

1. Logistic Regression thuộc nhánh nào trong Machine Learning?
 - ☒ (a) Supervised Learning.
 - (b) Unsupervised Learning.
 - (c) Reinforcement Learning.
 - (d) Self-supervised Learning.
2. Logistic Regression thường được áp dụng để giải quyết vấn đề nào dưới đây?
 - ☒ (a) Phân lớp nhị phân.
 - (b) Dự đoán chuỗi thời gian.
 - (c) Giảm chiều dữ liệu.
 - (d) Phân cụm dữ liệu.
3. Bài toán nào sau đây có thể được giải quyết hiệu quả bằng Logistic Regression?
 - (a) Dự đoán giá nhà.
 - ☒ (b) Phân loại email spam.
 - (c) Đề xuất phim.
 - (d) Dự đoán giá cổ phiếu.
4. Hàm Cross-Entropy được chọn làm hàm loss trong Logistic Regression vì lý do gì?
 - (a) Nó giúp quá trình huấn luyện nhanh hơn.
 - (b) Nó giữ giá trị dự đoán trong khoảng $[0, 1]$.
 - ☒ (c) Nó có dạng lồi, dễ tối ưu.
 - (d) Nó giảm lỗi dự đoán một cách hiệu quả.
5. Trong Logistic Regression, nếu giá trị batch_size được cài đặt bằng 1, kiểu huấn luyện này được gọi là gì?
 - (a) Batch Gradient Descent.
 - ☒ (b) Stochastic Gradient Descent.
 - (c) Mini-batch Gradient Descent.
 - (d) Standard Gradient Descent.
6. Với một mẫu dữ liệu được phân loại chính xác bởi Logistic Regression, giá trị loss của mẫu này sẽ như thế nào?
 - (a) Bằng 0.5.
 - ☒ (b) Gần bằng 0.
 - (c) Bằng 1.
 - (d) Gần bằng 0.5.

7. Hàm nào sau đây mô tả đúng gradient trong quá trình tối ưu Logistic Regression?

- (a) $\nabla J(\theta) = \frac{1}{m} X^T (h_\theta(X) - y)$
- (b) $\nabla J(\theta) = \frac{1}{m} X (y - h_\theta(X))$
- (c) $\nabla J(\theta) = \frac{1}{m} X (h_\theta(X) - y)$
- (d) $\nabla J(\theta) = \frac{1}{m} \sum (h_\theta(X) - y)$

8. Hàm Sigmoid trả về giá trị trong khoảng nào?

- (a) $[-1, 1]$
- (b) $(0, 1)$
- (c) $(0, \infty)$
- (d) $[-\infty, 0]$

9. Trong quá trình huấn luyện mô hình Logistic Regression sử dụng Gradient Descent, khi cài đặt batch_size nhỏ hơn số lượng mẫu ($1 < \text{batch_size} < n_samples$), kỹ thuật này gọi là gì?

- (a) Stochastic Gradient Descent.
- (b) Mini-batch Gradient Descent.
- (c) Batch Gradient Descent.
- (d) Standard Gradient Descent.

10. Đây là lý do chính khi Logistic Regression không sử dụng Mean Squared Error làm hàm loss?

- (a) Vì Cross-Entropy dễ tối ưu hơn cho phân loại nhị phân.
- (b) Vì Mean Squared Error không hội tụ.
- (c) Vì Mean Squared Error chỉ phù hợp với hồi quy tuyến tính.
- (d) Mean Squared Error làm mô hình dễ bị overfitting.

11. Hàm nào sau đây mô tả đúng hàm loss trong Logistic Regression với y là giá trị thực và $h_\theta(x)$ là giá trị dự đoán?

- (a) $L(y, h_\theta(x)) = -[y \log(h_\theta(x)) + (1 - y) \log(1 - h_\theta(x))]$
- (b) $L(y, h_\theta(x)) = (y - h_\theta(x))^2$
- (c) $L(y, h_\theta(x)) = |y - h_\theta(x)|$
- (d) $L(y, h_\theta(x)) = y \log(1 - h_\theta(x)) + (1 - y) \log(h_\theta(x))$

12. Trong các độ đo dưới đây, độ đo nào thường không được sử dụng để đánh giá một mô hình Logistic Regression?

- (a) Accuracy.
- (b) Precision.
- (c) Binary Cross Entropy.
- (d) Mean Absolute Error.

