CSCI 556 Data Analysis & Visualization

Classification

Instructor: Dr. Jinoh Kim

Topics

- 1R learner
- Naïve Bayes approach
- Logistic regression classifier
- Evaluating classification models
- Strategies for unbalanced data

Classification

- One of the most important forms of prediction
- Binary classification: want to know the predicted probability that a case belongs to a class
- General approach:

1. Establish a cutoff probability for the class of interest above which we consider a record as belonging to that class.

2. Estimate (with any model) the probability that a record belongs to the class of interest.

3. If that probability is above the cutoff probability, assign the new record to the class of interest.

Inferring rudimentary rules

- 1R rule learner: learns a 1-level decision tree
  - A set of rules that all test one particular attribute that has been identified as the one that yields the lowest classification error
- Basic version for finding the rule set from a given training set (assumes nominal attributes):
  - For each attribute,
    - Make one branch for each value of the attribute
    - To each branch, assign the most frequent class value of the instances pertaining to that branch
    - Error rate: proportion of instances that do not belong to the majority class of their corresponding branch
  - Choose attribute with lowest error rate

1R learner example with Weather data

Data instances

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

1R rules

Attribute	Rules	Errors	Total errors
Outlook	Sunny → No	2/5	4/14
	Overcast → Yes	0/4	
	Rainy → Yes	2/5	
Temp	Hot → No*	2/4	5/14
	Mild → Yes	2/6	
	Cool → Yes	1/4	
Humidity	High → No	3/7	4/14
	Normal → Yes	1/7	
Windy	False → Yes	2/8	5/14
	True → No*	3/6	

* indicates a tie

Simple probabilistic modeling

- "Opposite" of 1R: use all the attributes
- Two assumptions:
  - Attributes are equally important
  - Attributes are statistically independent (given the class value)
- This means knowing the value of one attribute tells us nothing about the value of another takes on (if the class is known)
- Independence assumption is almost never correct!
- But … this scheme often works surprisingly well in practice
- The scheme is easy to implement in a program and very fast
- Known as *naïve Bayes*

Naïve Bayes

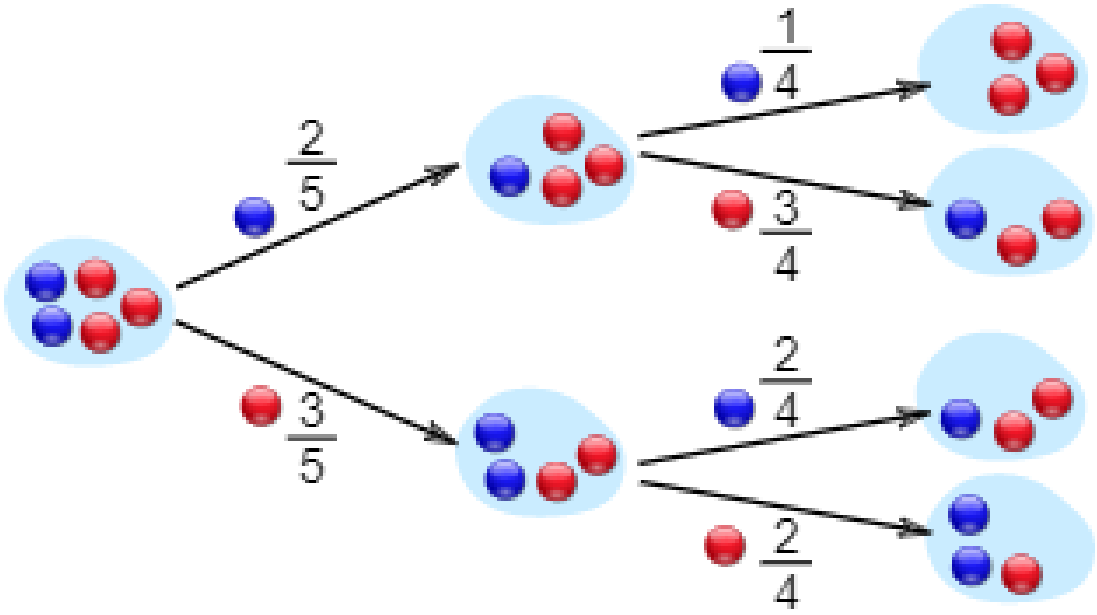
- Uses the probability of observing predictor (feature) values, given an outcome, to estimate the probability of observing outcome $Y = i$, given a set of predictor values
- Conditional probability: The probability of observing some event (say $X = i$) given some other event (say $Y = i$)
- Posterior probability: The probability of an outcome after the predictor information has been incorporated
- Prior probability: The probability of outcomes, not taking predictor information into account

Conditional probability

- Conditional probability: likelihood of an event occurring given that another event has already happened
- Notations:
  - P(A) = Probability of event A
  - P(B) = Probability of event B
  - P(A|B) = Probability of A given B

Conditional probability example

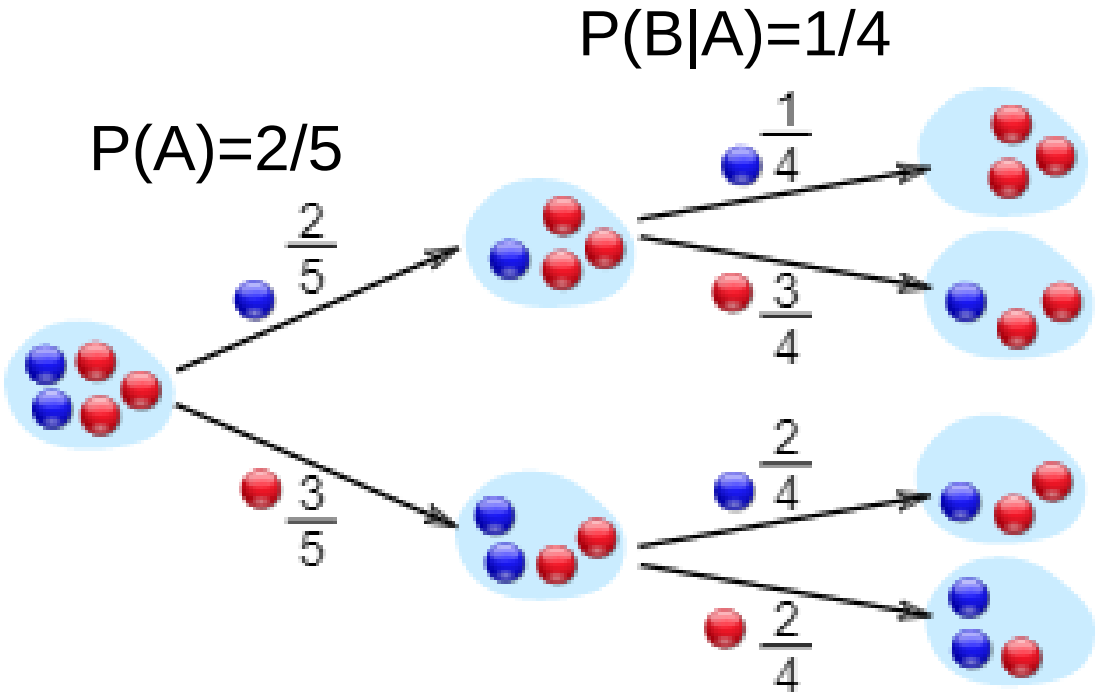
- Assume 2 blue and 3 red marbles are in a bag
- We take one marble from the bag, and then take another next.



