Data Analytics and Machine Learning Group
Department of Informatics
Technical University of Munich

**Esolution**
Place student sticker here

**Note:**

- During the attendance check a sticker containing a unique code will be put on this exam.
- This code contains a unique number that associates this exam with your registration number.
- This number is printed both next to the code and to the signature field in the attendance check list.

# Machine Learning

| | | | |
|---|---|---|---|
| **Graded Exercise:** | IN2064 / Retake | **Date:** | Thursday 1$^{st}$ April, 2021 |
| **Examiner:** | Prof. Dr. Stephan Günnemann | **Time:** | 16:30 – 18:30 |

## Working instructions

- This graded exercise consists of  **pages** with a total of **31 problems**.
  Please make sure now that you received a complete copy of the answer sheet.
- The total amount of achievable credits in this graded exercise is 108 credits.
- Allowed resources:
  - all materials that you will use on your own (lecture slides, calculator etc.)
  - **not allowed are any forms of collaboration between examinees and plagiarism**
- You have to sign the code of conduct. (Typing your name is fine)
- You have to either print this document and scan your solutions or paste scans/pictures of your handwritten solutions into the solution boxes in this PDF. **Editing the PDF digitally is prohibited except for signing the code of conduct and answering multiple choice questions**.
- Make sure that the **QR codes are visible** on every uploaded page. Otherwise, we cannot grade your submission.
- **You must solve the specified version of the problem**. Different problems may have different version: e.g. Problem 1 (Version A), Problem 5 (Version C), etc. If you solve the wrong version you get **zero** points.
- Only write on the provided sheets, **submitting your own additional sheets is not possible**.
- Last two pages can be used as scratch paper.
- All sheets (including scratch paper) have to be submitted to the upload queue. Missing pages will be considered empty.
- **Only use a black or blue color (no red or green)! Pencils are allowed.**
- Write your answers only in the provided solution boxes or the scratch paper.
- **For problems that say "Justify your answer" you only get points if you provide a valid explanation.**
- **For problems that say "Prove" you only get points if you provide a valid mathematical proof.**
- If a problem does not say "Justify your answer" or "Prove" it's sufficient to only provide the correct answer.
- Instructor announcements and clarifications will be posted **on Piazza** with email notifications.
- Exercise duration - 120 minutes.

Left room from ________ to ________ / Early submission at ________

## Problem 1 (Version A) (4 credits)

a)

0 1 2

Yes, it is possible, however the likelihood includes the unobserved parameter $\theta$. So to maximize the likelihood, either $\theta$ needs to be marginalized out first

$$p(\mathcal{D} \mid a, b) = \int p(\mathcal{D} \mid \theta)\, p(\theta \mid a, b)\, d\theta = \mathop{\mathbb{E}}_{p(\theta \mid a, b)} \left[ p(\mathcal{D} \mid \theta) \right]$$

or one has to use an approximate algorithm such as Expectation Maximization.

b)

0 1 2

No, it is not immediately possible because for MAP estimation you need a prior on the variables you want to infer. So you would need to introduce a hyper-prior on the parameters $a$ and $b$, for example an Exponential($\lambda$) distribution. Then you can compute an MAP estimate by maximizing $p(\mathcal{D} \mid a, b)\, p(a, b)$ where the data likelihood is computed as in a).

## Problem 1 (Version B) (4 credits)

0
1
2

a)

Yes, it is possible, however the likelihood includes the unobserved parameter $\theta$. So to maximize the likelihood, either $\theta$ needs to be marginalized out first

$$\mathrm{p}(\mathcal{D} \mid a, b) = \int \mathrm{p}(\mathcal{D} \mid \theta)\, \mathrm{p}(\theta \mid a, b)\, \mathrm{d}\theta = \mathop{\mathbb{E}}_{\mathrm{p}(\theta \mid a, b)} \left[\mathrm{p}(\mathcal{D} \mid \theta)\right]$$

or one has to use an approximate algorithm such as Expectation Maximization.

0
1
2

b)

No, it is not immediately possible because for MAP estimation you need a prior on the variables you want to infer. So you would need to introduce a hyper-prior on the parameters $a$ and $b$, for example an Exponential($\lambda$) distribution. Then you can compute an MAP estimate by maximizing $\mathrm{p}(\mathcal{D} \mid a, b)\, \mathrm{p}(a, b)$ where the data likelihood is computed as in a).

## Problem 2 (Version A) (5 credits)

a)

Bagging at feature level.

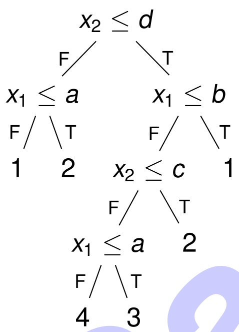
b)

There are many valid solutions. One solution is $\mathcal{D} = \{([a, b], 0), ([a, b], 1)\}$, where we have two instances with exactly the same features $\boldsymbol{x}_1 = \boldsymbol{x}_2 = [a, b] \in \mathbb{R}^2$ for any constants $a, b$, but different labels, $y_1 \neq y_2$. No matter what the split is, both instances will end up in the same leaf and the purity will not change.