13. Cho đoạn chương trình sau:

```
1 def predict(X, theta):
2     z = np.dot(X, theta)
3
4     return 1 / (1 + np.exp(-z))
```

Khi truyền vector $\mathbf{X} = [[22.3, -1.5, 1.1, 1]]$ và vector $\mathbf{\theta} = [0.1, -0.15, 0.3, -0.2]$ vào hàm `predict()` trên, kết quả trả về của hàm là:

- (a) 0.14239088
- (b) 0.71259201
- (c) 0.92988994
- (d) 0.54991232

14. Cho đoạn chương trình sau:

```
1 def compute_loss(y_hat, y):
2     y_hat = np.clip(
3         y_hat, 1e-7, 1 - 1e-7
4     )
5
6     return (-y * np.log(y_hat) - (1 - y) * np.log(1 - y_hat)).mean()
```

Khi truyền vector $\mathbf{y} = \text{np.array}([1, 0, 0, 1])$ và vector $\mathbf{y_hat} = \text{np.array}([0.8, 0.75, 0.3, 0.95])$ vào hàm `compute_loss()` trên, kết quả trả về của hàm là (làm tròn đến hàng thập phân thứ 3):

- (a) 0.504
- (b) 0.201
- (c) 0.921
- (d) 0.623

15. Khi mô hình Logistic Regression dự đoán giá trị 0.8 trong bài toán phân loại cảm xúc, điều đó có nghĩa là gì?

- (a) Văn bản có 80% tỉ lệ là tiêu cực.
- (b) Văn bản có 80% tỉ lệ là tích cực.
- (c) Văn bản có 20% tỉ lệ là tích cực.
- (d) Không xác định được tỉ lệ.

16. Cho đoạn chương trình sau:

```
1 def compute_gradient(X, y_true, y_pred):
2     gradient = np.dot(X.T, (y_pred - y_true)) / y_true.size
3
4     return gradient
5
```

Khi truyền $\mathbf{X} = [[1, 2], [2, 1], [1, 1], [2, 2]]$, $\mathbf{y_true} = [0, 1, 0, 1]$ và $\mathbf{y_pred} = [0.25, 0.75, 0.4, 0.8]$ vào hàm `compute_gradient()` trên, kết quả trả về của hàm là:

18

AI VIETNAM (AIO2024)

aivietnam.edu.vn

- (a) [0.100, 0.250]
- (b) [0.150, 0.200]
- (c) [-0.062, 0.062]
- (d) [0.175, 0.275]

17. Cho đoạn chương trình sau:

```
1 def compute_accuracy(y_true, y_pred):
2     y_pred_rounded = np.round(y_pred)
3     accuracy = np.mean(y_true == y_pred_rounded)
4
5     return accuracy
6
```

Khi truyền vector $\mathbf{y_true} = [1, 0, 1, 1]$ và $\mathbf{y_pred} = [0.85, 0.35, 0.9, 0.75]$ vào hàm `compute_accuracy()` trên, kết quả trả về của hàm là:

- (a) 0.75
- (b) 0.80
- (c) 0.90
- (d) 1.00

18. Cho đoạn chương trình sau:

```
1 def compute_gradient(X, y_true, y_pred):  
2     gradient = np.dot(X.T, (y_pred - y_true)) / y_true.size  
3  
4     return gradient  
5
```

Khi truyền $X = \begin{bmatrix} 1 & 3 \\ 2 & 1 \\ 3 & 2 \\ 1 & 2 \end{bmatrix}$, $y_true = [1, 0, 1, 1]$ và $y_pred = [0.7, 0.4, 0.6, 0.85]$ vào hàm `compute_gradient()` trên, kết quả trả về của hàm là:

- (a) $[-0.212, -0.4]$
- (b) $[0.025, 0.15]$
- (c) $[0.045, 0.20]$
- (d) $[0.05, 0.25]$

- Hết -