Source: mathsisfun.com

Conditional probability example

- Event A is "get a Blue Marble first”
- Event B is "get a Blue Marble second"
- P(B|A) = Event B given Event A



Source: mathsisfun.com

Conditional probability math

P(A|B) = "probability of A given B"

Definition: $P(A|B) = \frac{P(A\ and\ B)}{P(B)}$

Similarly, $P(B|A) = \frac{P(B\ and\ A)}{P(A)}$

Thus, $P(A\ and\ B) = P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$

And, $P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)}$

Bayes' theorem

Likelihood: the probability of "B" being True, given "A" is True

Prior: the probability "A" being True. This is the knowledge.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Posterior: the probability of "A" being True, given "B" is True

Marginalization: the probability "B" being True.

Incomparability and independence

- Two *incomparable* events cannot be true simultaneously: $P(A\ and\ B) = 0$

$$P(A\ or\ B) = P(A) + P(B) - P(A\ and\ B) = P(A) + P(B)$$

- Two events are *independent*, if the realization of one is not linked in any way to the realization of the other: $P(A|B) = P(A)$ and $P(B|A) = P(B)$

$$P(A\ and\ B) = P(A|B) \cdot P(B) = P(A) \cdot P(B)$$

Naïve Bayes for classification

- Classification learning: What is the probability of the class given an instance?
- Probability of an event *H* given observed evidence *E:*

$$P(H|E) = \frac{P(E|H)\cdot P(H)}{P(E)}$$

  - Evidence *E* = instance's non-class feature values (i.e., a vector of E_i = <E_1, E_2, …, E_n>)
  - Event *H* = class value of instance
- Naïve assumption: evidence splits into predictors (features) that are conditionally *independent*
- Given *n* attributes, we can write Bayes' rule using a product of per-predictor probabilities:

$$P(H|E) = \frac{P(E_1|H) \times P(E_2|H) \times \cdots \times P(E_n|H) \times P(H)}{P(E)}$$

Example: Weather data

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Probabilities for weather data

Outlook			Temperature			Humidity			Windy			Play	
	Yes	*No*		*Yes*	*No*		*Yes*	*No*		*Yes*	*No*	*Yes*	*No*
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/	5/
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5	14	14
Rainy	3/9	2/5	Cool	3/9	1/5								

Summary (frequency)

Instances →

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Probabilities for weather data

Outlook			Temperature			Humidity			Windy			Play	
	Yes	*No*		*Yes*	*No*		*Yes*	*No*		*Yes*	*No*	*Yes*	*No*
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/	5/
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5	14	14
Rainy	3/9	2/5	Cool	3/9	1/5								

- A new day:

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

Likelihood of the two classes

For "yes" = $2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0053$

For "no" = $3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0206$

Conversion into a probability by "normalization":

P("yes") = 0.0053 / (0.0053 + 0.0206) = 0.205

P("no") = 0.0206 / (0.0053 + 0.0206) = 0.795

Naïve Bayes approach

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

← ***Evidence E***

$$P(yes \mid E) = P(Outlook = Sunny \mid yes)$$

$$P(Temperature = Cool \mid yes)$$

$$P(Humidity = High \mid yes)$$

$$P(Windy = True \mid yes)$$

$$P(yes) / P(E)$$

$$= \frac{2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14}{P(E)}$$

Probability of class "yes"

Naïve Bayes algorithm

1. For a binary response $Y = i$ ($i = 0$ or 1), estimate the individual conditional probabilities for each predictor $P(X_j \mid Y = i)$; these are the probabilities that the predictor value is in the record when we observe $Y = i$. This probability is estimated by the proportion of X_j values among the $Y = i$ records in the training set.

2. Multiply these probabilities by each other, and then by the proportion of records belonging to $Y = i$.

3. Repeat steps 1 and 2 for all the classes.

4. Estimate a probability for outcome i by taking the value calculated in step 2 for class i and dividing it by the sum of such values for all classes.

5. Assign the record to the class with the highest probability for this set of predictor values.

Logistic regression

- Analogous to multiple linear regression, except the outcome is binary (e.g., Yes/No, Pass/Fail, etc)
  - Output of linear regression is continuous
- Structured model approach with fast computational speed
- Example:
  - Linear regression: What is the expected exam score given the number of hours studying?
  - Logistic regression: Will it be pass of the exam given the number of hours studying?