Another solution is to write down the XOR dataset.

c)

The decision tree looks as follows:



## Problem 2 (Version B) (5 credits)

a)

Bagging at feature level.

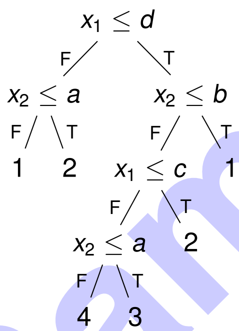
b)

There are many valid solutions. One solution is $\mathcal{D} = \{([a, b], 0), ([a, b], 1)\}$, where we have two instances with exactly the same features $\boldsymbol{x}_1 = \boldsymbol{x}_2 = [a, b] \in \mathbb{R}^2$ for any constants $a, b$, but different labels, $y_1 \neq y_2$. No matter what the split is, both instances will end up in the same leaf and the purity will not change.

Another solution is to write down the XOR dataset.

c)

The decision tree looks as follows:



## Problem 2 (Version C) (5 credits)

a)

Bagging at feature level.

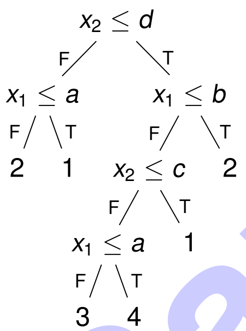
b)

There are many valid solutions. One solution is $\mathcal{D} = \{([a,b],0),([a,b],1)\}$, where we have two instances with exactly the same features $\boldsymbol{x}_1 = \boldsymbol{x}_2 = [a,b] \in \mathbb{R}^2$ for any constants $a, b$, but different labels, $y_1 \neq y_2$. No matter what the split is, both instances will end up in the same leaf and the purity will not change.

Another solution is to write down the XOR dataset.

c)

The decision tree looks as follows:



## Problem 2 (Version D) (5 credits)

a)

Bagging at feature level.

b)

There are many valid solutions. One solution is $\mathcal{D} = \{([a, b], 0), ([a, b], 1)\}$, where we have two instances with exactly the same features $\boldsymbol{x}_1 = \boldsymbol{x}_2 = [a, b] \in \mathbb{R}^2$ for any constants $a, b$, but different labels, $y_1 \neq y_2$. No matter what the split is, both instances will end up in the same leaf and the purity will not change.

Another solution is to write down the XOR dataset.

c)

The decision tree looks as follows:

- $x_1 \leq d$
  - F: $x_2 \leq a$
    - F: 2
    - T: 1
  - T: $x_2 \leq b$
    - F: $x_1 \leq c$
      - F: $x_2 \leq a$
        - F: 3
        - T: 4
      - T: 1
    - T: 2

## Problem 3 (Version A) (2 credits)

0
1
2

$f$ is linear in the parameters $\boldsymbol{w} = \begin{pmatrix} a & b & c \end{pmatrix}^\top$ which becomes more apparent if we rewrite it as

$$f(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{w}^\top \begin{pmatrix} \sin(\boldsymbol{x}_2) \\ \frac{1}{2}\|\boldsymbol{x}\|_1 \\ -\boldsymbol{x}_1^2 \boldsymbol{x}_2 \end{pmatrix}.$$

In this form, the problem is linear in the parameters. We define a feature transformation

$$\phi(\boldsymbol{x}) = \begin{pmatrix} \sin(\boldsymbol{x}_2) \\ \frac{1}{2}\|\boldsymbol{x}\|_1 \\ -\boldsymbol{x}_1^2 \boldsymbol{x}_2 \end{pmatrix}.$$

and can now write the problem in the usual linear regression form

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{w}$$

where $\boldsymbol{y}_i = y_i$ and $\boldsymbol{\Phi}_{i,:} = \phi(\boldsymbol{x}_i)^\top$. Then we can apply the closed form for ordinary least squares and get

$$\boldsymbol{w}^* = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{y}.$$

## Problem 3 (Version B) (2 credits)

0
1
2

$f$ is linear in the parameters $\mathbf{w} = \begin{pmatrix} a & b & c \end{pmatrix}^\top$ which becomes more apparent if we rewrite it as

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^\top \begin{pmatrix} \|\mathbf{x}\|_2 \\ -\frac{1}{2}\mathbf{x}_1^2\mathbf{x}_2 \\ \cos(\mathbf{x}_1) \end{pmatrix}.$$

