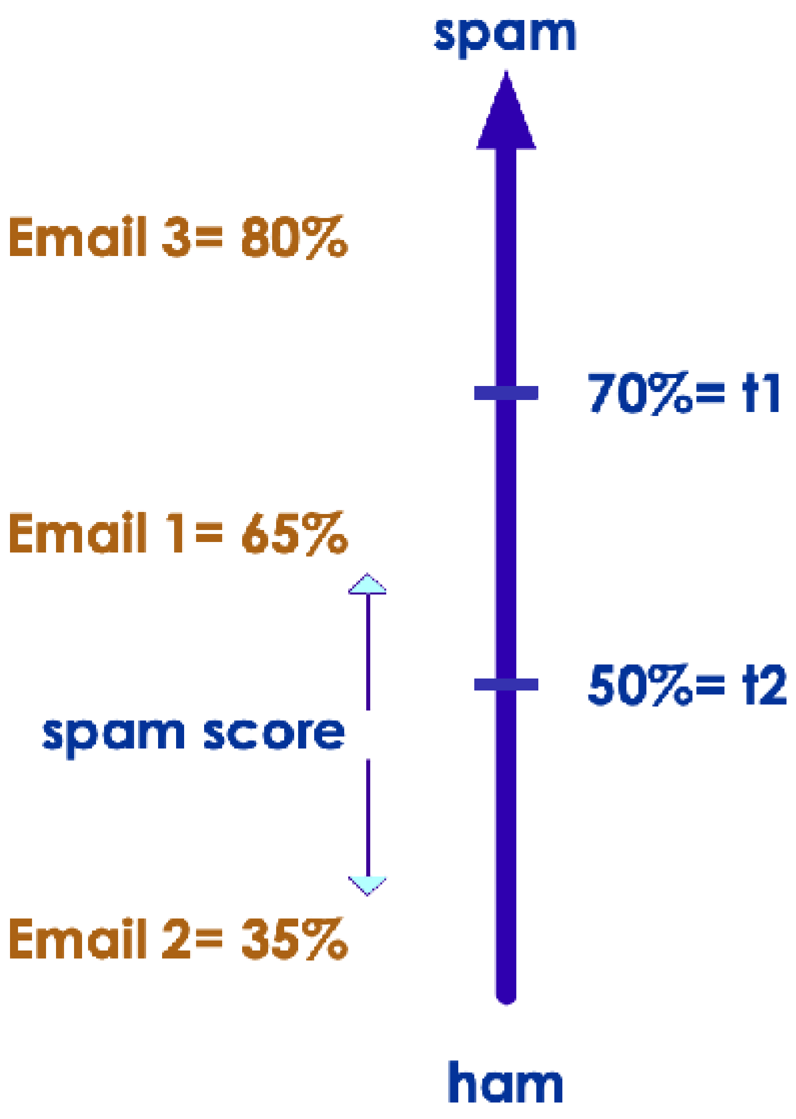
Sources: [21][22][23][24][25][26][27][28][29] view · talk · edit

		Predicted condition			
	Total population $= P + N$	Positive (PP)	Negative (PN)	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	Positive (P)	**True positive (TP), hit**	**False negative (FN), type II error, miss, underestimation**	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate $= \frac{FN}{P} = 1 - TPR$
	Negative (N)	**False positive (FP), type I error, false alarm, overestimation**	**True negative (TN), correct rejection**	False positive rate (FPR), probability of false alarm, fall-out $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$	Negative predictive value (NPV) $= \frac{TN}{PN}$ $= 1 - FOR$	Markedness (MK), deltaP (Δp) $= PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2 PPV \times TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