Logistic response function

- Consider outcome variable the probability p that the label is a "1"
- If we simply model p as a linear function:

$$p = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_q x_q$$

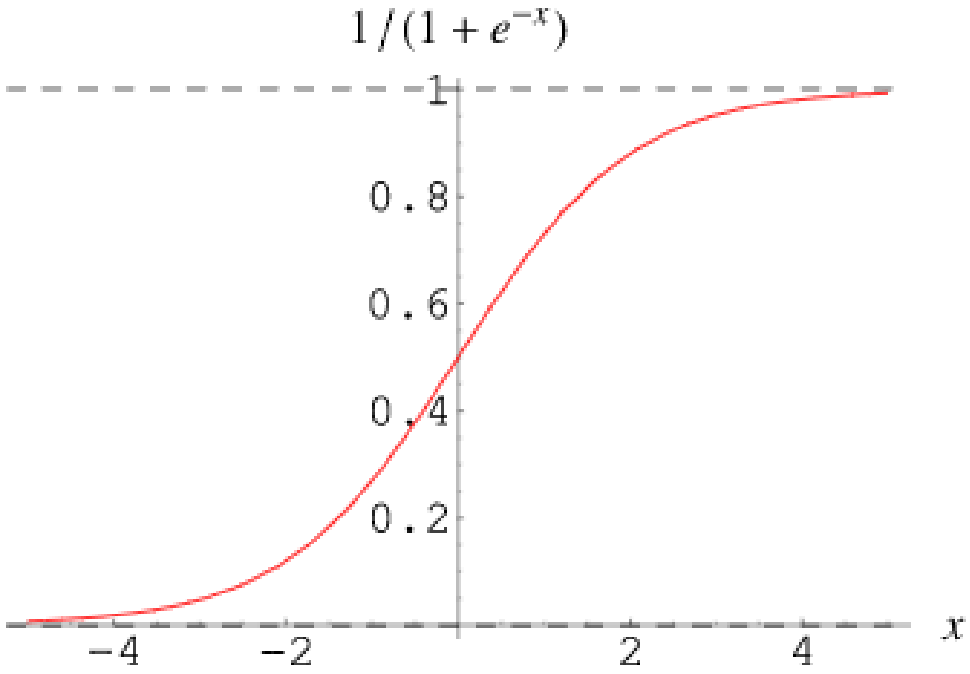
  - The probability p should be ranged between 0 and 1, but the linear model does not ensure it
- If we model p using a logistic function (Sigmoid function):

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_q x_q)}}$$

  - This transform ensures that the p stays between 0 and 1

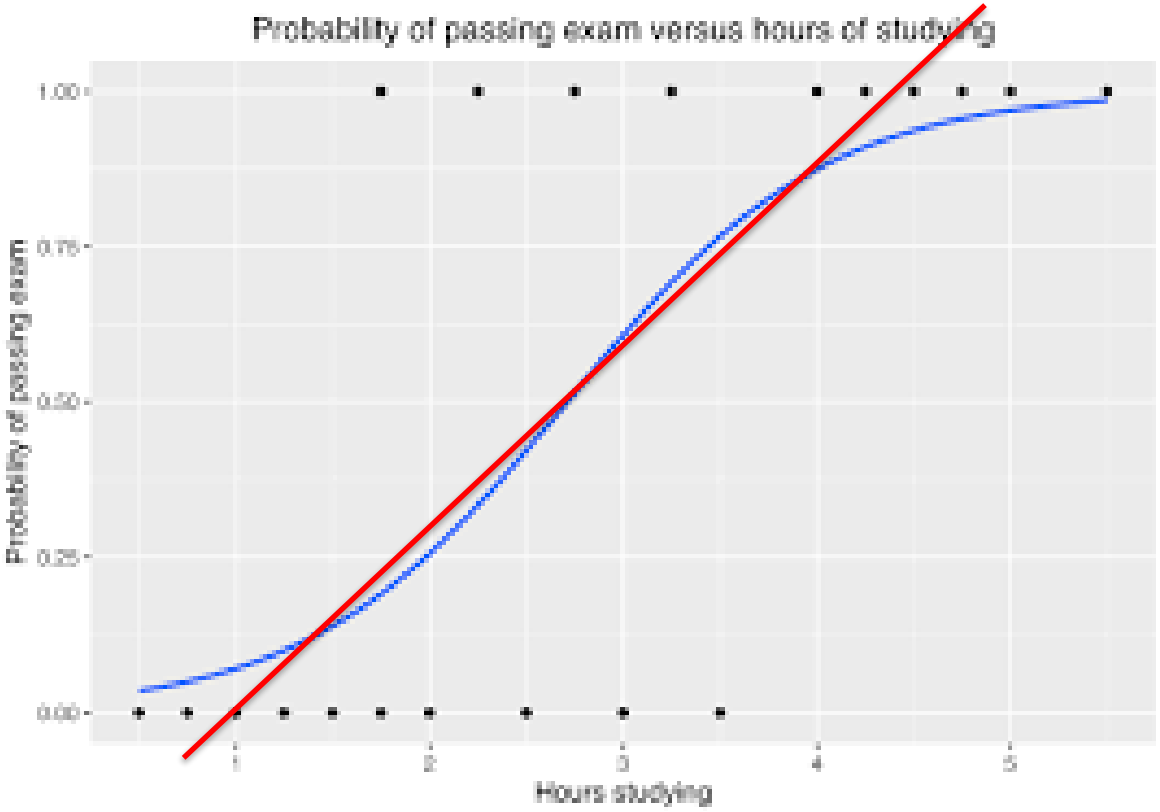
Sigmoid function

- $S(x) = \frac{1}{1+e^{-x}}$
- Also known as logistic function, always ranged between 0 and 1 (y-axis)



Graph from Wolfram MathWorld

Example: linear vs. logistic



Graph from Wiki

- Input variable is an integer value (hours studying)
- Outcome variable is binary (pass or not)
- Linear regression (red line) vs. logistic regression (blue line)

Odds ratio

- To get the exponential expression out of the denominator, we consider odds instead of probabilities.

$$p = \frac{1}{1 + e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_q x_q)}}$$

- Odds: Ratio of "successes" (1) to "nonsuccesses" (0)

$$\text{Odds}(Y = 1) = \frac{p}{1 - p}$$

- In terms of probabilities, odds are the probability of an event divided by the probability that the event will not occur
- Example: If the probability that a horse will win is 0.5, the probability of "won't win" is (1 – 0.5) = 0.5, and the odds are 1.0 (=0.5/(1-0.5))

Log-odds ("logit" function)

- Logit function is the logarithm of the odds ratio

$$log(Odds(Y = 1) = log\frac{p}{(1-p)}$$

- Takes input values in the range 0 to 1 and transforms them to values over the entire real number range
- Suppose z = $log\frac{p}{(1-p)}$, then we get:

