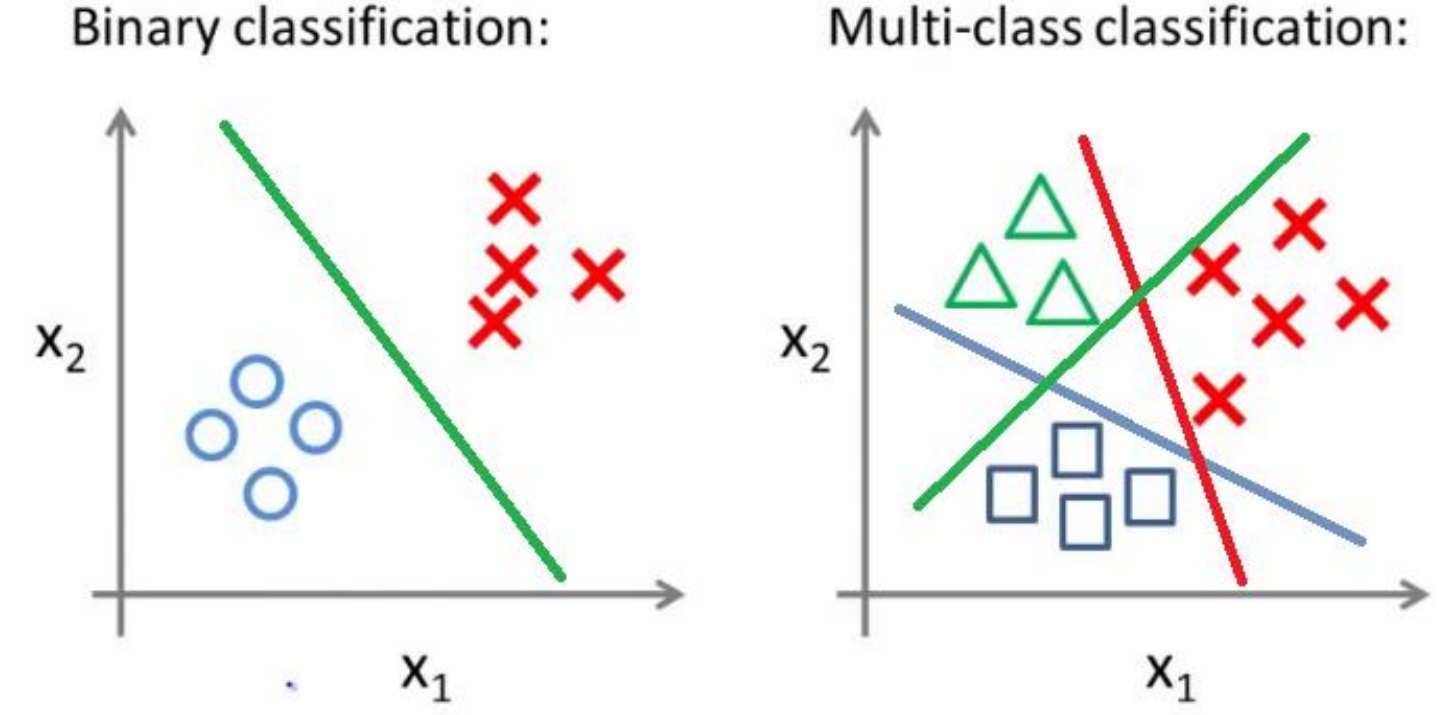
Lecture 4. Classification

Lecturer: Tetiana Marynych, marynychtetyana@gmail.com

Lecture 5 Goals:

