1 [15pts] AUC

对于有限样例，请证明

$$\mathrm{AUC} = \frac{1}{m^+m^-}\sum_{x^+\in D^+}\sum_{x^-\in D^-}\left(\mathbb{I}(f(x^+) > f(x^-)) + \frac{1}{2}\mathbb{I}(f(x^+) = f(x^-))\right)$$

$$\mathrm{AUC} = \sum_{i=1}^{m-1}\left[\frac{1}{2}(x_{i+1} - x_i)(y_{i+1} - y_i)\right]$$

$$= \sum_{x^-\in D^-}\frac{1}{2m^-}\left[\frac{2}{m^+}\sum_{x^+\in D^+}\mathbb{I}\big(f(x^+) > f(x^-)\big) + \frac{1}{m^+}\sum_{x^-\in D^-}\mathbb{I}\big(f(x^+) = f(x^-)\big)\right]$$

$$= \sum_{x^+\in D^+}\left[\frac{1}{m^+m^-}\sum_{x^-\in D^-}\mathbb{I}\big(f(x^+) > f(x^-)\big) + \frac{1}{2}\cdot\frac{1}{m^+m^-}\cdot\sum_{x^-\in D^-}\mathbb{I}\big(f(x^+) = f(x^-)\big)\right]$$

$$= \sum_{x^+\in D^+}\left[\frac{1}{m^+m^-}\sum_{x^-\in D^-}\mathbb{I}\big(f(x^+) > f(x^-)\big) + \frac{1}{2}\cdot\frac{1}{m^+m^-}\cdot\sum_{x^-\in D^-}\mathbb{I}\big(f(x^+) = f(x^-)\big)\right]$$

$$= \frac{1}{m^+m^-}\sum_{x^+\in D^+}\left[\sum_{x^-\in D^-}\mathbb{I}\big(f(x^+) > f(x^-)\big) + \frac{1}{2}\cdot\sum_{x^-\in D^-}\mathbb{I}\big(f(x^+) = f(x^-)\big)\right]$$

$$= \frac{1}{m^+m^-}\sum_{x^+\in D^+}\sum_{x^-\in D^-}\left(\mathbb{I}\big(f(x^+) > f(x^-)\big) + \frac{1}{2}\mathbb{I}\big(f(x^+) = f(x^-)\big)\right)$$

Each small bar of ROC curve is rectangular or trapezoidal. Either way, we can use $(\mathrm{top} + \mathrm{bottom}) \times \mathrm{height}/2$ to calculate the area, where top is $\frac{1}{m^+}\sum_{x^+\in D^+}\mathbb{I}\big(f(x^+) > f(x^-)\big)$, bottom is $\frac{1}{m^+}\left(\sum_{x^+\in D^+}\mathbb{I}\big(f(x^+) > f(x^-)\big) + \sum_{x^-\in D^-}\mathbb{I}\big(f(x^+) = f(x^-)\big)\right)$ and the height is $1/m^-$.

But in fact, we can obtain another formula in the following way.

Consider a classification algorithm that assigns to a random observation $\boldsymbol{x} \in \mathbb{R}^p$ a score (or probability) $f(\boldsymbol{x}) \in [0,1]$ signifying membership in class 1. If the final classification between class 1 and class 0 is determined by a decision threshold $t \in [0,1]$, then we can denote TPR and FPR as follows:

$$TPR = T(t) = P[f(x) > t|y(x) = 1]$$

$$FPR = F(t) = P[f(x) > t|y(x) = 0]$$

$y(x) = 1$ and $y(x) = 0$ means x belong to class 1 and x belong to class 0, respectively. If we view T as a function of F, we can get

$$
\begin{aligned}
\text{AUC} &= \int_0^1 T(F_0)dF_0 \\
&= \int_0^1 P[f(x) > F^{-1}(F_0)|y(x) = 1]dF_0 \\
&= \int_1^0 P\big[f(x) > F^{-1}\big(F(t)\big)|y(x) = 1\big] \cdot \frac{\partial F(t)}{\partial t}dt \\
&= \int_0^1 P[f(x) > t|y(x) = 1] \cdot P[f(x') = t|y(x') = 0]dt \\
&= \int_0^1 P[f(x) > t, f(x') = t|y(x) = 1, y(x') = 0]dt \\
&= P[f(x) > f(x')|y(x) = 1, y(x') = 0]
\end{aligned}
$$

At the fourth equal sign, we used the fact that the probability density function $P[f(x') = t|y(x') = 0]$ is the derivative with respect to t of the cumulative distribution function $P(f'(x) \leq t|y(x') = 0) = 1 - F(t)$.