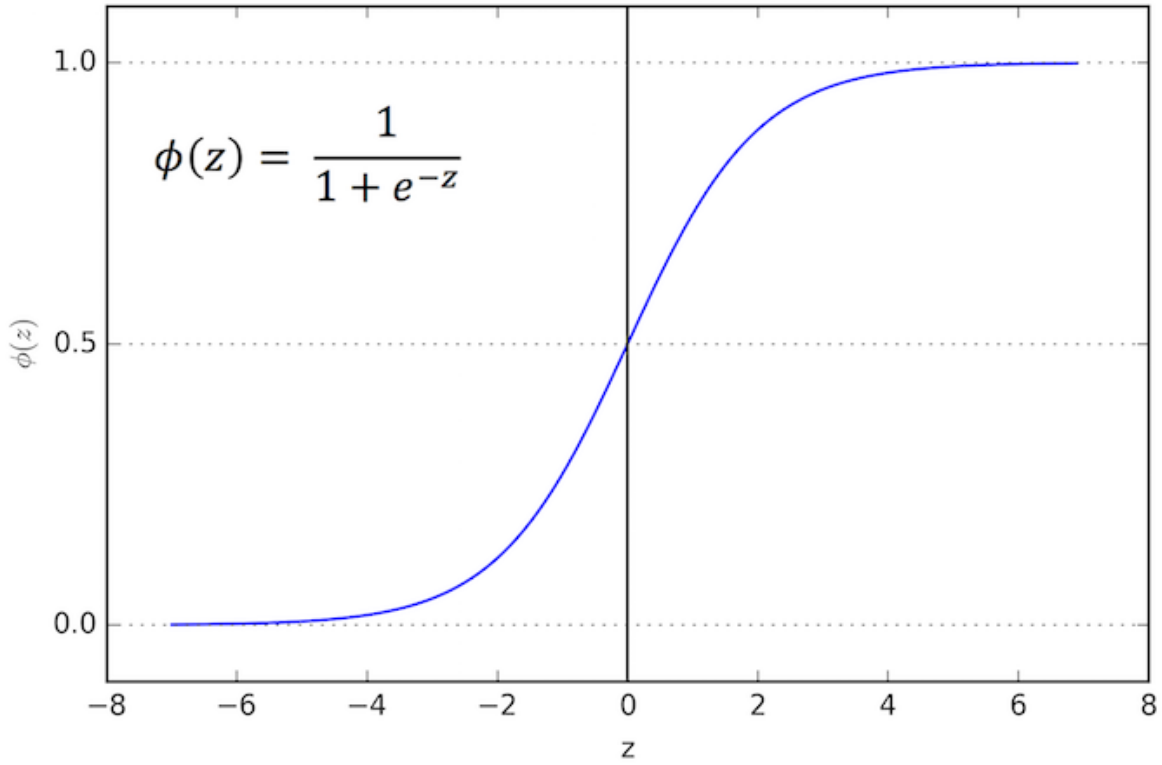
$$\frac{p}{1-p} = e^z$$
$$p = e^z(1-p)$$
$$p = e^z - pe^z$$
$$p + pe^z = e^z$$
$$p(1+e^z) = e^z$$
$$p = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}}$$

- As we want to model

$$p = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_q x_q)}}$$

- We get: $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_q x_q$

Logistic function



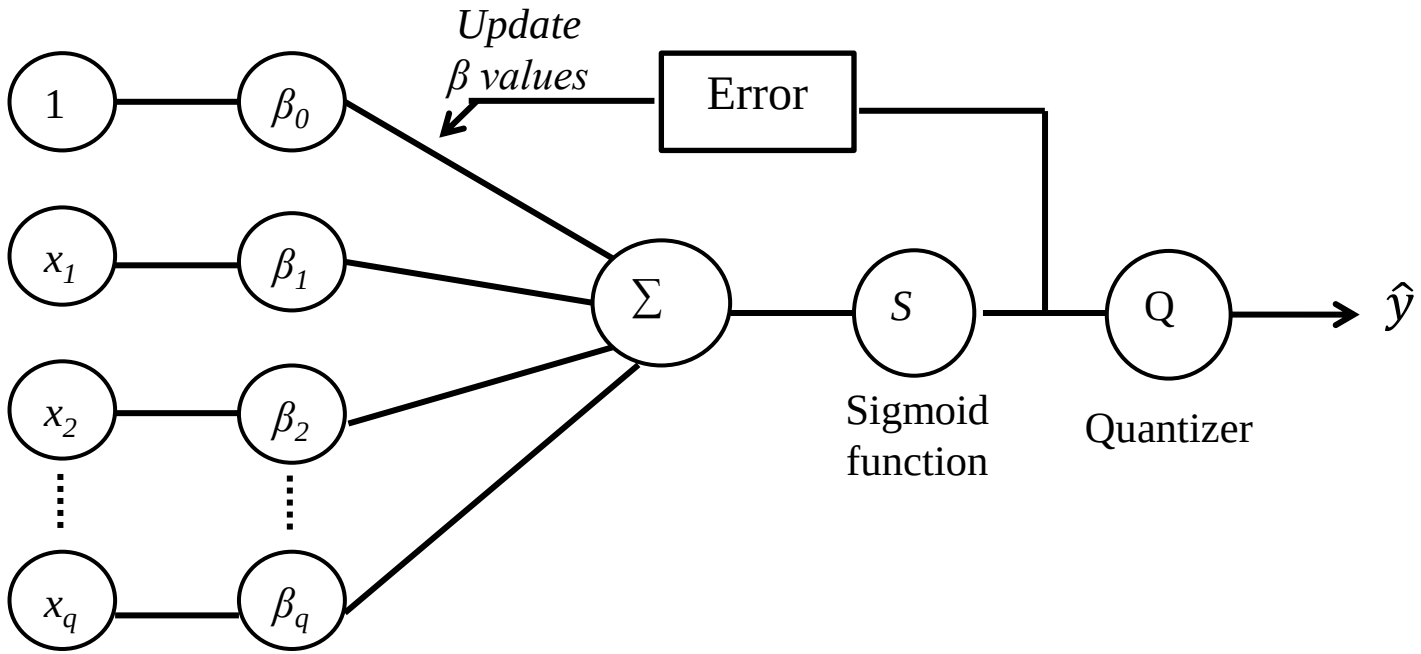
- $\phi(z)$ approaches 1 for $z \longrightarrow +\infty$, while $\phi(z)$ goes to 0 for $z \longrightarrow -\infty$.