Image Credit:https://en.wikipedia.org/wiki/Confusion_matrix

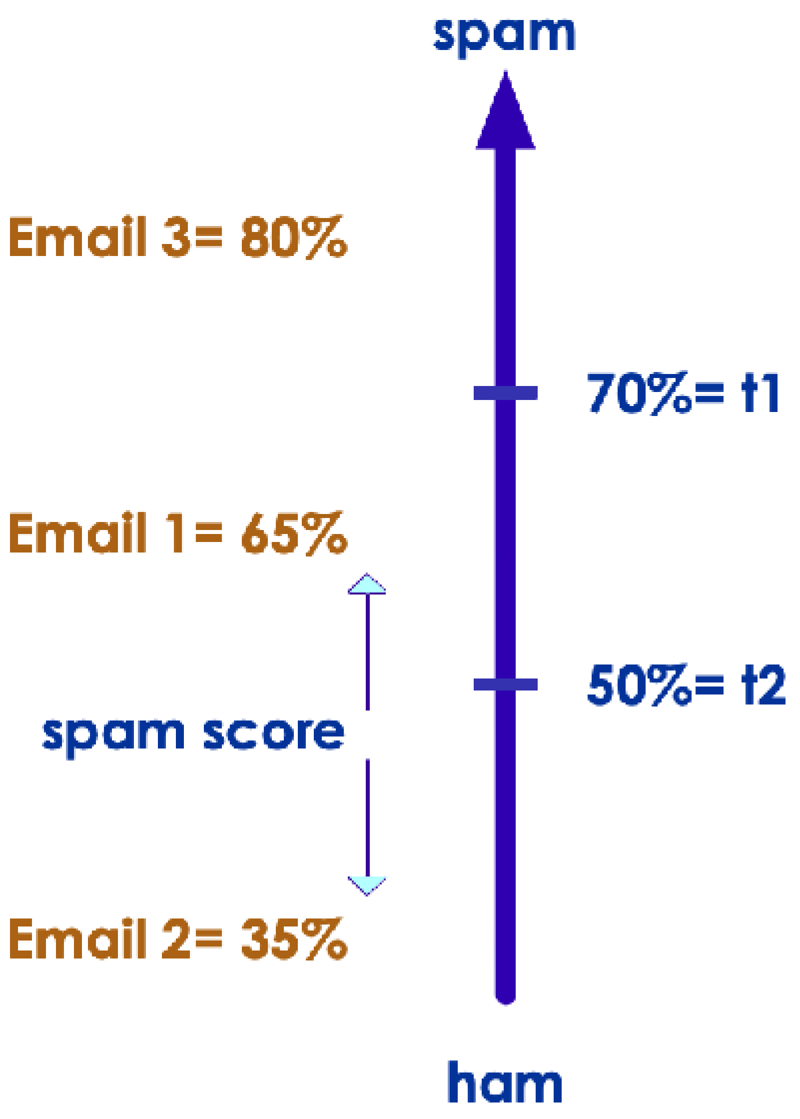
Threshold

- Our spam classifier provides a ‘spam probability’ for each email
  - Probability is between 0.0. and 1.0 (or 0 to 100%)
  - 1.0 definitely spam
  - 0.0 definitely not spam
  - When an email’s ‘spam score’ is above a certain number we mark it as spam
- This is called ‘threshold’



Threshold

- If threshold is lower (say 50%)
  - more emails will be classified as spam (email1, email3)
  - Users will miss emails (as they are in Spam folder)
- If threshold is higher (70%)
  - Fewer emails will be classified as spam (email3)
  - Users will see more spam emails be in Inbox
- We need to find the sweet spot for threshold