In this form, the problem is linear in the parameters. We define a feature transformation

$$\phi(\mathbf{x}) = \begin{pmatrix} \|\mathbf{x}\|_2 \\ -\frac{1}{2}\mathbf{x}_1^2\mathbf{x}_2 \\ \cos(\mathbf{x}_1) \end{pmatrix}.$$

and can now write the problem in the usual linear regression form

$$\mathbf{y} = \mathbf{\Phi}\mathbf{w}$$

where $\mathbf{y}_i = y_i$ and $\mathbf{\Phi}_{i,:} = \phi(\mathbf{x}_i)^\top$. Then we can apply the closed form for ordinary least squares and get

$$\mathbf{w}^* = (\mathbf{\Phi}^\top\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top\mathbf{y}.$$

## Problem 3 (Version C) (2 credits)

0
1
2

$f$ is linear in the parameters $\mathbf{w} = \begin{pmatrix} a & b & c \end{pmatrix}^{\top}$ which becomes more apparent if we rewrite it as

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^{\top} \begin{pmatrix} -\mathbf{x}_1^2 \mathbf{x}_2 \\ \tan(\mathbf{x}_2) \\ \frac{1}{2} \|\mathbf{x}\|_{\infty} \end{pmatrix}.$$

In this form, the problem is linear in the parameters. We define a feature transformation

$$\phi(\mathbf{x}) = \begin{pmatrix} -\mathbf{x}_1^2 \mathbf{x}_2 \\ \tan(\mathbf{x}_2) \\ \frac{1}{2} \|\mathbf{x}\|_{\infty} \end{pmatrix}.$$

and can now write the problem in the usual linear regression form

$$\mathbf{y} = \mathbf{\Phi} \mathbf{w}$$

where $\mathbf{y}_i = y_i$ and $\mathbf{\Phi}_{i,:} = \phi(\mathbf{x}_i)^{\top}$. Then we can apply the closed form for ordinary least squares and get

$$\mathbf{w}^* = (\mathbf{\Phi}^{\top} \mathbf{\Phi})^{-1} \mathbf{\Phi}^{\top} \mathbf{y}.$$

## Problem 4 (Version A) (6 credits)

0 1 2

a)

A weight $\boldsymbol{w}^*$ on $\mathcal{D}$ is equivalent to a weight $\boldsymbol{w}^{*,\alpha} = \boldsymbol{w}^*/\alpha$ on $\mathcal{D}_\alpha$ and vice versa. The norm of $\boldsymbol{w}^*$ will stay finite because the dataset is not linearly separable. Therefore, the two classifiers will reach the same log-likelihood and for the optimal weights it holds that $\boldsymbol{w}^{*,\alpha} = \boldsymbol{w}^*/\alpha$. So for any $\boldsymbol{x}_{\text{test}}$ with $\boldsymbol{w}^* \cdot \boldsymbol{x}_{\text{test}} = 0$ (i.e. on the decision boundary for a classifier with threshold 0.5), we will have $s = t$.
A logistic regression model is

$$f(\boldsymbol{x}; \boldsymbol{w}) = \sigma(\boldsymbol{x}^\mathsf{T} \boldsymbol{w})$$

and the sigmoid function is strictly monotonic, meaning that $\boldsymbol{x}^\mathsf{T}\boldsymbol{w} < \boldsymbol{x}^\mathsf{T}\boldsymbol{w}' \Rightarrow \sigma(\boldsymbol{x}^\mathsf{T}\boldsymbol{w}) < \sigma(\boldsymbol{x}^\mathsf{T}\boldsymbol{w}')$.
Because of $\alpha > 1$, we have

$$|\boldsymbol{w}^{*,\alpha} \cdot \boldsymbol{x}_{\text{test}}| = \alpha^{-1} |\boldsymbol{w}^* \cdot \boldsymbol{x}_{\text{test}}| < |\boldsymbol{w}^* \cdot \boldsymbol{x}_{\text{test}}|.$$

So we can have both $s < t$ and $s > t$ depending on the sign of $\boldsymbol{w}^* \cdot \boldsymbol{x}_{\text{test}}$.

0 1 2

b)

In the unregularized setting, the MLE will have infinite norm. But with a weight vector $\boldsymbol{w}$ of infinite norm $\|\boldsymbol{w}\| = \infty$, the model $f(\boldsymbol{x}; \boldsymbol{w})$ will only take on three values: 0 or 1 if $\boldsymbol{x}$ is on either side of the hyperplane defined by $\boldsymbol{w}$ or $\frac{1}{2}$ if $\boldsymbol{x}$ is exactly on the hyper plane. Therefore any $\boldsymbol{w}$ such that the mapping $\boldsymbol{w} \cdot \boldsymbol{x}$ linearly separates the data are equivalent. Because in this setup class 1 is truly contained in the first quadrant and class 0 in the third, any hyperplane that separates the two achieves the same log-likelihood, so the optimal $\boldsymbol{w}$ is not unique.