Logistic regression classifier

- Map to a class label by applying a cutoff rule
- Any record with a probability greater than the cutoff (e.g., 0.5) is classified as a 1, that is:

$$\hat{y} = \begin{cases} 1 & if\ \emptyset(z) \geq 0.5 \\ 0 & otherwise \end{cases}$$

Building logistic regression model



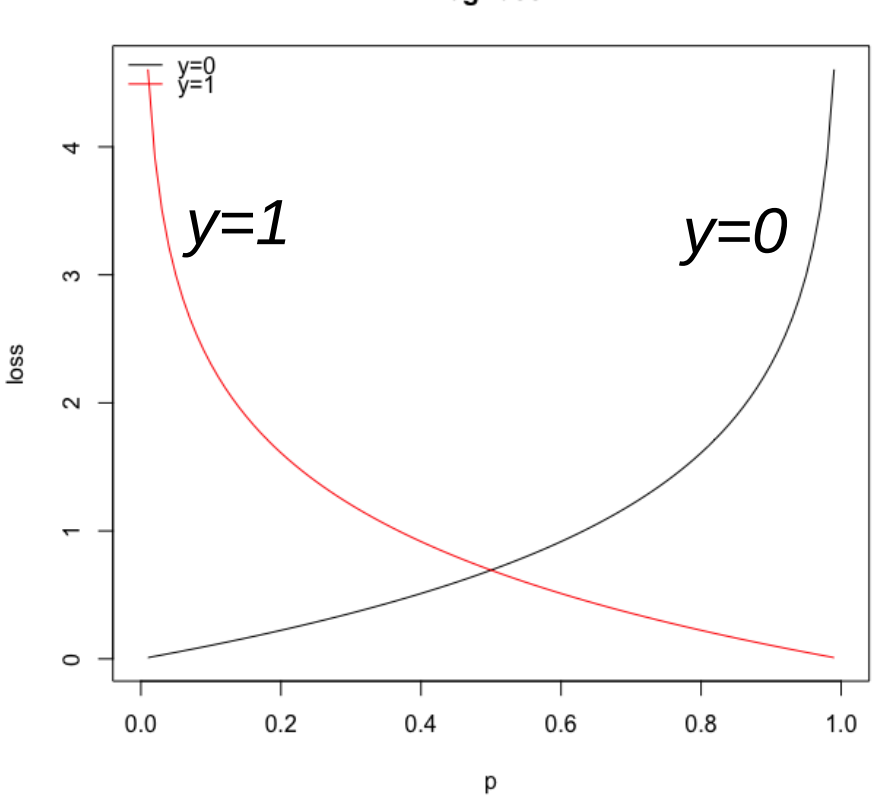
- Recall $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_q x_q$
- If the sigmoid output is wrong in training, β values are updated
- Quantizer produces $\hat{y}$ having either 0 or 1

Learning weights (β values)

- Loss function *L* is defined as:

$$L(\emptyset(z), y; q) = \begin{cases} -\log(\emptyset(z)) & if\ y = 1 \\ -\log(1 - \emptyset(z)) & if\ y = 0 \end{cases}$$

- Loss approaches 0 in case of correct prediction, while loss toes to infinity if the prediction is wrong
- The model is trained in a way to reduce the loss

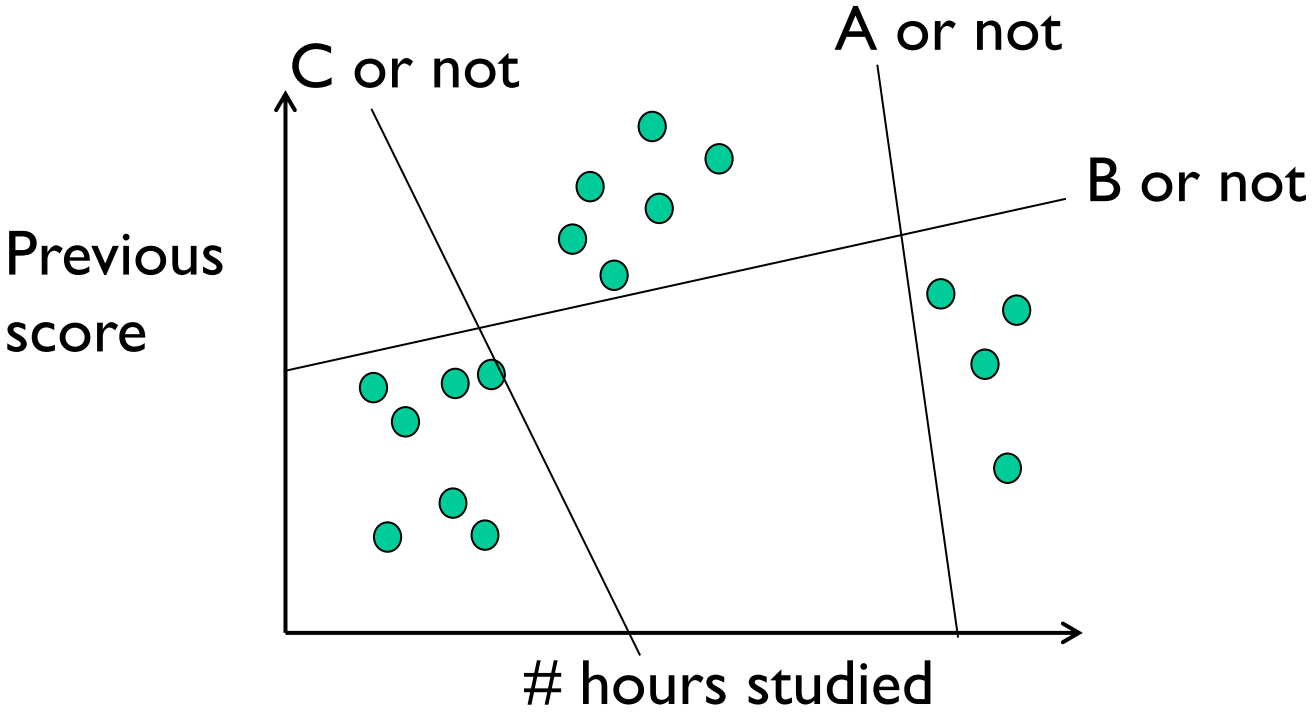


Fitting the model

- Linear regression is fit using least squares, and the quality of the fit is evaluated using RMSE and R-squared statistics
- Logistic regression relies on *maximum likelihood estimation* (MLE)
- MLE finds the solution such that the estimated log odds best describes the observed outcome
- The mechanics of the algorithm involve a quasi-Newton optimization that iterates between a scoring step (*Fisher's scoring*), based on the current parameters, and an update to the parameters to improve the fit (but beyond the scope)

