Logistic Regression

Pritam Prakash Shete

Computer Division, BARC

Centre for Excellence in Basic Sciences

Topics

- Supervised learning – Binary classification
- Sigmoid function
- Logistic regression
- Gradient descent
- Regularization
- Confusion matrix
- Applications
- Advantages
- Disadvantages

Supervised Learning

- Training set – { $(x^{(1)},y^{(1)})$, $(x^{(2)},y^{(2)})$, …, $(x^{(m)},y^{(m)})$}
- Labeled dataset

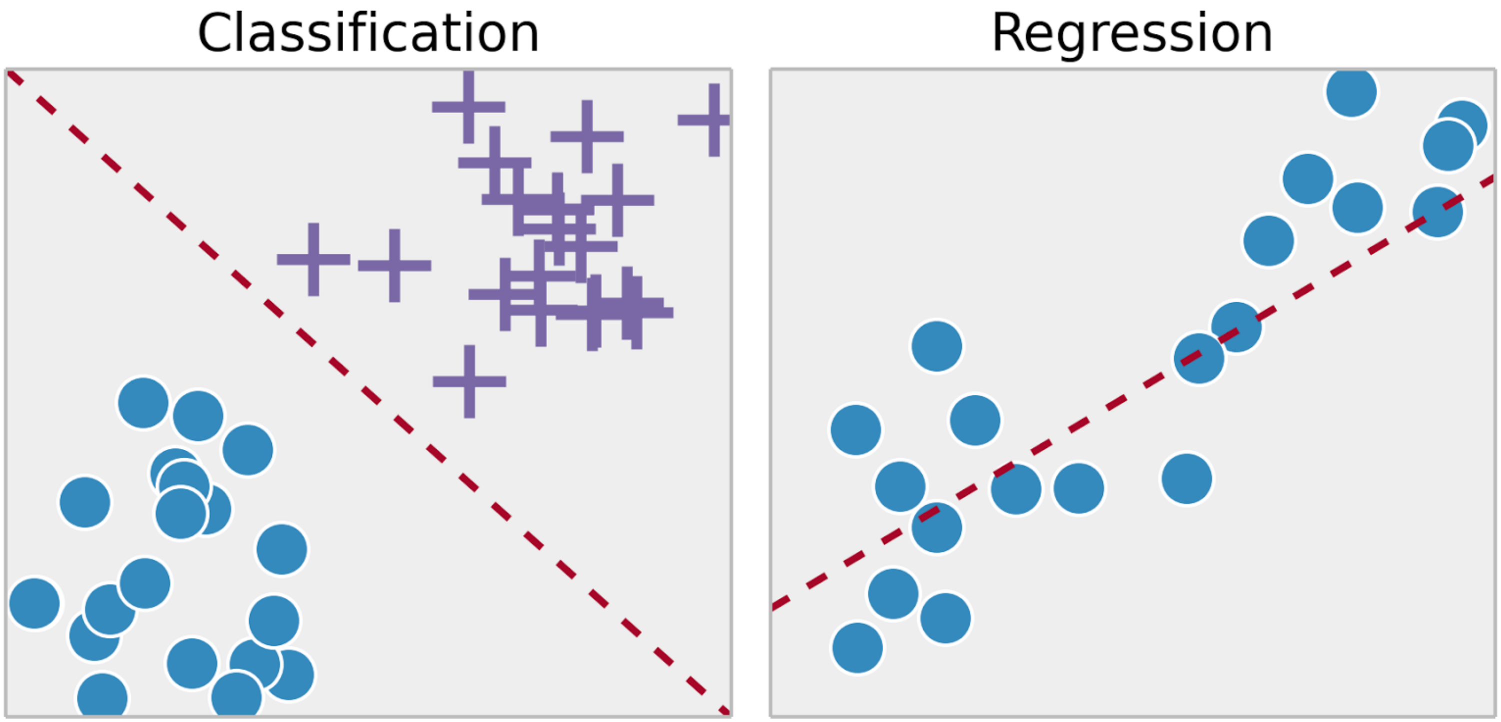


Image ack – Google

Binary Classification

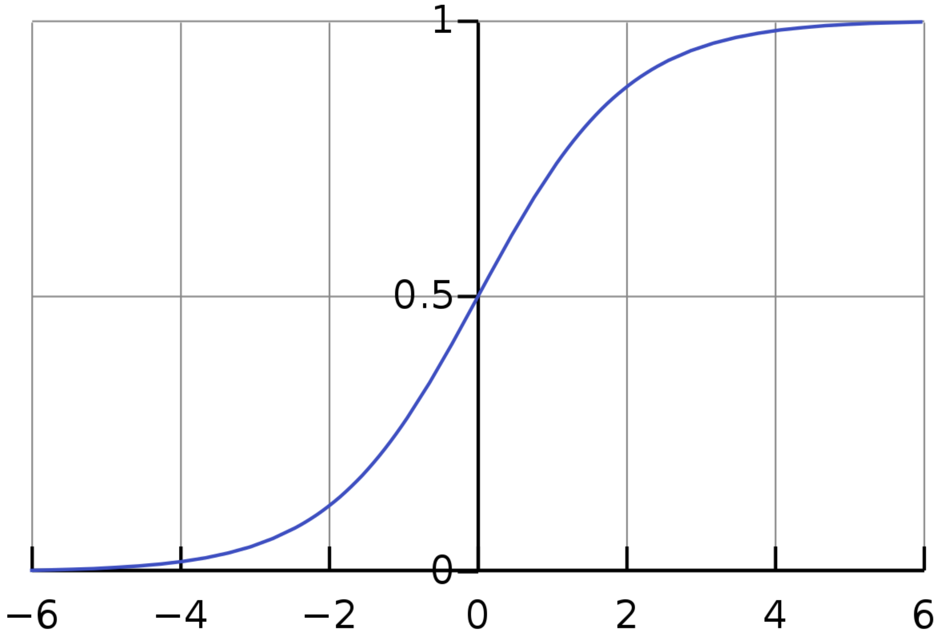
- Classify elements of given set into two groups
  - Classify dog and non-dog images

Dog

Non-dog

Image ack – Google

Sigmoid Function



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

Sigmoid Function

- $\sigma(Z) \sim 1$ – For $Z >> 0$