Threshold

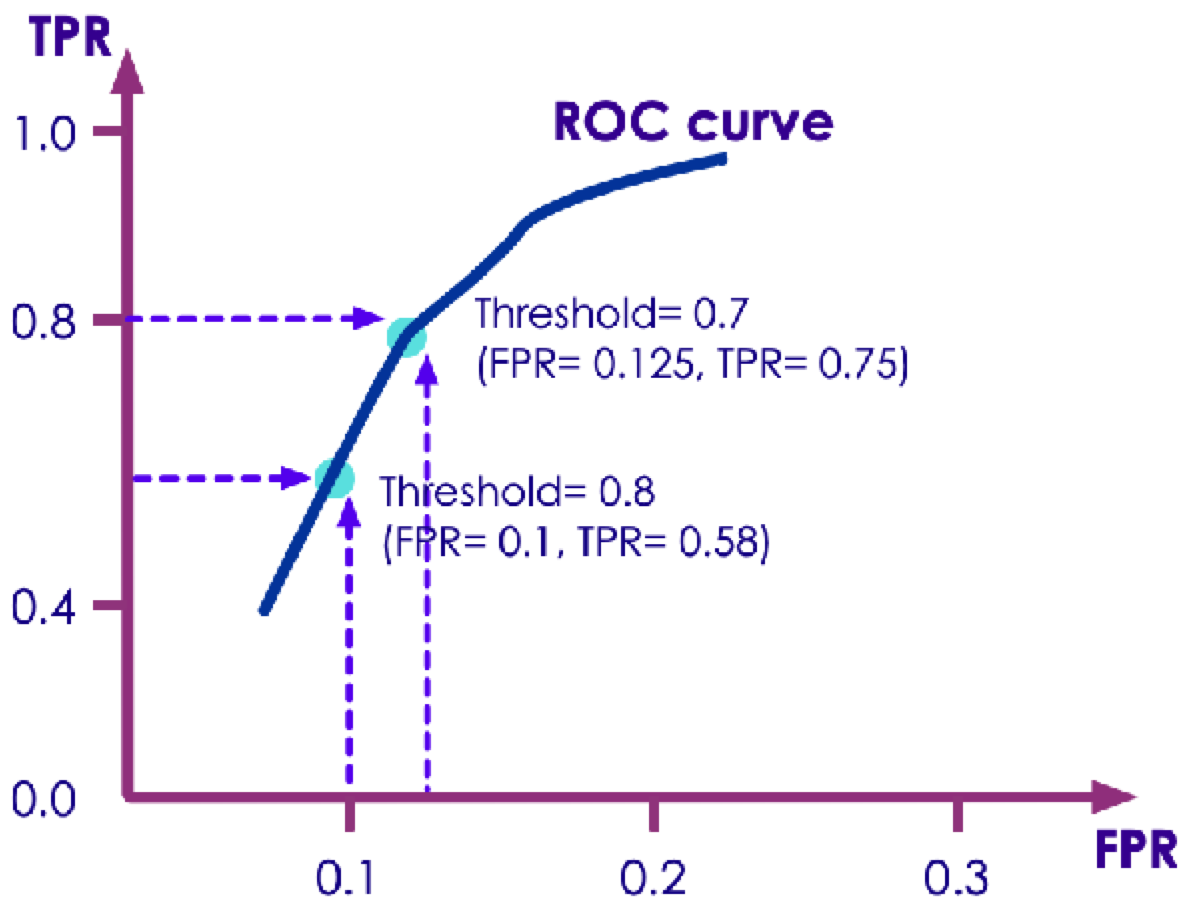
- In first table our threshold is 0.7
  - 90 emails are correctly predicted as spam
- Next table, our threshold is higher 0.8
  - Only 70 emails are classified as spam Lower TPR

		Predicted Condition	
Threshold = 0.7		**Predicted Spam**	**Predicted Not Spam**
Actual condition (total = 200)	**Spam (n = 120)**	True positive (n = 90) TPR = TP / positive = 90/120 = **75%**	False negative (n = 30)
	Not Spam (n = 80)	False Positive (n = 10) FPR = FP / negative = 10/80 = **12.5%**	True negative (n = 70)

		Predicted Condition	
Threshold = 0.8 (higher)		**Predicted Spam**	**Predicted Not Spam**
Actual condition (total = 200)	**Spam (n = 120)**	True positive (n = 70) TPR = TP / positive = 70 / 120 = 58.33%	False negative (n = 50)
	Not Spam (n = 80)	False Positive (n = 8) FPR = FP / negative = 8 / 80 = 10%	True negative (n = 72)