0 1 2

c)

$$\boldsymbol{w}^{*,a} = \begin{pmatrix} \infty \\ 0 \end{pmatrix} \quad \text{and} \quad \boldsymbol{w}^{*,b} = \begin{pmatrix} 0 \\ \infty \end{pmatrix}.$$

As explained in the previous problem, any hyperplane that separates the two quadrants will work and so we can choose the $y$ and $x$ axes. A differently classified test point is

$$\boldsymbol{x}_{\text{test}} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

because

$$f(\boldsymbol{x}_{\text{test}}; \boldsymbol{w}^{*,a}) = \sigma(-\infty) = 0 < \frac{1}{2} < 1 = \sigma(\infty) = f(\boldsymbol{x}_{\text{test}}; \boldsymbol{w}^{*,b}).$$

## Problem 4 (Version B) (6 credits)

a)

A weight $\boldsymbol{w}^*$ on $\mathcal{D}$ is equivalent to a weight $\boldsymbol{w}^{*,\alpha} = \boldsymbol{w}^*/\alpha$ on $\mathcal{D}_\alpha$ and vice versa. The norm of $\boldsymbol{w}^*$ will stay finite because the dataset is not linearly separable. Therefore, the two classifiers will reach the same log-likelihood and for the optimal weights it holds that $\boldsymbol{w}^{*,\alpha} = \boldsymbol{w}^*/\alpha$. So for any $\boldsymbol{x}_{\text{test}}$ with $\boldsymbol{w}^* \cdot \boldsymbol{x}_{\text{test}} = 0$ (i.e. on the decision boundary for a classifier with threshold 0.5), we will have $s = t$.
A logistic regression model is

$$f(\boldsymbol{x}; \boldsymbol{w}) = \sigma(\boldsymbol{x}^\mathsf{T} \boldsymbol{w})$$

and the sigmoid function is strictly monotonic, meaning that $\boldsymbol{x}^\mathsf{T}\boldsymbol{w} < \boldsymbol{x}^\mathsf{T}\boldsymbol{w}' \Rightarrow \sigma(\boldsymbol{x}^\mathsf{T}\boldsymbol{w}) < \sigma(\boldsymbol{x}^\mathsf{T}\boldsymbol{w}')$.
Because of $\alpha > 1$, we have

$$|\boldsymbol{w}^{*,\alpha} \cdot \boldsymbol{x}_{\text{test}}| = \alpha^{-1} |\boldsymbol{w}^* \cdot \boldsymbol{x}_{\text{test}}| < |\boldsymbol{w}^* \cdot \boldsymbol{x}_{\text{test}}|.$$

So we can have both $s < t$ and $s > t$ depending on the sign of $\boldsymbol{w}^* \cdot \boldsymbol{x}_{\text{test}}$.

b)

In the unregularized setting, the MLE will have infinite norm. But with a weight vector $\boldsymbol{w}$ of infinite norm $\|\boldsymbol{w}\| = \infty$, the model $f(\boldsymbol{x}; \boldsymbol{w})$ will only take on three values: 0 or 1 if $\boldsymbol{x}$ is on either side of the hyperplane defined by $\boldsymbol{w}$ or $\frac{1}{2}$ if $\boldsymbol{x}$ is exactly on the hyper plane. Therefore any $\boldsymbol{w}$ such that the mapping $\boldsymbol{w} \cdot \boldsymbol{x}$ linearly separates the data are equivalent. Because in this setup class 1 is truly contained in the first quadrant and class 0 in the third, any hyperplane that separates the two achieves the same log-likelihood, so the optimal $\boldsymbol{w}$ is not unique.

c)

$$\boldsymbol{w}^{*,a} = \begin{pmatrix} \infty \\ 0 \end{pmatrix} \quad \text{and} \quad \boldsymbol{w}^{*,b} = \begin{pmatrix} 0 \\ \infty \end{pmatrix}.$$

As explained in the previous problem, any hyperplane that separates the two quadrants will work and so we can choose the $y$ and $x$ axes. A differently classified test point is

$$\boldsymbol{x}_{\text{test}} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

because

$$f(\boldsymbol{x}_{\text{test}}; \boldsymbol{w}^{*,a}) = \sigma(-\infty) = 0 < \frac{1}{2} < 1 = \sigma(\infty) = f(\boldsymbol{x}_{\text{test}}; \boldsymbol{w}^{*,b}).$$

## Problem 5 (Version A) (3 credits)

0
1

a)

Yes, since the line search procedure is guaranteed to find a point where the objective function is at least as low as at the previous iteration.

Specifically, if the gradient is nonzero, line search is guaranteed to find a point in the direction of the gradient where the objective function is lower. Such point always exists since the function $f$ is continuously differentiable. If the gradient is zero, line search will produce $\theta_{t+1} = \theta_t$, which means that $f(\theta_{t+1}) = f(\theta_t)$.

0
1

b)

No. If the step size is too large, we may overshoot and land at a point that has a higher value of the objective function.

0
1

c)

No. The adaptive step size can still be too large, and we still can overshoot, similar to the fixed size case.