- $\sigma(Z) \sim 0$ – For $Z << 0$
- $\sigma(Z) = 0.5$ – For $Z = 0$

Z	**σ(Z)**
-2	0.12
-1.5	0.18
-1	0.27
-0.5	0.38
0	0.50
0.5	0.62
1	0.73
1.5	0.82
2	0.88

Logistic Regression

- Input – X
  - Vector
  - X ε R
  - Dimension – n_X
- Output – ŷ
  - Scalar
  - $0 <= \hat{y} <= 1.0$

Linear Regression

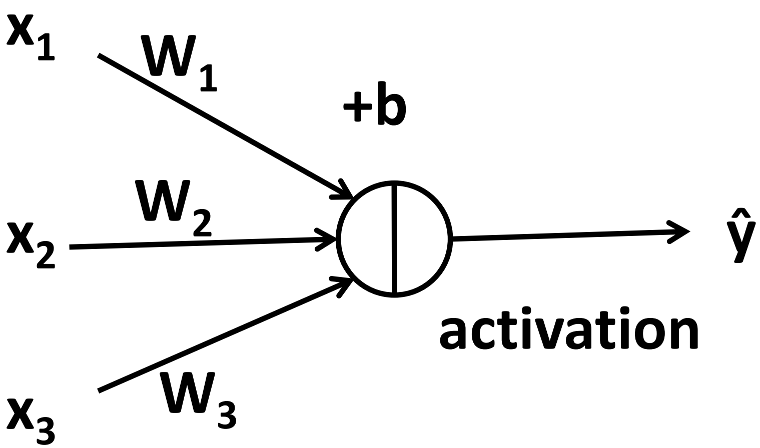
- Weights – W
  - Vector
  - W ε R
  - Dimension – n_X
- Bias – b
  - Scalar
  - b ε R
- $y = W^T X + b$

Logistic Regression

- Weights – W
  - Vector
  - W ε R
  - Dimension – n_X
- Bias – b
  - Scalar
  - b ε R
- $\cancel{y = W^T X + b}$

Logistic Regression

- Weights – W
  - Vector
  - W ε R
  - Dimension – n_X
- Bias – b
  - Scalar
  - b ε R
- $Z = W^T X + b$
- $\hat{y} = \sigma(Z)$ – Activation (sigmoid) function