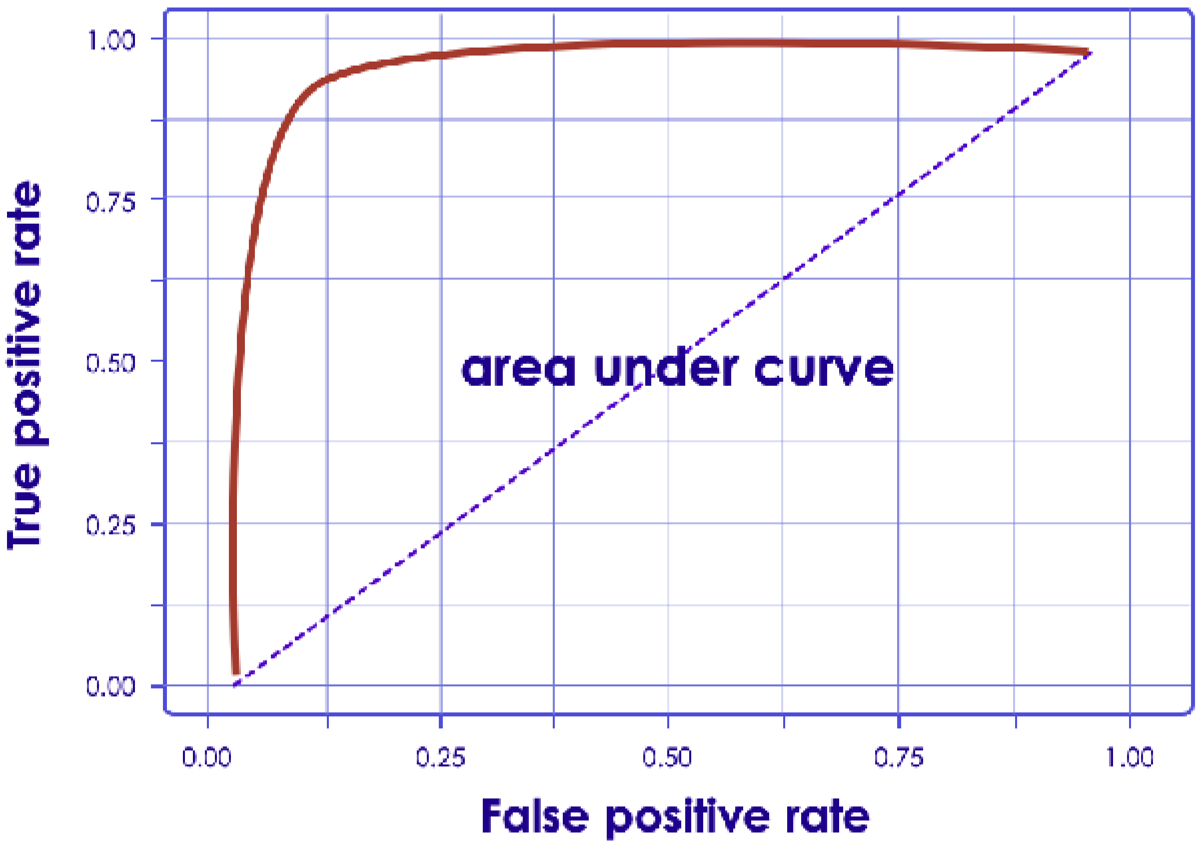
ROC Curve

- Receiver Operator Characteristic (ROC) curve
  - a graphical plot used to show the diagnostic ability of binary classifiers.
- Y-axis: True Positive Rate (TPR)
  - Actual=positive, predicted=positive
  - Correct!
- X-axis: False Positive Rate (FPR)
  - Actual=negative, predicted=positive
  - Incorrect!
- 0.0 <= TPR & FPR <= 1.0
- Plot TPR / FPR while varying ‘threshold’