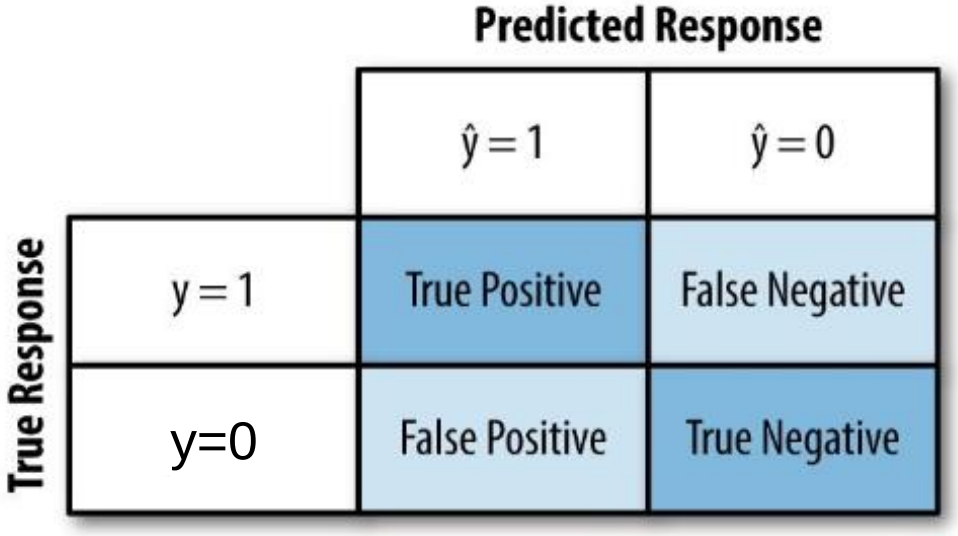
Multiple classes

- Logistic regression for two classes is also called binomial logistic regression
- What do we do when have a problem with *k* classes?
- Multinomial logistic regression: extension of the logistic regression model for multi-class classification



Evaluating classification models

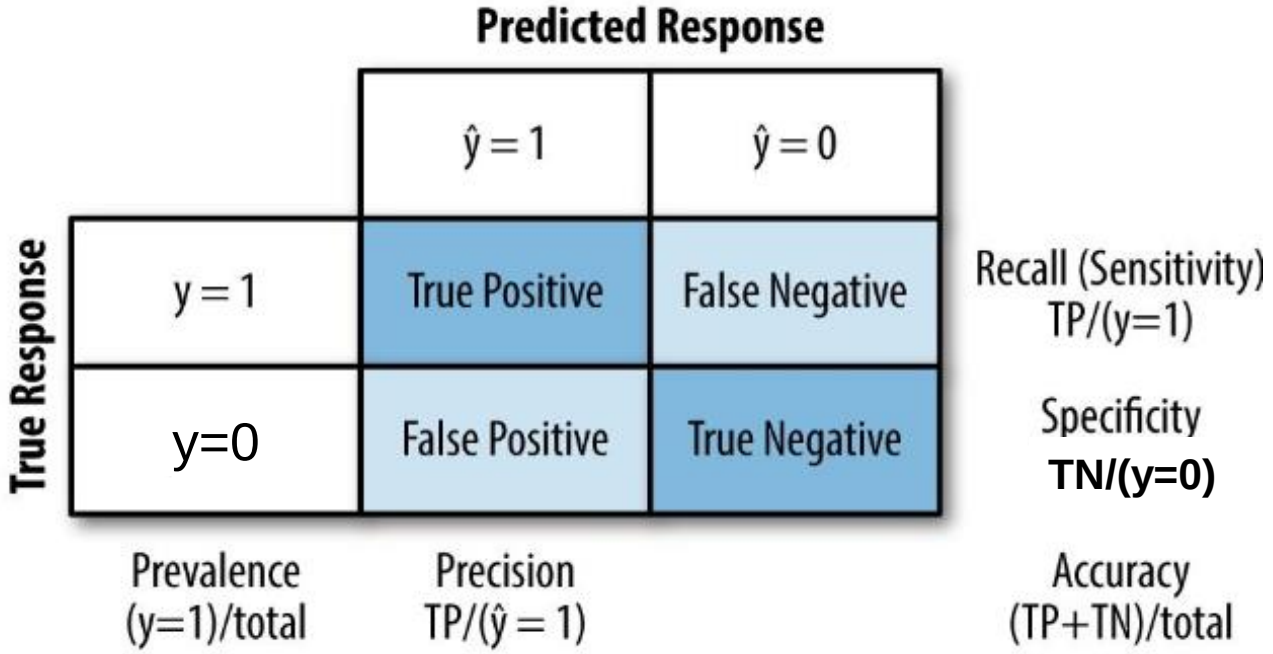
- Confusion matrix is a table showing the number of correct and incorrect predictions categorized by type of response



- "True": if actual value == predicted value ("False" otherwise)
- "Positive": if predicted value == 1 ("Negative" otherwise)

Evaluating classification models (2)

- Confusion matrix is a table showing the number of correct and incorrect predictions categorized by type of response



- Accuracy is simply a measure of total error:

$$\text{accuracy} = \frac{\sum \text{TruePositive} + \sum \text{TrueNegative}}{\text{SampleSize}}$$

Precision, recall, and others

- Precision: measures the accuracy of a predicted positive outcome

$$\text{precision} = \frac{\sum \text{TruePositive}}{\sum \text{TruePositive} + \sum \text{FalsePositive}}$$

- Recall (sensitivity): measures the strength of the model to predict a positive outcome — the proportion of the 1s that it correctly identifies

$$\text{recall} = \frac{\sum \text{TruePositive}}{\sum \text{TruePositive} + \sum \text{FalseNegative}}$$

- Specificity: measures a model's ability to predict a negative outcome

$$\text{specificity} = \frac{\sum \text{TrueNegative}}{\sum \text{TrueNegative} + \sum \text{FalseNegative}}$$