Logistic Regression

- Weights – W
  - Vector
  - W ε R
  - Dimension – n_X
- Bias – b
  - Scalar
  - b ε R
- $Z = W^T X + b$
- $\hat{y} = P(\ y=1\ |\ X)$

Loss Function

- Input dataset
  - $\{ (x^{(1)},y^{(1)}), (x^{(2)},y^{(2)}), \ldots, (x^{(m)},y^{(m)})\}$
- Equations
  - $Z = W^T X + b$
  - $\hat{y} = \sigma(Z) = y_p$
- Output
  - $\hat{y}^{(i)} \sim y^{(i)}$
  - W and b

Loss Function

- $L(\hat{y}, y) = -y * \log(\hat{y}) - (1 - y) * \log(1 - \hat{y})$

Loss Function

- $L(\hat{y}, y) = -y * \log(\hat{y}) - (1 - y) * \log(1 - \hat{y})$

  $y = 1$

  $L(\hat{y}, y) = -\log(\hat{y})$

  $-\log(\hat{y})$ – Minimize

  $\log(\hat{y})$ – Maximize

  $\hat{y}$ – Maximize

  $\hat{y}$ – 1.0

Loss Function

- $L(\hat{y}, y) = -y * \log(\hat{y}) - (1 - y) * \log(1 - \hat{y})$

  $y = 0$

  $L(\hat{y}, y) = -\log(1 - \hat{y})$

  $-\log(1 - \hat{y})$ – Minimize

  $\log(1 - \hat{y})$ – Maximize

  $1 - \hat{y}$ – Maximize

  $\hat{y}$ – Minimize

  $\hat{y}$ – 0.0

Loss Function

- Loss function
  - One sample – i^{th} sample
  - $L(\hat{y}^{(i)}, y^{(i)}) = -y^{(i)} * \log(\hat{y}^{(i)}) - (1 - y^{(i)}) * \log(1 - \hat{y}^{(i)})$
- Cost function
  - Average of loss function for all samples

$$J(W,b) = \frac{1}{\mathrm{m}} \sum_{i=1}^{m} L(y^{(i)}, {y_p}^{(i)})$$

$$J(W,b) = -\frac{1}{\mathrm{m}} \sum_{i=1}^{m} \left[y^{(i)} * \log({y_p}^{(i)}) + (1 - y^{(i)}) * \log(1 - {y_p}^{(i)}) \right]$$

Gradient Descent

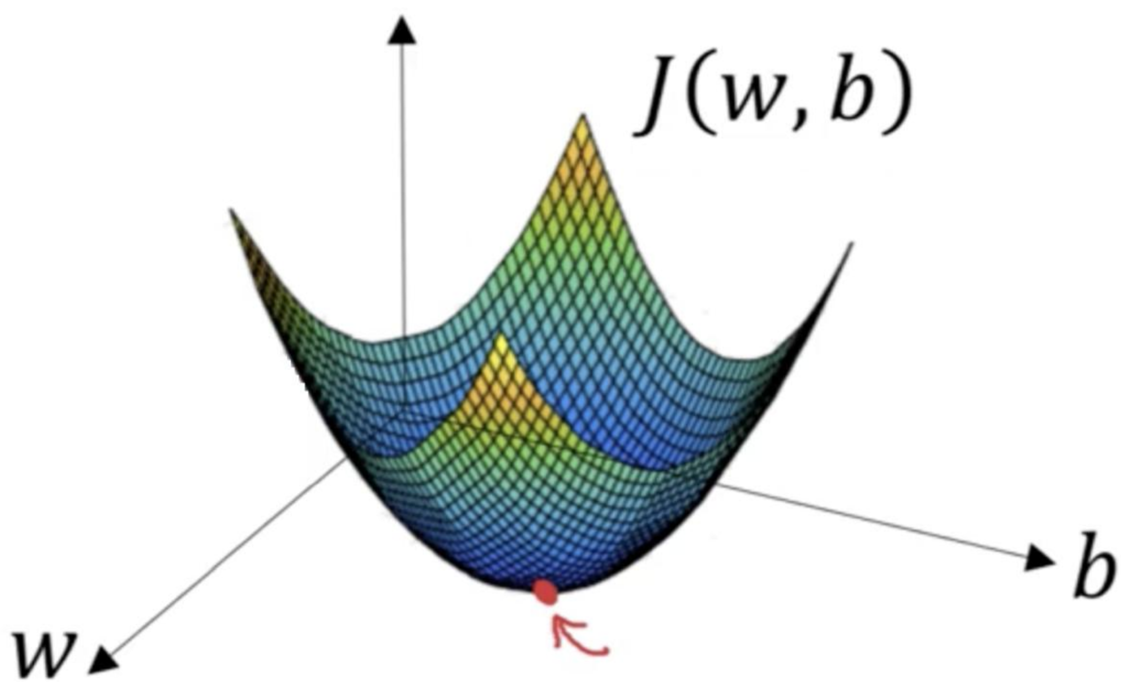
- Input dataset – $\{ (x^{(1)},y^{(1)}), (x^{(2)},y^{(2)}), \ldots, (x^{(m)},y^{(m)}) \}$
- Equations – $Z = W^T X + b$ and $\hat{y} = \sigma(Z) = y_p$
- Loss function – $L(\hat{y}, y) = -y * \log(\hat{y}) - (1-y) * \log(1-\hat{y})$
- Cost function – $J(W,b) = \frac{1}{m}\sum_{i=1}^{m} L(y^{(i)}, y_p{}^{(i)})$