## Problem 5 (Version B) (3 credits)

a)

0
1

No. If the step size is too large, we may overshoot and land at a point that has a higher value of the objective function.

b)

0
1

No. The adaptive step size can still be too large, and we still can overshoot, similar to the fixed size case.

c)

0
1

Yes, since the line search procedure is guaranteed to find a point where the objective function is at least as low as at the previous iteration.

Specifically, if the gradient is nonzero, line search is guaranteed to find a point in the direction of the gradient where the objective function is lower. Such point always exists since the function $f$ is continuously differentiable. If the gradient is zero, line search will produce $\theta_{t+1} = \theta_t$, which means that $f(\theta_{t+1}) = f(\theta_t)$.

## Problem 5 (Version C) (3 credits)

0 1

a)

Yes, since the line search procedure is guaranteed to find a point where the objective function is at least as low as at the previous iteration.

Specifically, if the gradient is nonzero, line search is guaranteed to find a point in the direction of the gradient where the objective function is lower. Such point always exists since the function $f$ is continuously differentiable. If the gradient is zero, line search will produce $\theta_{t+1} = \theta_t$, which means that $f(\theta_{t+1}) = f(\theta_t)$.

0 1

b)

No. The adaptive step size can still be too large, and we still can overshoot, similar to the fixed size case.

0 1

c)

No. If the step size is too large, we may overshoot and land at a point that has a higher value of the objective function.

## Problem 5 (Version D) (3 credits)

a)

0
1

No. The adaptive step size can still be too large, and we still can overshoot, similar to the fixed size case.

b)

0
1

No. If the step size is too large, we may overshoot and land at a point that has a higher value of the objective function.

c)

0
1

Yes, since the line search procedure is guaranteed to find a point where the objective function is at least as low as at the previous iteration.

Specifically, if the gradient is nonzero, line search is guaranteed to find a point in the direction of the gradient where the objective function is lower. Such point always exists since the function $f$ is continuously differentiable. If the gradient is zero, line search will produce $\theta_{t+1} = \theta_t$, which means that $f(\theta_{t+1}) = f(\theta_t)$.

## Problem 6 (Version A) (3 credits)

0
1

a)

Option 1:
```
out = np.log(1 + np.exp(x @ y))
```

Option 2:
```
out = np.log1p(np.exp(x @ y))
```

We can also replace `x @ y` with `np.dot(x, y)` or `x.dot(y)`.

0
1
2

b)

The gradients w.r.t. $\boldsymbol{x}$ and $\boldsymbol{y}$ are

$$\frac{\partial}{\partial \boldsymbol{x}} \log(1 + \exp(\boldsymbol{x}^T\boldsymbol{y})) = \frac{1}{1 + \exp(\boldsymbol{x}^T\boldsymbol{y})} \exp(\boldsymbol{x}^T\boldsymbol{y})\boldsymbol{y}^T$$
$$\frac{\partial}{\partial \boldsymbol{y}} \log(1 + \exp(\boldsymbol{x}^T\boldsymbol{y})) = \frac{1}{1 + \exp(\boldsymbol{x}^T\boldsymbol{y})} \exp(\boldsymbol{x}^T\boldsymbol{y})\boldsymbol{x}^T$$

We can implement these computations in Numpy as
```
s = np.exp(x @ y) / (1 + np.exp(x @ y))
d_x = d_out * s * y
d_y = d_out * s * x
```

We can also replace
```
s = np.exp(x @ y) / (1 + np.exp(x @ y))
```
with
```
s = 1 / (1 + np.exp(-x @ y))
```
since these are two equivalent ways to compute the sigmoid function.

## Problem 6 (Version B) (3 credits)

a)

Option 1:
```
out = np.log(np.exp(x @ y) - 1)
```
Option 2:
```
out = np.log(np.expm1(x @ y))
```
We can also replace `x @ y` with `np.dot(x, y)` or `x.dot(y)`.

b)

The gradients w.r.t. $\boldsymbol{x}$ and $\boldsymbol{y}$ are

$$\frac{\partial}{\partial \boldsymbol{x}} \log(\exp(\boldsymbol{x}^T \boldsymbol{y}) - 1) = \frac{1}{\exp(\boldsymbol{x}^T \boldsymbol{y}) - 1} \exp(\boldsymbol{x}^T \boldsymbol{y}) \boldsymbol{y}^T$$
$$\frac{\partial}{\partial \boldsymbol{y}} \log(\exp(\boldsymbol{x}^T \boldsymbol{y}) - 1) = \frac{1}{\exp(\boldsymbol{x}^T \boldsymbol{y}) - 1} \exp(\boldsymbol{x}^T \boldsymbol{y}) \boldsymbol{x}^T$$

We can implement these computations in Numpy as
```
s = np.exp(x @ y) / (np.exp(x @ y) - 1)
d_x = d_out * s * y
d_y = d_out * s * x
```

## Problem 6 (Version C) (3 credits)

0
1

a)

Option 1:

```
out = np.log(1 + np.exp(x @ y))
```

Option 2:

```
out = np.log1p(np.exp(x @ y))
```

We can also replace `x @ y` with `np.dot(x, y)` or `x.dot(y)`.