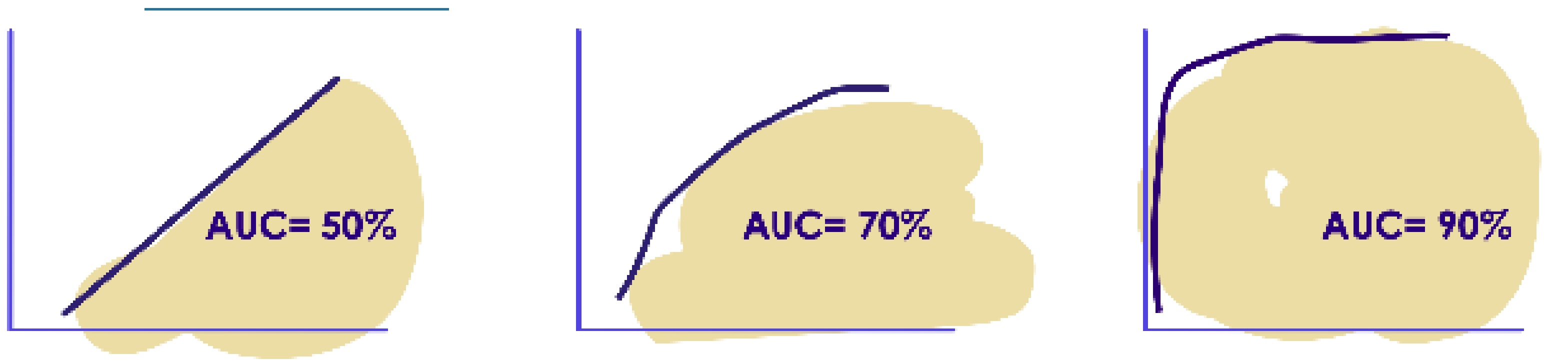
ROC Curve

- Shows tradeoff of TPR (sensitivity) vs. FPR (1 - specificity)
- The closer to top-left , the more accurate the model
  - Upper left corner (0,1) = perfect classification!
- The closer to middle line (45 degree) the less accurate the test
- Middle line represents: random classification (50%)



ROC Area Under the Curve

- Measures the percentage of area ‘under the curve’
  - AUC is between 0 and 1.0
  - Higher AUC –> more accurate the model
- See 3 scenarios below
  - Leftmost is bad (50%), Middle: OK (70%), Rightmost: very good (90%)



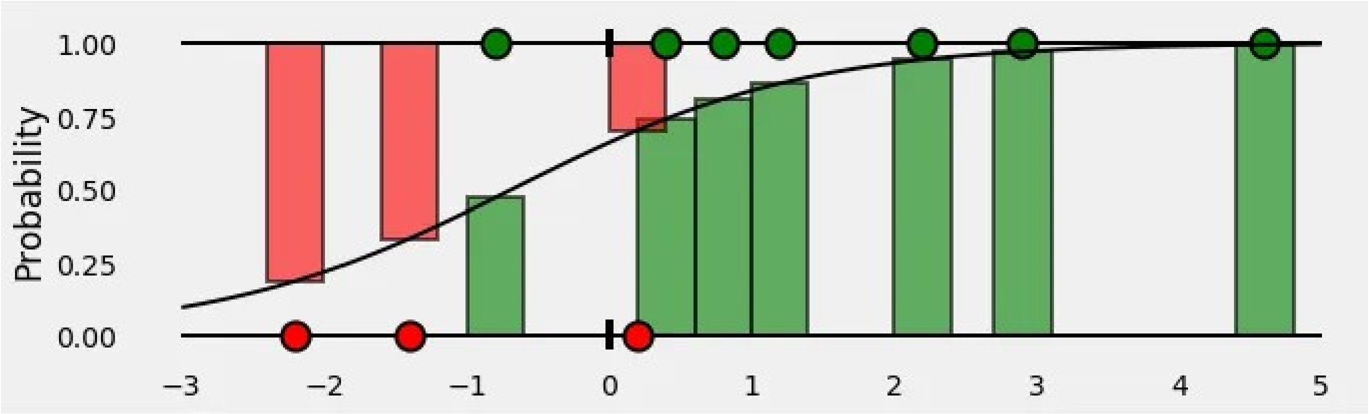
Logistic Regression Training

- We use the same process we saw for regression
  - We start with a set of coefficients
  - We have a loss function we can compute
  - We then apply gradient descent to compute a better set of coefficients
- Loss functions for classification
  - Binary Class Entrophy
  - Categorical Crossentropy / Sparse Categorical Crossentropy
  - Negative Log Likelihood
  - Margin Classifier
  - Soft Margin Classifier

Loss Functions Binary Entrophy

- Binary Class Entrophy
  - Measures the divergence of probability distributions between actual and predicted values

$$E = -\frac{1}{n}\sum_{i=1}^{n}\left[y^{(i)}\log(\hat{y}^{(i)}) + (1-y^{(i)})\log(1-\hat{y}^{(i)})\right]$$



Strengths, Weaknesses, and Parameters

- Strengths
  - Relatively simple to understand
  - Linear algorithm -> Fast learner
  - Works well on high dimensional (100s of features) datasets
  - Very scalable to large data sets
- Weaknesses
  - Can underfit some times
- Parameters
  - Use regularization to minimize overfitting

End of Module