$$J(W,b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} * \log(y_p{}^{(i)}) + (1-y^{(i)}) * \log(1-y_p{}^{(i)})\right]$$

- Output
  - $\hat{y}^{(i)} \sim y^{(i)}$
  - W and b – Minimize J(W, b)

Gradient Descent

- Convex function
- Global optimum



Gradient Descent

- Forward pass
  - $Z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$
  - $\hat{y} = a^{[1]} = \sigma(Z^{[1]})$
  - $L(\hat{y}^{(i)}, y^{(i)})$
  - $J(W^{[1]}, b^{[1]})$
- Backward pass

$$dW^{[1]} = \frac{\partial J}{\partial W^{[1]}}$$

$$db^{[1]} = \frac{\partial J}{\partial b^{[1]}}$$

Gradient Descent

- Forward pass
  - $Z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$
  - $\hat{y} = a^{[1]} = \sigma(Z^{[1]})$
  - $L(\hat{y}^{(i)}, y^{(i)})$
  - $J(W^{[1]}, b^{[1]})$

- Backward pass

$$dW^{[1]} = \frac{\partial J}{\partial W^{[1]}}$$

$$db^{[1]} = \frac{\partial J}{\partial b^{[1]}}$$

$$W^{[1]} = W^{[1]} - \alpha * dW^{[1]}$$

$$b^{[1]} = b^{[1]} - \alpha * db^{[1]}$$

Gradient Descent

- Forward pass
  - $Z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$
  - $\hat{y} = a^{[1]} = \sigma(Z^{[1]})$
  - $L(\hat{y}^{(i)}, y^{(i)})$
  - $J(W^{[1]}, b^{[1]})$

- Backward pass

$$da^{[1]} = \frac{\partial L}{\partial a^{[1]}}$$

$$da^{[1]} = \frac{-y}{a} + \frac{1-y}{1-a}$$

Gradient Descent

- Forward pass
  - $Z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$
  - $\hat{y} = a^{[1]} = \sigma(Z^{[1]})$
  - $L(\hat{y}^{(i)}, y^{(i)})$
  - $J(W^{[1]}, b^{[1]})$
- Backward pass

$$da^{[1]} = \frac{-y}{a} + \frac{1-y}{1-a}$$

Gradient Descent

- Forward pass
  - $Z^{[1]} = W^{[1]} a^{[0]} + b^{[1]}$
  - $\hat{y} = a^{[1]} = \sigma(Z^{[1]})$
  - $L(\hat{y}^{(i)}, y^{(i)})$
  - $J(W^{[1]}, b^{[1]})$

- Backward pass

$$da^{[1]} = \frac{-y}{a} + \frac{1-y}{1-a}$$

$$dZ^{[1]} = \frac{\partial L}{\partial Z^{[1]}} = \frac{\partial L}{\partial a^{[1]}} \frac{\partial a^{[1]}}{\partial Z^{[1]}}$$

Gradient Descent

- Forward pass
  - $Z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$
  - $\hat{y} = a^{[1]} = \sigma(Z^{[1]})$
  - $L(\hat{y}^{(i)}, y^{(i)})$
  - $J(W^{[1]}, b^{[1]})$
- Backward pass

$$da^{[1]} = \frac{-y}{a} + \frac{1-y}{1-a}$$

$$dZ^{[1]} = \left(\frac{-y}{a} + \frac{1-y}{1-a}\right)\left(a\left(1-a\right)\right)$$