So given a randomly chosen observation x belonging to class 1 , and randomly chosen observation x' belonging the class 0, the AUC is the probability that the evaluated classification algorithm will assign a higher score to x than to x'.

According this idea, assuming that the data set has a total of M positive samples and N negative samples, the predicted value is $M + N$. We sort all the samples according to the predicted value in **increasing order**, and sort the numbers from 1 to $M + N$.

- For the sample with the highest probability of positive sample (assuming the sort number is rank_1), the number of negative samples with a smaller probability than it is $\text{rank}_1 - M$.
- For the sample with the second highest probability of positive samples (assuming the sorting number is rank_2), the number of negative samples with a lower probability than it is $\text{rank}_2 - (M - 1)$
- And so on…
- For the smallest positive sample probability, assume that the sorting number is rank_M, the number of negative samples with a smaller probability than it is $\text{rank}_\text{M} - 1$

Then in all cases, the number of positive sample scores greater than negative samples is $\sum_{i=1}^{M} \text{rank}_i - (M + 1 - i)$, then AUC can be written as

$$
\text{AUC} = \frac{\sum_{x^+ \in D^+} rank(x^+) - \frac{M \times (1 + M)}{N}}{M \times N}
$$

.

2 [40+5 pts] 编程题, Logistic Regression

请结合编程题指南（见下页）进行理解

试考虑对率回归与线性回归的关系。最简单的对率回归的所要学习的任务仅是根据训练数据学得一个 $\beta = (\omega; b)$，而学习 β 的方式将有下述两种不同的实现：

0. [闭式解] 直接将分类标记作为回归目标做线性回归，其闭式解为

$$\beta = (\hat{X}^T \hat{X})^{-1} \hat{X}^T y \tag{2.1}$$

，其中 $\hat{X} = (X; \vec{1})$

1. [数值方法] 利用牛顿法或梯度下降法解数值问题

$$\min_{\beta} \sum_{i=1}^{m} (-y_i \beta^T \hat{x}_i + \ln(1 + e^{\beta^T \hat{x}_i})). \tag{2.2}$$

得到 β 后两个算法的决策过程是一致的，即：

(1) $z = \beta X_i$

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

(3) 决策函数

$$y_i = \begin{cases} 1, & \text{if } f > \theta \\ 0, & \text{else} \end{cases} \tag{2.3}$$

其中 θ 为分类阈值。回答下列问题：

(1) **[10 pts]** 试实现用闭式解方法训练分类器。若设分类阈值 $\theta = 0.5$，此分类器在 Validation sets 下的准确率、查准率、查全率是多少?

```
accuracy:  0.725
precision: 0.6615384615384615
recall:    1.0
```

(2) **[10 pts]** 利用所学知识选择合适的分类阈值，并输出闭式解方法训练所得分类器在 test sets 下的预测结果。

We can obtain the most suitable threshold by calculating the F1 score, the specific method is as follows:

```python
def find_best_threshold(y_true, X, model):
    z = sigmoid(X @ model)
    thresholds = np.sort(np.unique(z))
    best_f1_score = 0
    best_threshold = 0
    for threshold in thresholds:
        y_pred = classify(z, threshold)
        precision, recall = calc_pr(y_pred, y_true)
        f1_score = 2 * precision * recall / (precision + recall)
        if f1_score > best_f1_score:
            best_f1_score = f1_score
            best_threshold = threshold
    return best_threshold
```

Then, we can get the best threshold is 0.5139247728112848(In fact, 0.7192036272932429 is ok either. My point is there are very many values with the same maximum F1 score). By using this threshold, we just need to train on test set, predict and save data to a .csv file.

(3) **[10 pts]** 利用数值方法重新训练一个新的分类器。若设分类阈值 $\theta = 0.5$，此分类器在 Validation sets 下的准确率、查准率、查全率是多少?