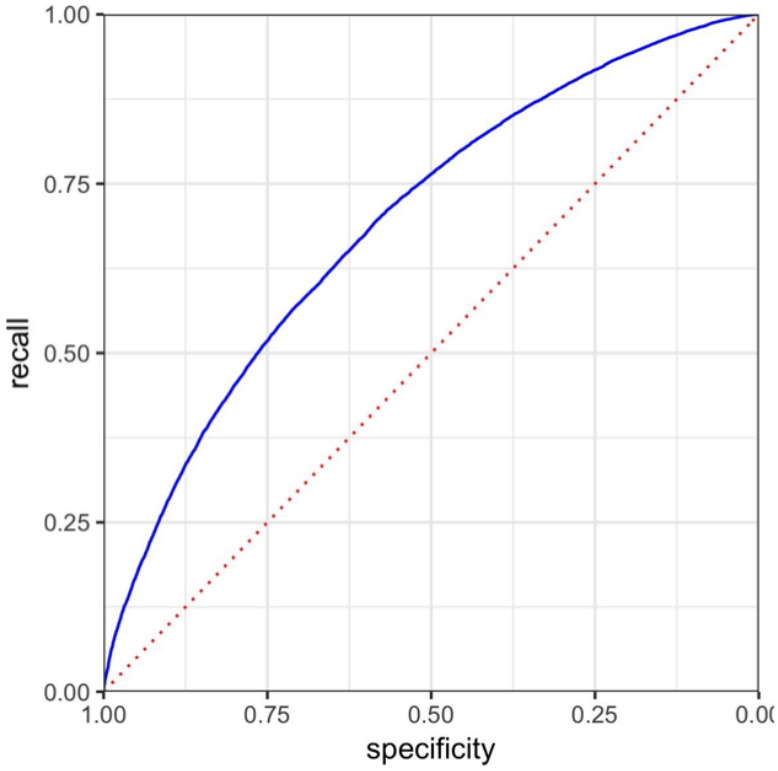
- F-measure (F1 score)=(2 × recall × precision)/(recall+precision)

Rare class problem

- Class imbalance: There is an imbalance in the classes to be predicted, with one class much more prevalent than the other in many cases
- Example: legitimate insurance claims (many) vs. fraudulent ones (small)
- The rare class (e.g., the fraudulent claims) is usually the class of more interest, and is typically designated 1, in contrast to the more prevalent 0s
- The most accurate classification model may be one that simply classifies everything as a 0
- Example: if only 0.1% of the browsers at a web store end up purchasing, a model that predicts that each browser will leave without purchasing will be 99.9% accurate (but useless)

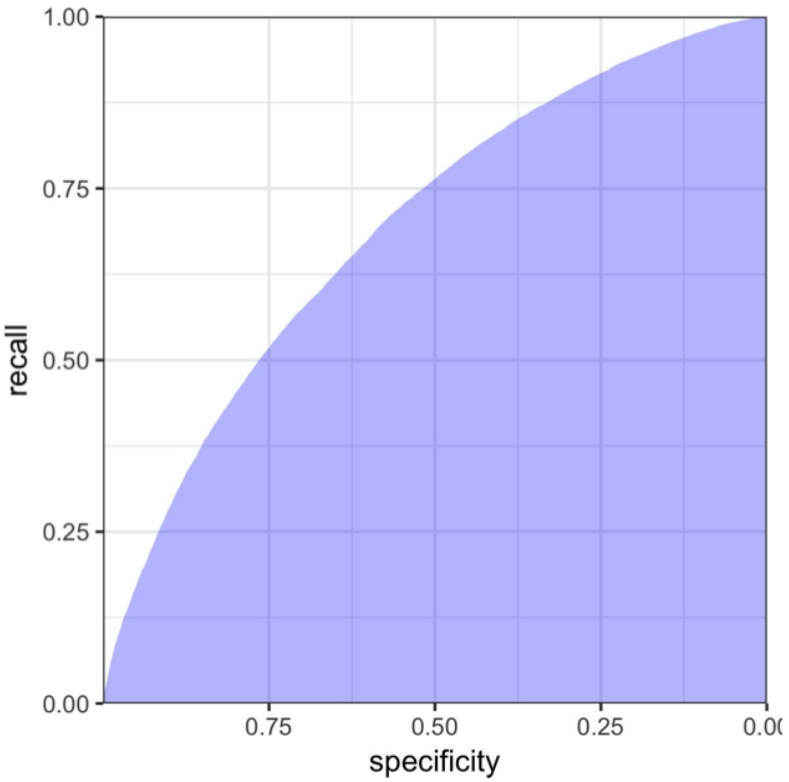
ROC curve

- Tradeoff between recall and specificity
  - Capturing more 1s generally means misclassifying more 0s as 1s
- Receiver Operating Characteristics (ROC) curve plots recall on y-axis against specificity on x-axis
- The dotted diagonal line corresponds to a classifier no better than random chance
- Effective classifier will have an ROC that hugs the upper-left corner