Gradient Descent

- Forward pass
  - $Z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$
  - $\hat{y} = a^{[1]} = \sigma(Z^{[1]})$
  - $L(\hat{y}^{(i)}, y^{(i)})$
  - $J(W^{[1]}, b^{[1]})$
- Backward pass

$$da^{[1]} = \frac{-y}{a} + \frac{1-y}{1-a}$$

$$dZ^{[1]} = a - y$$

Gradient Descent

- Forward pass
  - $Z^{[1]} = W^{[1]} a^{[0]} + b^{[1]}$
  - $\hat{y} = a^{[1]} = \sigma(Z^{[1]})$
  - $L(\hat{y}^{(i)}, y^{(i)})$
  - $J(W^{[1]}, b^{[1]})$
- Backward pass

$$da^{[1]} = \frac{-y}{a} + \frac{1-y}{1-a}$$

$$dZ^{[1]} = a - y$$

$$dW^{[1]} = \frac{\partial L}{\partial W^{[1]}} = \frac{\partial L}{\partial a^{[1]}} \frac{\partial a^{[1]}}{\partial Z^{[1]}} \frac{\partial Z^{[1]}}{\partial W^{[1]}}$$

Gradient Descent

- Forward pass
  - $Z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$
  - $\hat{y} = a^{[1]} = \sigma(Z^{[1]})$
  - $L(\hat{y}^{(i)}, y^{(i)})$
  - $J(W^{[1]}, b^{[1]})$
- Backward pass

$$da^{[1]} = \frac{-y}{a} + \frac{1-y}{1-a}$$

$$dZ^{[1]} = a - y$$

$$dW^{[1]} = a(a - y)$$

Gradient Descent

- Forward pass
  - $Z^{[1]} = W^{[1]} a^{[0]} + b^{[1]}$
  - $\hat{y} = a^{[1]} = \sigma(Z^{[1]})$
  - $L(\hat{y}^{(i)}, y^{(i)})$
  - $J(W^{[1]}, b^{[1]})$

- Backward pass

$$da^{[1]} = \frac{-y}{a} + \frac{1-y}{1-a}$$

$$dZ^{[1]} = a - y$$

$$dW^{[1]} = a(a - y)$$

$$db^{[1]} = \frac{\partial L}{\partial b^{[1]}} = \frac{\partial L}{\partial a^{[1]}} \frac{\partial a^{[1]}}{\partial Z^{[1]}} \frac{\partial Z^{[1]}}{\partial b^{[1]}}$$

Gradient Descent

- Forward pass
  - $Z^{[1]} = W^{[1]} a^{[0]} + b^{[1]}$
  - $\hat{y} = a^{[1]} = \sigma(Z^{[1]})$
  - $L(\hat{y}^{(i)}, y^{(i)})$
  - $J(W^{[1]}, b^{[1]})$

- Backward pass

$$da^{[1]} = \frac{-y}{a} + \frac{1-y}{1-a}$$

$$dZ^{[1]} = a - y$$

$$dW^{[1]} = a(a - y)$$

$$db^{[1]} = a - y$$

Regularization

- Lasso regression
- Ridge Regression
- Elastic Net Regression

Lasso Regression

- Least Absolute Shrinkage Selector Operator
- L1 regularization technique
- Reduce coefficients
- Feature selection
  - Select important features
  - Reduce coefficients of others to zero
- Suitable for more number of features

Ridge Regression

- L2 regularization technique
- Reduce coefficients
- Reduce model complexity
- Prevent multicollinearity