We use $f(\beta)$ as the objective function of the optimization problem:

$$f(\beta) = \min_{\beta} \sum_{i=1}^{m} \left(-y_i \beta^T \hat{x}_i + \ln\left(1 + e^{\beta^T \hat{x}_i}\right)\right)$$

Then take the derivative to get the recursive formula of gradient decent method:

$$f'(\beta) = \sum_{i=1}^{m} \left(-y_i \hat{x}_i + \frac{\hat{x}_i e^{\beta^T \hat{x}_i}}{1 + e^{\beta^T \hat{x}_i}}\right) = \sum_{i=1}^{m} \left(-y_i \hat{x}_i + \frac{\hat{x}_i}{1 + e^{-\beta^T \hat{x}_i}}\right)$$

$$\beta_{k+1} = \beta_k - \alpha f'(\beta_k)$$

For the learning rate, we can simply set it to 0.05. The result is shown as follows:

```
accuracy:  1.0
precision: 1.0
recall:    1.0
```

(5) **[选做][Extra 5 pts]** 谈谈两种方法下分类阈值的变化对预测结果的影响，简要说明看法。

The meaning of the closed-form solution is to predict the mark as a regression value, and the value of y_i is approximated, not the value of $\log\frac{y_i}{1-y_i}$. Therefore, after obtaining the regression value, any activation function can be used to complete the decision, such as tanh, sigmoid . Since the value of z is more inclined to be between [0,1] after the mark is used as the regression value, and the sigmoid function is used (note: the sigmoid function is not a gain operation in this problem), the gradient is large near z=0, That is, the disturbance caused by small errors will have a greater impact on the sigmoid output, so this method requires a more accurate threshold.

The numerical method approximates the rate value $\log\frac{y_i}{1-y_i}$, and the range of z is not only between [0,1], but the gradient of the sigmoid function is small when z is far away from 0, that is, the small error has little influence on the output of the sigmoid. Therefore a very precise threshold is not required.

3 [15pts] Friedman 检验 [编程题]

在数据集 D_1, D_2, D_3, D_4, D_5 运行了 A, B, C, D, E 五种算法，算法比较序值表如表1所示:

表 1: 算法比较序值表

数据集	算法 A	算法 B	算法 C	算法 D	算法 E
D_1	2	3	1	5	4
D_2	5	4	2	3	1
D_3	4	5	1	2	3
D_4	2	3	1	5	4
D_5	3	4	1	5	2
平均序值	3.2	3.8	1.2	4	2.8

使用 Friedman 检验 ($\alpha = 0.05$) 判断这些算法是否性能都相同。若不相同，进行 Nemenyi 后续检验 ($\alpha = 0.05$)，并说明性能最好的算法与哪些算法有显著差别。本题需编程实现 Friedman 检验和 Nemenyi 后续检验。(预计代码行数小于 50 行)

For Friedman test, we should calculate $\tau_F \sim F\big(k-1,(k-1)(N-1)\big)$, and

$$\tau_F = \frac{(N-1)\tau_{\chi^2}}{N(k-1)-\tau_{\chi^2}}$$

$$\tau_{\chi^2} = \frac{k-1}{k}\cdot\frac{12N}{k^2-1}\sum_{i=1}^{k}\left(r_i-\frac{k+1}{2}\right)^2 = \frac{12N}{k(k+1)}\left(\sum_{i=1}^{k}r_i^2-\frac{k(k+1)^2}{4}\right)$$

where N, k, r_i are the number of data sets, the number of algorithms and the mean rank of i^{th} algorithm, respectively.

We can obtain $\tau_{\chi^2} = 9.9200, \tau_F = 3.9365 > 3.007$, which means we need to turn down "all algorithms have the same performance". Therefore, we have to do Nemenyi test next.

From $CD = q_\alpha\sqrt{\frac{k(k+1)}{6N}}$ and $q_\alpha = 2.728$, we get $CD = 2.718$. And $1.2 + 2.728 = 3.928 < 4$, so there is a significant difference between the performance of algorithms C and D, but there is no significant difference between C and other algorithms.

4 [30pts] BP 算法推导

请给出教材《机器学习》5.3 节 BP 算法的完整推导过程。注意符号的一致性。

Solution. *(中英文回答均可)*

Prove that

$$\Delta w_{hj} = \eta g_j b_h \tag{1}$$

$$\Delta\theta_j = -\eta g_j \tag{2}$$

$$\Delta v_{ih} = \eta e_h x_i \tag{3}$$

$$\Delta\gamma_h = -\eta e_h \tag{4}$$

For formula (1), we have

$$\Delta w_{hj} = -\frac{\eta\partial E_k}{\partial w_{hj}}$$

Then,

$$\begin{aligned}
\frac{\partial E_k}{\partial w_{hj}} &= \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}} \\
&= \left(\hat{y}_j^k - y_j^k\right) \cdot f'\left(\beta_j - \theta_j\right) \cdot b_h \\
&= \left(\hat{y}_j^k - y_j^k\right) \cdot b_h \cdot \left[f\left(\beta_j - \theta_j\right) \times \left(1 - f\left(\beta_j - \theta_j\right)\right)\right] \\
&= \left(\hat{y}_j^k - y_j^k\right) \cdot b_h \cdot \hat{y}_j^k\left(1 - \hat{y}_j^k\right) \\
&= -g_j b_h
\end{aligned}$$