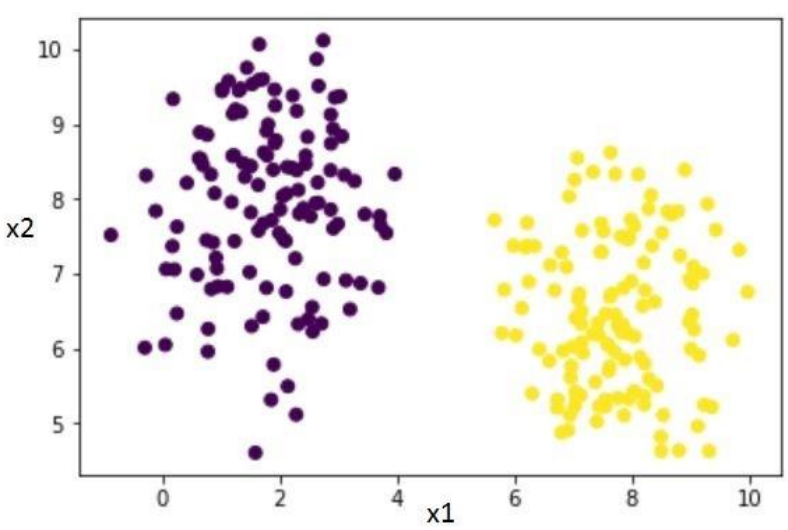
- Understand the usage of Generalized Linear Models.
- Know how to fit logistic regression and make inference.
- Know the basics of maximum likelihood method and deviance.
- Understand the principles of using confusion matrix and ROC curves in decision making.
- Practice other classification methods – Decision Tree & Random Forest.

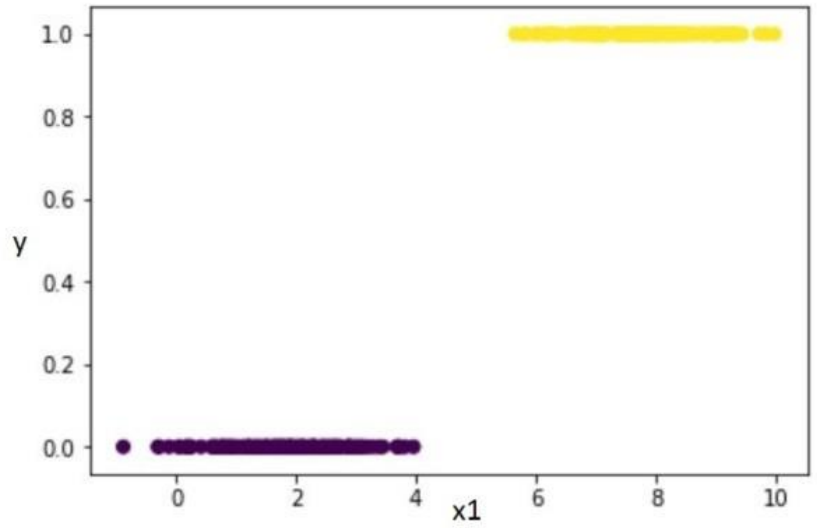
Classification: Separates the data from one to another