0
1
2

b)

The gradients w.r.t. $\boldsymbol{x}$ and $\boldsymbol{y}$ are

$$\frac{\partial}{\partial \boldsymbol{x}} \log(1 + \exp(\boldsymbol{x}^T \boldsymbol{y})) = \frac{1}{1 + \exp(\boldsymbol{x}^T \boldsymbol{y})} \exp(\boldsymbol{x}^T \boldsymbol{y}) \boldsymbol{y}^T$$
$$\frac{\partial}{\partial \boldsymbol{y}} \log(1 + \exp(\boldsymbol{x}^T \boldsymbol{y})) = \frac{1}{1 + \exp(\boldsymbol{x}^T \boldsymbol{y})} \exp(\boldsymbol{x}^T \boldsymbol{y}) \boldsymbol{x}^T$$

We can implement these computations in Numpy as

```
s = np.exp(x @ y) / (1 + np.exp(x @ y))
d_x = d_out * s * y
d_y = d_out * s * x
```

We can also replace

```
s = np.exp(x @ y) / (1 + np.exp(x @ y))
```

with

```
s = 1 / (1 + np.exp(-x @ y))
```

since these are two equivalent ways to compute the sigmoid function.

## Problem 6 (Version D) (3 credits)

a)

Option 1:
```
out = np.log(np.exp(x @ y) - 1)
```

Option 2:
```
out = np.log(np.expm1(x @ y))
```

We can also replace `x @ y` with `np.dot(x, y)` or `x.dot(y)`.

b)

The gradients w.r.t. $\boldsymbol{x}$ and $\boldsymbol{y}$ are

$$\frac{\partial}{\partial \boldsymbol{x}} \log(\exp(\boldsymbol{x}^T \boldsymbol{y}) - 1) = \frac{1}{\exp(\boldsymbol{x}^T \boldsymbol{y}) - 1} \exp(\boldsymbol{x}^T \boldsymbol{y}) \boldsymbol{y}^T$$

$$\frac{\partial}{\partial \boldsymbol{y}} \log(\exp(\boldsymbol{x}^T \boldsymbol{y}) - 1) = \frac{1}{\exp(\boldsymbol{x}^T \boldsymbol{y}) - 1} \exp(\boldsymbol{x}^T \boldsymbol{y}) \boldsymbol{x}^T$$

We can implement these computations in Numpy as
```
s = np.exp(x @ y) / (np.exp(x @ y) - 1)
d_x = d_out * s * y
d_y = d_out * s * x
```

## Problem 7 (Version A) (3 credits)

0 1 2 3

The rules we know are:

1. $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = k_1(\boldsymbol{x}_1, \boldsymbol{x}_2) + k_2(\boldsymbol{x}_1, \boldsymbol{x}_2)$
2. $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = c \cdot k_1(\boldsymbol{x}_1, \boldsymbol{x}_2)$, with $c > 0$
3. $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = k_1(\boldsymbol{x}_1, \boldsymbol{x}_2) \cdot k_2(\boldsymbol{x}_1, \boldsymbol{x}_2)$
4. $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = k_3(\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2))$, with the kernel $k_3$ on $\mathcal{X}' \subseteq \mathbb{R}^M$ and $\phi : \mathcal{X} \to \mathcal{X}'$
5. $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}_1^T \boldsymbol{A} \boldsymbol{x}_2$, with $\boldsymbol{A} \in \mathbb{R}^N \times \mathbb{R}^N$ symmetric and positive semidefinite
6. $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(k_6(\boldsymbol{x}_1, \boldsymbol{x}_2))$

$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sigma^2 \left(-\frac{1}{2}(\boldsymbol{x}_1 - \boldsymbol{x}_2)\Sigma^{-1}(\boldsymbol{x}_1 - \boldsymbol{x}_2)\right)$ is a kernel by rule (2), iff $\exp\left(-\frac{1}{2}(\boldsymbol{x}_1 - \boldsymbol{x}_2)\Sigma^{-1}(\boldsymbol{x}_1 - \boldsymbol{x}_2)\right)$ is.
Now let us rearrange the given equation:

$$\begin{aligned}
-\frac{1}{2}(\boldsymbol{x}_1 - \boldsymbol{x}_2)^\top\Sigma^{-1}(\boldsymbol{x}_1 - \boldsymbol{x}_2) &= -\frac{1}{2}[\boldsymbol{x}_1^\top\Sigma^{-1}\boldsymbol{x}_1 - \boldsymbol{x}_1^\top\Sigma^{-1}\boldsymbol{x}_2 - \boldsymbol{x}_2^\top\Sigma^{-1}\boldsymbol{x}_1 + \boldsymbol{x}_2^\top\Sigma^{-1}\boldsymbol{x}_2] \\
&= -\frac{1}{2}\boldsymbol{x}_1^\top\Sigma^{-1}\boldsymbol{x}_1 + \boldsymbol{x}_1^\top\Sigma^{-1}\boldsymbol{x}_2 - \frac{1}{2}\boldsymbol{x}_2^\top\Sigma^{-1}\boldsymbol{x}_2
\end{aligned}$$