Elastic Net Regression

- L1 and L2 regularization technique

Confusion Matrix

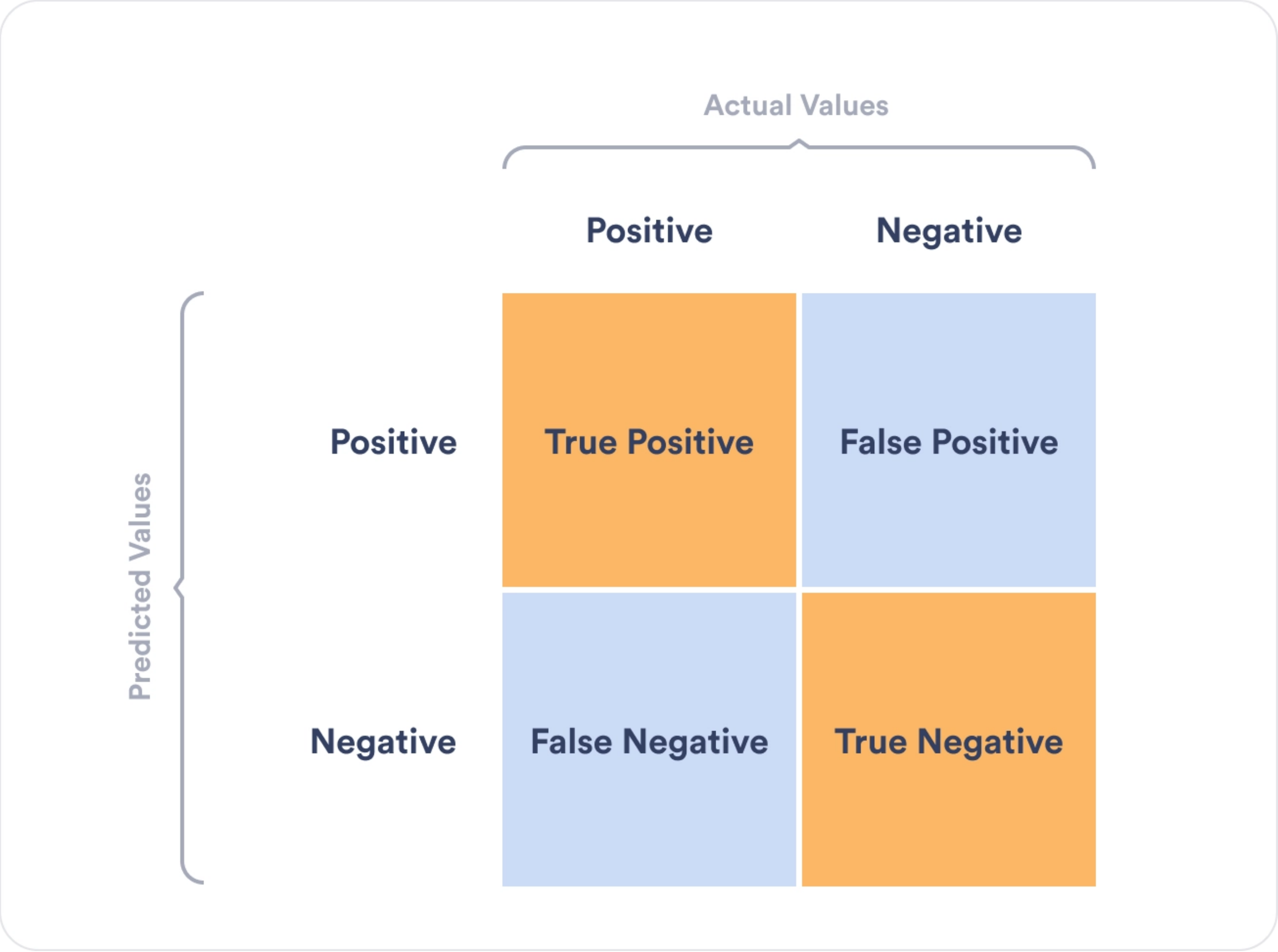


Image ack – Google

Confusion Matrix

- True positive
  - Actual positive
  - Predicted positive
- False positive – Type 1 error
  - Actual negative
  - Predicted positive
- False negative – Type 2 error
  - Actual positive
  - Predicted negative
- True negative
  - Actual negative
  - Predicted negative

Confusion Matrix

- Accuracy
  - (TP + TN) /(TP + TN + FP + FN)
- Recall
  - (TP) / (TP + FN)
- Precision
  - (TP) / (TP + FP)
- F1 score
  - (2 × Recall × Precision) / (Recall + Precision)

Applications

- Binary classification
- Positive class and negative class

Advantages

- Easy to implement and interpret
- Efficient to train
- No assumptions about distributions of classes
- Can be extended to multiple classes
- Good accuracy for linearly separable dataset

Disadvantages

- Overfit for small dataset
- Construct linear boundaries
- Assumption of linearity
  - Independent variables
  - Dependent variables
- Cannot solve non linear problems

Questions?

Thank you