Source: Madhu Sanjeevi. Logistic Regression with Math. URL: medium.com

Classification

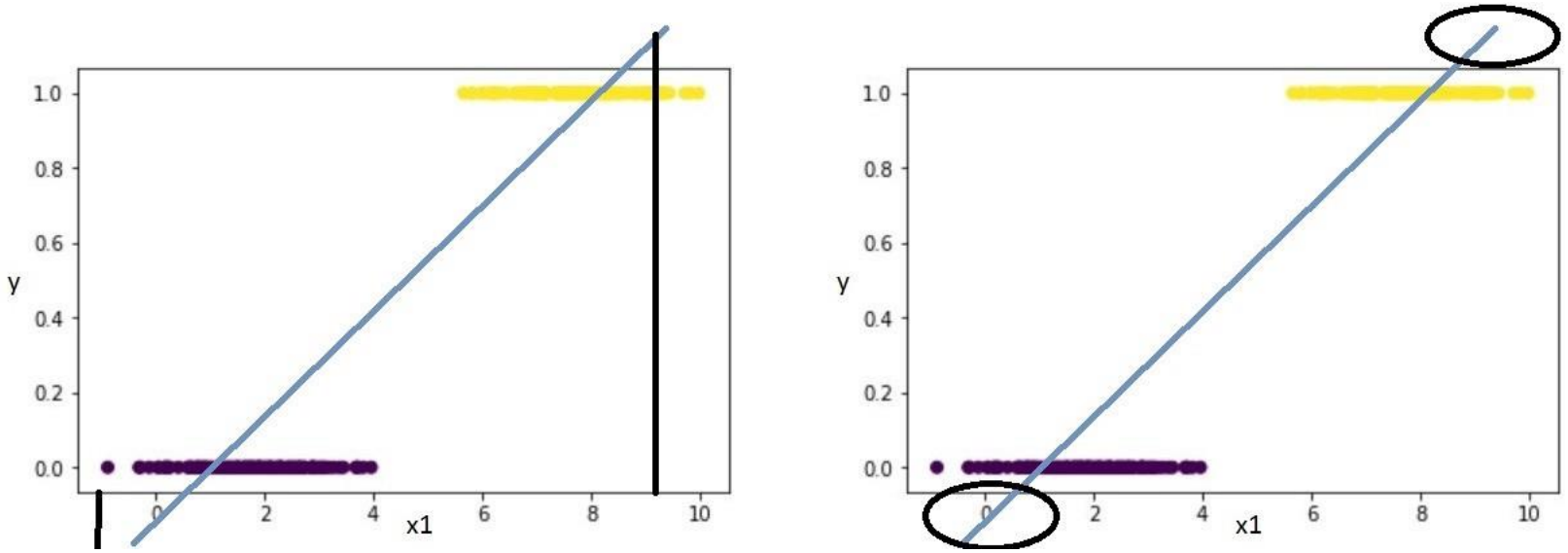




Goal is to find the straight line which separates the data at best

Given X or (Set of x values) we need to predict whether it's 0 or 1 (Yes/No).

Classification



We only accept the values between 0 and 1

We don’t accept other values to make a decision (Yes/No)

Logistic Regression: Statistical Perspective

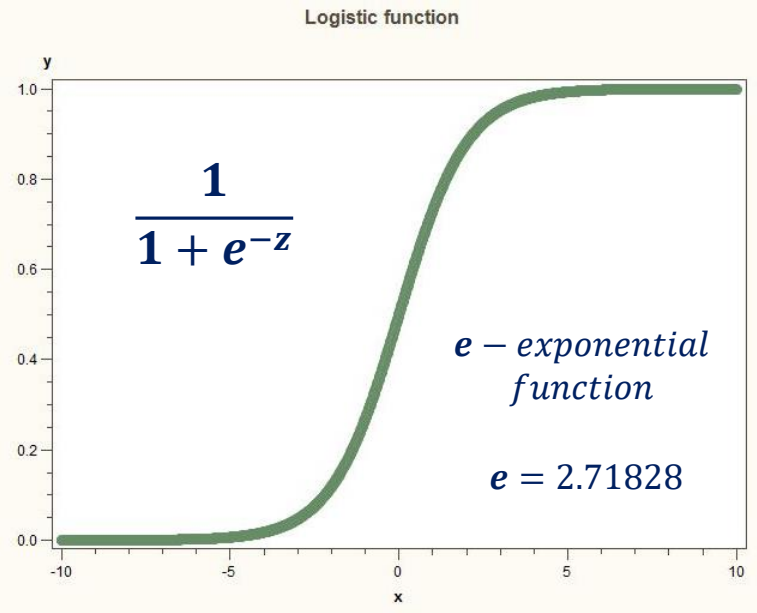
If Linear Regression predicts a continuous outcome, **Logistic Regression** models a relationship between predictor variables and a categorical response variable that has a small number of possible outcomes.

Rather than modeling this response Y directly, logistic regression models the **probability** that Y belongs to a particular category.

Types of Logistic Regression:

- Binary Logistic Regression
- Nominal Logistic Regression
- Ordinal Logistic Regression

Binary Logistic regression



We have a binary output variable Y, & we want to model a conditional probability Pr(Y=1|X=x) as $f(x)$:
$\mathrm{E}(Y_i|X_i) = 0 \cdot (1 - p_i) + 1 \cdot (p_i) = p.$
We can use ***logit transformation*** of linear function to bound p at [0,1].

The ***logistic regression model***:

$$\log\left(\frac{Pr(Y=1)}{Pr(Y=0)}\right) = \log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + x\beta$$

Solving for p, this gives $p_i = \mathrm{E}(Y_i|X_i) = \frac{e^{\beta_0+x\beta}}{1+e^{\beta_0+x\beta}} = \frac{1}{1+e^{-(\beta_0+x\beta)}}$

This means guessing $Y_i = 1$ if $\beta_0 + x\beta$ is positive, & $Y_i = 0$ otherwise.

Binary Logistic Regression

We can rewrite $\mathrm{E}(Y_i|X_i) = \frac{e^{\beta_0+x\beta}}{1+e^{\beta_0+x\beta}}$ *in terms of odds –*

assessing the likelihood of a particular event

$$\log(Odds) = \log(\frac{P(Y=1)}{P(Y=0)}) = \log\left(\frac{p_i}{1-p_i}\right) = \frac{e^{X\beta}/(1+e^{X\beta})}{(1+e^{X\beta})/(1+e^{X\beta}) - e^{X\beta}/(1+e^{X\beta})} = e^{X\beta}$$

Log-odds or "*logits*" are used to build the probability within the required limits - between 0 and 1.