$$\Rightarrow \exp\left(-\frac{1}{2}(\boldsymbol{x}_1 - \boldsymbol{x}_2)^\top\Sigma^{-1}(\boldsymbol{x}_1 - \boldsymbol{x}_2)\right) = \exp\left(-\frac{1}{2}\boldsymbol{x}_1^\top\Sigma^{-1}\boldsymbol{x}_1\right)\exp\left(\boldsymbol{x}_1^\top\Sigma^{-1}\boldsymbol{x}_2\right)\exp\left(-\frac{1}{2}\boldsymbol{x}_2^\top\Sigma^{-1}\boldsymbol{x}_2\right)$$

$\exp\left(-\frac{1}{2}\boldsymbol{x}_1^\top\Sigma^{-1}\boldsymbol{x}_1\right)\exp\left(-\frac{1}{2}\boldsymbol{x}_2^\top\Sigma^{-1}\boldsymbol{x}_2\right)$ is a kernel by rule (4) with $\phi(\boldsymbol{x}) = \exp(-\frac{1}{2}\boldsymbol{x}^\top\Sigma^{-1}\boldsymbol{x})$.
$\exp(\boldsymbol{x}_1^\top\Sigma^{-1}\boldsymbol{x}_2)$ is a kernel by rules (5) and rule (6) if $\Sigma^{-1}$ is PSD.
We know that $\Sigma \in \mathbb{R}^{D\times D}$ is invertible and positive semi-definite. We define $\boldsymbol{a} = \Sigma\boldsymbol{b}$ and write:

$$\boldsymbol{a}^\top\Sigma^{-1}\boldsymbol{a} = \boldsymbol{b}^\top\Sigma^\top\Sigma^{-1}\Sigma\boldsymbol{b} = \boldsymbol{b}^\top\Sigma\boldsymbol{b}$$

Since $\Sigma$ is PSD (i.e $\boldsymbol{b}^\top\Sigma\boldsymbol{b} > 0$, $\forall\boldsymbol{b} \in \mathbb{R}^d\backslash\{0\}$), then so is $\Sigma^{-1}$ (i.e $\boldsymbol{a}^\top\Sigma^{-1}\boldsymbol{a} > 0$, $\forall\boldsymbol{a} \in \mathbb{R}^d\backslash\{0\}$).

## Problem 7 (Version B) (3 credits)

0 1 2 3

The rules we know are:

1. $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = k_1(\boldsymbol{x}_1, \boldsymbol{x}_2) + k_2(\boldsymbol{x}_1, \boldsymbol{x}_2)$
2. $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = c \cdot k_1(\boldsymbol{x}_1, \boldsymbol{x}_2)$, with $c > 0$
3. $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = k_1(\boldsymbol{x}_1, \boldsymbol{x}_2) \cdot k_2(\boldsymbol{x}_1, \boldsymbol{x}_2)$
4. $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = k_3(\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2))$, with the kernel $k_3$ on $\mathcal{X}' \subseteq \mathbb{R}^M$ and $\phi : \mathcal{X} \to \mathcal{X}'$
5. $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}_1^T \boldsymbol{A} \boldsymbol{x}_2$, with $\boldsymbol{A} \in \mathbb{R}^N \times \mathbb{R}^N$ symmetric and positive semidefinite
6. $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(k_6(\boldsymbol{x}_1, \boldsymbol{x}_2))$

$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sigma^2 \left(-\frac{1}{2}(\boldsymbol{x}_1 - \boldsymbol{x}_2)\Sigma^{-1}(\boldsymbol{x}_1 - \boldsymbol{x}_2)\right)$ is a kernel by rule (2), iff $\exp\left(-\frac{1}{2}(\boldsymbol{x}_1 - \boldsymbol{x}_2)\Sigma^{-1}(\boldsymbol{x}_1 - \boldsymbol{x}_2)\right)$ is.
Now let us rearrange the given equation:

$$\begin{aligned} -\frac{1}{2}(\boldsymbol{x}_1 - \boldsymbol{x}_2)^\top \Sigma^{-1} (\boldsymbol{x}_1 - \boldsymbol{x}_2) &= -\frac{1}{2}[\boldsymbol{x}_1^\top \Sigma^{-1} \boldsymbol{x}_1 - \boldsymbol{x}_1^\top \Sigma^{-1} \boldsymbol{x}_2 - \boldsymbol{x}_2^\top \Sigma^{-1} \boldsymbol{x}_1 + \boldsymbol{x}_2^\top \Sigma^{-1} \boldsymbol{x}_2] \\ &= -\frac{1}{2}\boldsymbol{x}_1^\top \Sigma^{-1} \boldsymbol{x}_1 + \boldsymbol{x}_1^\top \Sigma^{-1} \boldsymbol{x}_2 - \frac{1}{2}\boldsymbol{x}_2^\top \Sigma^{-1} \boldsymbol{x}_2 \end{aligned}$$

