01/12/25

problem statement:

US company (insurance) predicting based on characteristics the claimant is going to hire an attorney or not.

confusion matrix:

$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 315 \rightarrow TN & 151 \rightarrow FP \\ 1 & 100 \; FN & 310 \; TP \end{array}$$

$$\text{Accuracy} = \frac{315+310}{315+310+100+151}$$

$$= \frac{625}{876}$$

Accuracy $= 0.7134$

precision 0 : $\frac{315}{315+100} = 0.7590$

≈ 0.7157

precision 1 : $\frac{310}{310+151} = \frac{310}{461} = 0.6724$

Recall 0 : $\frac{315}{315+151} = 0.6759$

~~F1 score = 2 ×~~

$= 0.71598$

$= \cancel{2.268}$

Recall 1 : $\frac{310}{100+310} = 0.7560$

F1 score = 0.785

* to check class balance
column
dataset. value counts()

* For balanced classes go with accuracy. for imbalanced classes the accuracy will show high; hence we need to go with F1 score.

* ROC - AUC curve: talks about seperability.

Loss function : talks about how bad the model is.

Gradient descent : talks about how bad the model is.

Evaluation Metrics for Iris dataset :

Confusion matrix:

$$\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 38 & 0 & 0 \\ 1 & 0 & 39 & 2 \\ 2 & 0 & 0 & 41 \end{array}$$

$$\text{Accuracy} = \frac{38+39+41}{38+39+2+41} = \frac{118}{120} = 0.983$$

$$\text{precision}_0 = \frac{38}{38+0+0} = 1$$

$$\text{precision}_1 = \frac{39}{0+39+0} = 1$$

$$\text{precision}_2 = \frac{41}{0+41+2} = 0.953$$

Avg. precision = 0.9844

$$\text{Recall}_0 = \frac{38}{38} = 1$$

$$\text{Recall}_1 = \frac{39}{41} = 0.951$$

$$\text{Recall}_2 = \frac{41}{41} = 1$$

$$\text{Avg. Recall} = \frac{1+0.951+1}{3} = 0.983$$

$$F1\ score = \frac{2(0.9844)(0.983)}{0.9844 + 0.983}$$

$$F1score = 0.9836$$

How is logistic regression classifying for multi class:

* Working:

- If we have n classes, the logistic regression trains n-seperate binary logistic models.

- For each model:

  it treats one class as positive & all others as negative

- During prediction:

  Each model outputs a probability, the class with highest prob is choosen.

probability output:

predict_proba() returns a matrix of shape (k_samples, n)

Loss function:

A loss function / cost function is a mathematical function that measures how wrong the model's predictions are compared to the actual values.

It converts prediction errors into a single numeric score.

It is used in binary classification as binary cross entropy.

For binary classifier:

$$Loss = -[y \log(\hat{y}) + (1-y)\log(1-\hat{y})]$$

Used in supervised learning.

for k classes

$$Loss = -\sum_{i=1}^{k} y_i \log(\hat{y}_i)$$

Gradient Descent:

It is an optimization algorithm used to minimize the loss function by updating the model parameters (weights) step by step.

It finds best weights that gives the lowest loss.

$$\theta = \theta - \alpha \frac{\partial Loss}{\partial \theta}$$

θ - model parameters (weights)

α - learning rate (step size)

$\partial Loss / \partial \theta$ - gradient (slope)

Types:

1. Batch GD : Uses all training data for each update. (slow but stable)
2. stochastic GD : Updates after each sample (fast but noisy)
3. Mini-batch GD : Uses small batches - most commonly used.

Softmax Function:

for each class k:

$$\hat{y}_{i,k} = \frac{e^{z_{i,k}}}{\sum_{j=1}^{k} e^{z_{i,j}}}$$

where, $z_{i,k} = w_k^T x_i + b_k$

k : no. of classes

$\hat{y}_{i,k}$ = prob that sample i belong to class k.

Binary logistic Regression Vs Multinomial Logistic Regression

Binary	Multinomial
Sigmoid	softmax
Binary cross entropy	Multi class entropy
$-y \log \hat{y} - (1-y) \log(1-\hat{y})$	$-\Sigma y_k \log(\hat{y}_k)$
single probability	prob distribution over k classes