Odds ratio can be any positive integer:

- Odds = 1 indicates no relationship between response and predictors
- Odds> 1 if y = 1 is more probable
- Odds <1 if y = 0 is more probable

Binary Logistic Regression

In particular, Odds (odds ratios) increase multiplicatively by $exp(\beta i)$ with each increase in *Xi* per unit (when the other variables remain fixed).

We use inverse transformation to express logistic regression coefficients through log (odds):

$$\log Odds(P(Y = 1) = X\beta$$

$$Odds(P(Y = 1) = \exp(X\beta) = \omega$$

$$p_i = \frac{Odds}{1 + Odds} = \frac{\omega}{1 + \omega} \qquad \omega = \frac{p_i}{1 - p_i}$$

Logistic regression: Loss function

The loss function for linear regression is squared loss.
The loss function for logistic regression is Log Loss:

$$Log\ Loss = \sum -ylog(y') - (1-y)\log(1-y')$$

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n} Loss(h_\theta(x^{(i)}), y^{(i)})$$

$$Loss(h_\theta(x), y) = \begin{Bmatrix} -log(h_\theta(x))\ if\ y = 1 \\ -log(1-h_\theta(x))\ if\ y = 0 \end{Bmatrix}$$

We can further apply **Gradient descent** optimization algorithm to find the ϑ parameters values.

Logistic regression: Maximum Likelihood estimation

The Probability Statistical method – **Maximum Likelihood** estimation assesses the parameters ϑ *(unknown coefficients* β*)* that max the probability of getting *X* data.

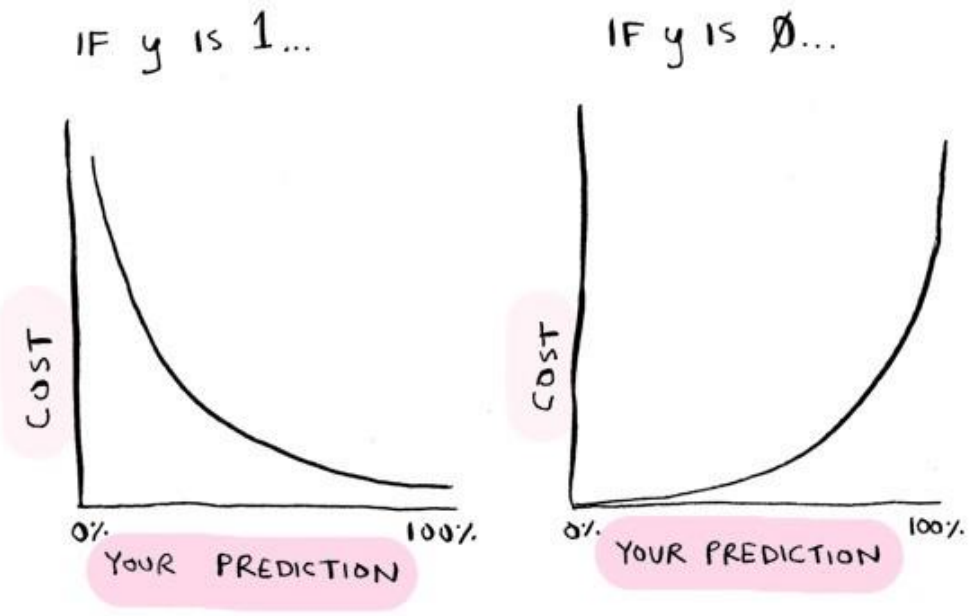
$L(Y, \theta) \rightarrow max_{\theta \in \Theta}$ Θ – the space of parameters