So

$$\Delta w_{hj} = -\frac{\eta\partial E_k}{\partial w_{hj}} = \eta g_j h_h$$

For formula (2), we have

$$\Delta\theta_j = -\frac{\eta\partial E_k}{\partial \theta_j}$$

Then,

$$\begin{aligned}
\frac{\partial E_k}{\partial \theta_j} &= \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \theta_j} \\
&= \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial f\left(\beta_j - \theta_j\right)}{\partial \theta_j} \\
&= \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot f'\left(\beta_j - \theta_j\right) \times (-1) \\
&= \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot f\left(\beta_j - \theta_j\right) \times \left[1 - f\left(\beta_j - \theta_j\right)\right] \times (-1) \\
&= \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \hat{y}_j^k\left(1 - \hat{y}_j^k\right) \times (-1)
\end{aligned}$$

$$= \frac{\partial\left[\frac{1}{2}\sum_{j=1}^{l}\left(\hat{y}_j^k - y_j^k\right)^2\right]}{\partial \hat{y}_j^k} \cdot \hat{y}_j^k\left(1-\hat{y}_j^k\right) \times (-1)$$

$$= \frac{1}{2} \times 2\left(\hat{y}_j^k - y_j^k\right) \times 1 \cdot \hat{y}_j^k\left(1-\hat{y}_j^k\right) \times (-1)$$

$$= \left(y_j^k - \hat{y}_j^k\right)\hat{y}_j^k\left(1-\hat{y}_j^k\right)$$

$$= g_j$$

So

$$\Delta\theta_j = -\frac{\eta \partial E_k}{\partial \theta_j} = -\eta g_j$$

For formula (3), we have

$$\Delta v_{ih} = -\frac{\eta \partial E_k}{\partial v_{ih}}$$

Then,

$$\frac{\partial E_k}{\partial v_{ih}} = \sum_{j=1}^{l} \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial b_h} \cdot \frac{\partial b_h}{\partial \alpha_h} \cdot \frac{\partial \alpha_h}{\partial v_{ih}}$$

$$= \sum_{j=1}^{l} \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial b_h} \cdot \frac{\partial b_h}{\partial \alpha_h} \cdot x_i$$

$$= \sum_{j=1}^{l} \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial b_h} \cdot f'(\alpha_h - \gamma_h) \cdot x_i$$

$$= \sum_{j=1}^{l} \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot w_{hj} \cdot f'(\alpha_h - \gamma_h) \cdot x_i$$

$$= \sum_{j=1}^{l} \left(-g_j\right) \cdot w_{hj} \cdot f'(\alpha_h - \gamma_h) \cdot x_i$$

$$= -f'(\alpha_h - \gamma_h) \sum_{j=1}^{l} g_j \cdot w_{hj} \cdot x_i$$

$$= -b_h(1-b_h) \sum_{j=1}^{l} g_j \cdot w_{hj} \cdot x_i$$

$$= -e_h \cdot x_i$$

So

$$\Delta v_{ih} = -\frac{\eta \partial E_k}{\partial v_{ih}} = \eta e_h x_i$$

For formula (4), we have

$$\Delta\gamma_h = -\frac{\eta \partial E_k}{\partial \gamma_h}$$

Then,

$$\frac{\partial E_k}{\partial \gamma_h} = \sum_{j=1}^{l} \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial b_h} \cdot \frac{\partial b_h}{\partial \gamma_h}$$

$$= \sum_{j=1}^{l} \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial b_h} \cdot f'(\alpha_h - \gamma_h) \times (-1)$$

$$= -\sum_{j=1}^{l} \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot w_{hj} \cdot f'(\alpha_h - \gamma_h)$$

$$= -\sum_{j=1}^{l} \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot w_{hj} \cdot b_h(1 - b_h)$$

$$= \sum_{j=1}^{l} g_j \cdot w_{hj} \cdot b_h(1 - b_h)$$

$$= c_h$$

So

$$\Delta\gamma_h = -\frac{\eta \partial E_k}{\partial \gamma_h} = -\eta e_h$$