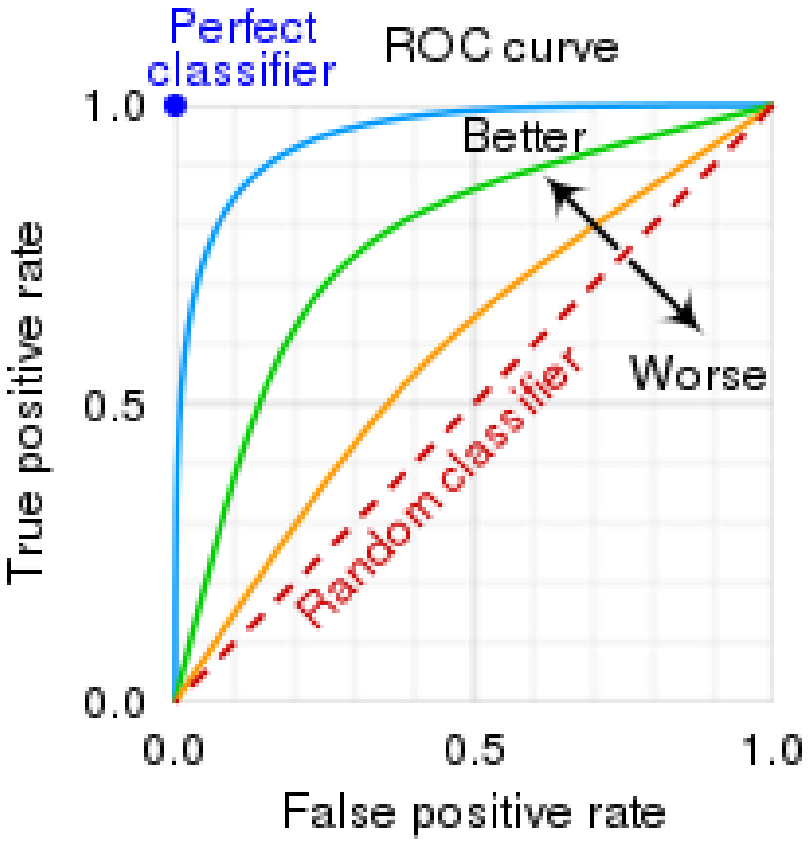
AUC

- ROC doesn't constitute a single measure for the performance of a classifier
- Area underneath the curve (AUC) is simply the total area under the ROC curve
- Larger the value of AUC, the more effective the classifier.
- An AUC of 1 indicates a perfect classifier



ROC/AUC alternative way

- ROC/AUC can be analyzed by comparing false positive rate (x-axis) and true positive rate (y-axis)
- Used in signal detection to show trade-off between hit rate and false alarm rate over noisy channel



Source: Wiki

Strategies for imbalanced data

- Strategies to improve predictive modeling performance with unbalanced data
- Under-sampling: Use fewer of the prevalent class records in the classification model
- Over-sampling: Use more of the rare class records in the classification model, bootstrapping if necessary
- Data generation: Use synthetically created records (often for minor class)

Undersampling

- Data can be balanced between 0s and 1s by downsampling the prevalent class
- Good if you have enough data
- Intuition: data for the dominant class may be redundant (with many redundant records)
- Dealing with a smaller, more balanced data set yields benefits in model performance, and makes it easier to prepare the data, and to explore and pilot models

Oversampling

- Criticism of the undersampling method – It throws away data and is not using all the information at hand
  - Critical if data size is small
- Oversampling: Upsample the rarer class by drawing additional rows with replacement (bootstrapping)
- Up (down) weighting: Attach more (less) weight to the rare (prevalent) class in the model
  - Similar effect to oversampling

Data generation

- A variation of upsampling via bootstrapping is *data generation* by perturbing existing records to create new records
- SMOTE (Synthetic Minority Oversampling Technique)
  - Creates a synthetic record that is a randomly weighted average of the original record and the neighboring record
  - Number of synthetic oversampled records created depends on the oversampling ratio required to bring the data set into approximate balance, with respect to outcome classes
- Use of neural networks: Generative adversarial network (GAN)

SMOTE example

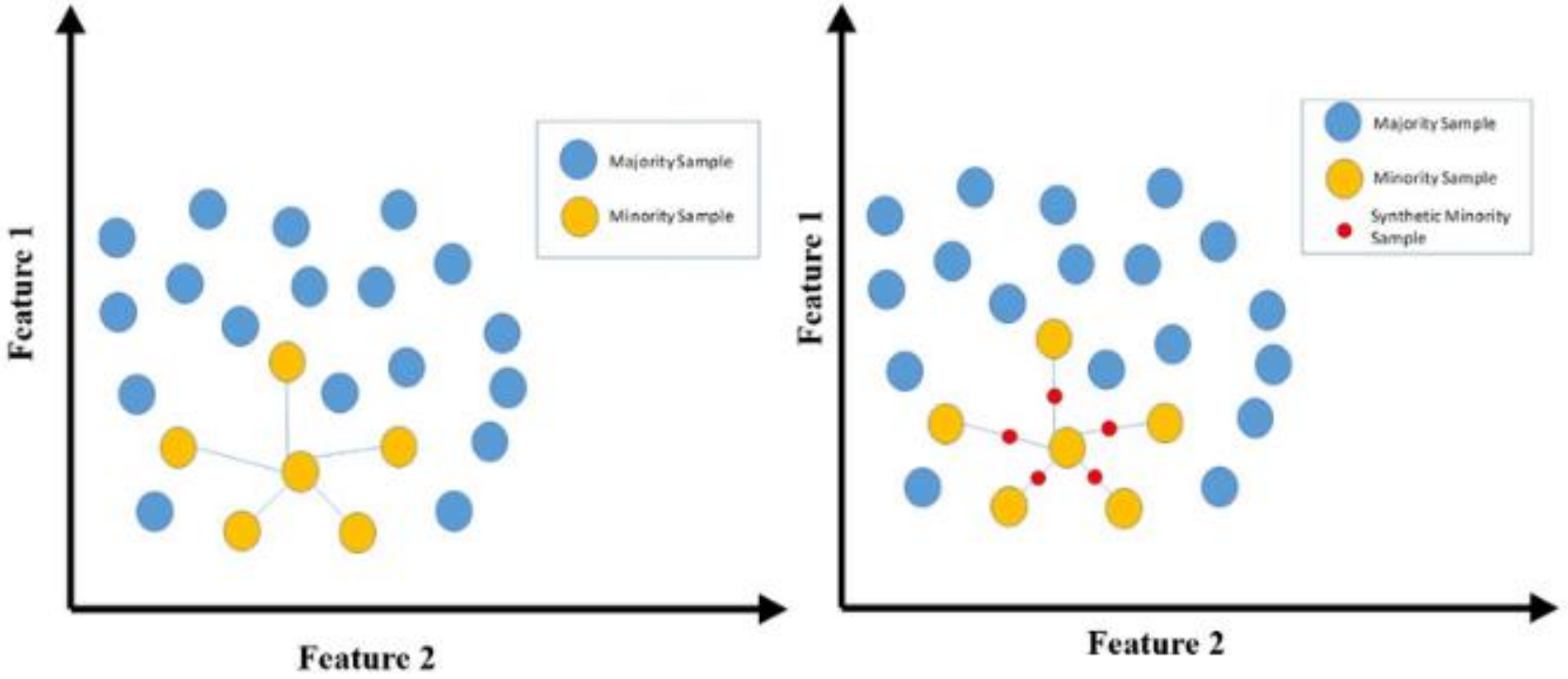


Image from researchgate.net

Summary

- Classification
- Naïve Bayes approach
- Logistic regression classifier
- Evaluating classification models
- Strategies for unbalanced data