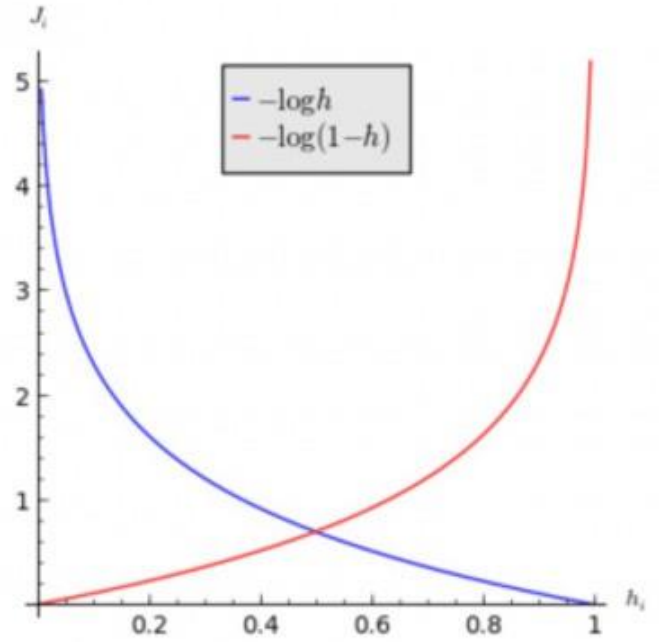
$$L(Y, \theta) = \prod_{i=1}^{n} f(Y_i, \theta)$$

$$L(\beta, Y, X) = \prod_{i=1}^{n} p_i^{y_i}(1 - p_i)^{1-y_i} = \prod_{i=1}^{n} \left(\frac{\exp(X_i\beta)}{1+\exp(X_i\beta)}\right)^{y_i} \left(\frac{1}{1+\exp(X_i\beta)}\right)^{1-y_i}$$

The maximum and minimum of functions are determined by differentiation. The procedure consists in finding the derivative function, equating it to zero and finding the desired parameter through the equation.

Logistic regression: Loss function





If y =1 and y' =0 the cost goes to ∞. If y =1 and y'=1 the cost goes to minimum.

If y =0 and y' =1 the cost goes to ∞. If y =0 and y'=0 the cost goes to minimum.

If we apply log to hypothesis (predicted) we can estimate the overall error.

Logistic Regression: Thresholding

- Logistic regression returns a **probability**.
- You can convert the returned probability to a **binary value.**
- Define a **classification threshold** (also called the **decision threshold**).
- Classification prediction models are summarized by **confusion matrix**:
  - a **true positive** – an outcome where the model *correctly* predicts the *positive* class (y=1).
  - a **true negative** – an outcome where the model *correctly* predicts the *negative* class (y=0).
  - a **false positive** – an outcome where the model *incorrectly* predicts the *positive* class.
  - a **false negative** – an outcome where the model *incorrectly* predicts the *negative* class.

Logistic Regression: Thresholding

Threshold (Cut-Off) Value *t* :

- If $P(Spam = 1) \geq t$, predict spam;
- If $P(Spam = 1) < t$, predict not a spam.

The choice of the threshold ***t*** depends on the type of error which is acceptable :

- If t is big, predict spam rarely – more errors FN.
- If t is small, predict non-spam rarely – more errors FP.
- Without any advantage between mistakes, choose t = 0,5 (applies only for balanced data)

Classification Accuracy

Confusion matrix: Type I and II Errors

	Actually	
Predicted	Yes ($p_i = 1$) (Positive)	No ($p_i = 0$) (Negative)
Yes ($\hat{p}_i = 1$)	TP	FP
No ($\hat{p}_i = 0$)	FN	TN

CORRECT	Type II Error
Type I Error	CORRECT

Classification Accuracy

Confusion matrix: I and II type errors

In general, Positive = identified and negative = rejected:

- True Positive (**TP**) = correctly identified (we predict a spam and it is actually a spam)
- False Positive (**FP**) = incorrectly identified – II type Error (we predict a spam when it is not)
- True negative (**TN**) = correctly rejected (we predict not a spam and it isn't)
- False negative (**FN**) = incorrectly rejected – I type Error (we predict not a spam, when it is a spam)

Classification Accuracy: Confusion Matrix Analysis

- $\text{Accuracy} = \frac{TN+TP}{n}$
- $Error = \frac{FN+FP}{n}$
- ***Sensitivity*** or ***Recall*** – True Positive Rate:

$$TPR = \frac{TP}{Actual\ Results} = \frac{TP}{TP+FN} \cdot 100\%$$

- ***Precision***:

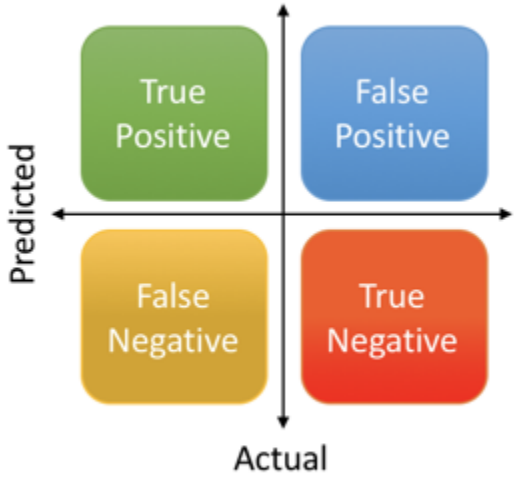
$$Precision\ = \frac{TP}{TP+FP} \cdot 100\%$$

Confusion Matrix Analysis

$$\textbf{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

$$\textbf{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

$$\textbf{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{Total}}$$



Source: Shruti Saxena. Precision vs. Recall. URL: https://towardsdatascience.com/

Confusion Matrix Analysis

- **True Negative Rate** – **Specificity** – the proportion of true-negative samples that have been correctly classified:

$$TNR = \frac{TN}{TN+FP} \cdot 100\%$$

- **False Positive Rate**

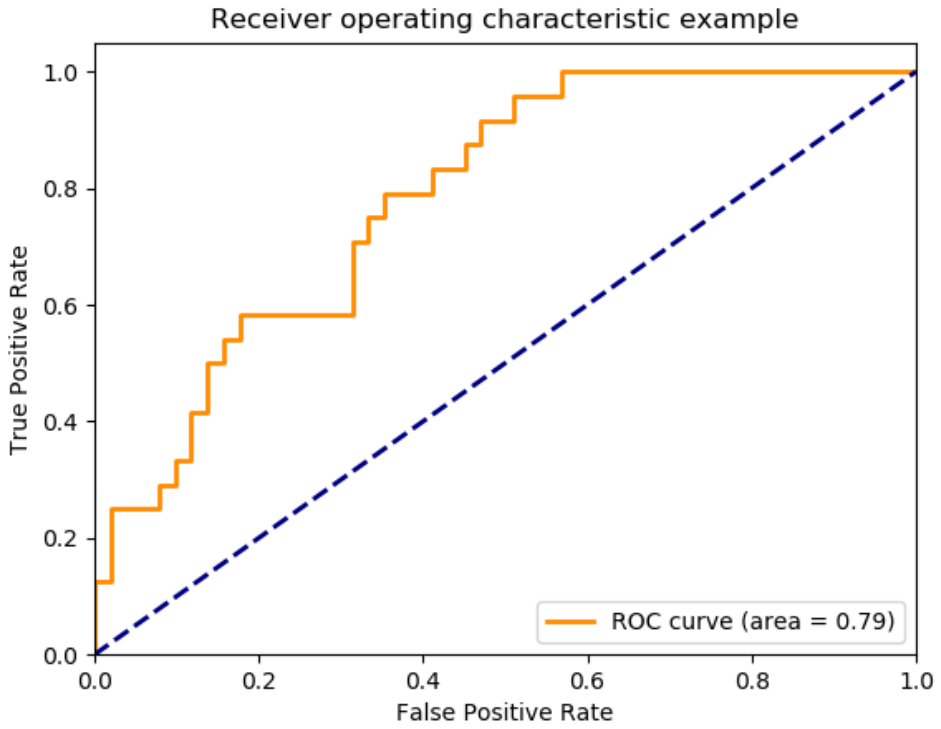
$$FPR = \frac{FP}{TN+FP} \cdot 100\%$$

- **False Negative Rate**

$$FNR = \frac{FN}{TP+FN} \cdot 100\%$$

Classification Accuracy: ROC – AUC Analysis

A **ROC curve (Receiver Operator Characteristic)** is used to evaluate the quality of the binary classification prediction.



AUC – area under the curve

ROC-AUC Interpretation

The classification quality is determined by the area under the ROC curve, which is denoted as AUC (area under curve).

The higher the AUC, the higher the model's predictive value.

The following quality characteristics of the models are used:

- 0.9 – 1.0 — excellent
- 0.8 – 0.9 — very good
- 0.7 – 0.8 — good
- 0.6 – 0.7 — satisfactory
- 0.5 – 0.6 — bad