$$\Rightarrow \exp\left(-\frac{1}{2}(\boldsymbol{x}_1 - \boldsymbol{x}_2)^\top \Sigma^{-1} (\boldsymbol{x}_1 - \boldsymbol{x}_2)\right) = \exp\left(-\frac{1}{2}\boldsymbol{x}_1^\top \Sigma^{-1} \boldsymbol{x}_1\right) \exp\left(\boldsymbol{x}_1^\top \Sigma^{-1} \boldsymbol{x}_2\right) \exp\left(-\frac{1}{2}\boldsymbol{x}_2^\top \Sigma^{-1} \boldsymbol{x}_2\right)$$

$\exp\left(-\frac{1}{2}\boldsymbol{x}_1^\top \Sigma^{-1} \boldsymbol{x}_1\right) \exp\left(-\frac{1}{2}\boldsymbol{x}_2^\top \Sigma^{-1} \boldsymbol{x}_2\right)$ is a kernel by rule (4) with $\phi(\boldsymbol{x}) = \exp(-\frac{1}{2}\boldsymbol{x}^\top \Sigma^{-1} \boldsymbol{x})$.
$\exp(\boldsymbol{x}_1^\top \Sigma^{-1} \boldsymbol{x}_2)$ is a kernel by rules (5) and rule (6) if $\Sigma^{-1}$ is PSD.
We know that $\Sigma \in \mathbb{R}^{D \times D}$ is invertible and positive semi-definite. We define $\boldsymbol{a} = \Sigma \boldsymbol{b}$ and write:

$$\boldsymbol{a}^\top \Sigma^{-1} \boldsymbol{a} = \boldsymbol{b}^\top \Sigma^\top \Sigma^{-1} \Sigma \boldsymbol{b} = \boldsymbol{b}^\top \Sigma \boldsymbol{b}$$

Since $\Sigma$ is PSD (i.e $\boldsymbol{b}^\top \Sigma \boldsymbol{b} > 0$, $\forall \boldsymbol{b} \in \mathbb{R}^d \backslash \{0\}$), then so is $\Sigma^{-1}$ (i.e $\boldsymbol{a}^\top \Sigma^{-1} \boldsymbol{a} > 0$, $\forall \boldsymbol{a} \in \mathbb{R}^d \backslash \{0\}$).

## Problem 8 (Version A) (4 credits)

0
1
2

a)

The negative log-likelihood (NLL) is

$$
\begin{aligned}
-\log p(\boldsymbol{X}|\boldsymbol{a}, \boldsymbol{b}) &= -\sum_{i=1}^{N}\sum_{j=1}^{D} \log p(X_{ij}|\boldsymbol{a}, \boldsymbol{b}) \\
&= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{D}(X_{ij} - a_i b_j)^2 \\
&= \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{a}\boldsymbol{b}^T\|_F^2
\end{aligned}
$$

We see that minimizing the NLL is equivalent to minimizing the Frobenius norm of the difference between $\boldsymbol{X}$ and the rank-1 matrix $\boldsymbol{a}\boldsymbol{b}^T$.

In other words, we are looking for the optimal rank-1 approximation of the matrix $\boldsymbol{X}$. We know from the lecture that this can be done using the left and right singular vectors of $\boldsymbol{X}$ corresponding to the largest singular value of $\boldsymbol{X}$. That is, we can set $\boldsymbol{a} = \sqrt{\sigma_1}\boldsymbol{u}_1$ and $\boldsymbol{b} = \sqrt{\sigma_1}\boldsymbol{v}_1$.

0
1
2

b)

No, the solution is not unique. If $\boldsymbol{a}^\star$ and $\boldsymbol{b}^\star$ minimize the negative log-likelihood, then so do $\frac{1}{c}\boldsymbol{a}^\star$ and $c\boldsymbol{b}^\star$ for any scalar $c \in \mathbb{R}$.

## Problem 9 (Version A) (2 credits)

0
1
2

(1) and (5) are correct for $\sigma = 2$ and $\sigma = 5$, respectively.

1. First column is correct.
2. Second column shows a distance instead of similarity.
3. Third column misses one instance in the lower left cluster and it is located at the center instead $(2.75, 3.5)$.
4. Fourth column shows an asymmetrical matrix.
5. Upper row $\sigma = 2$, lower row $\sigma = 5$.

## Problem 9 (Version B) (2 credits)

0
1
2

(3) and (7) are correct for $\sigma = 5$ and $\sigma = 2$, respectively.

1. First column shows a distance instead of similarity.
2. Second column misses one instance in the lower left cluster and it is located at the center instead $(2.75, 3.5)$.
3. Third column is correct.
4. Fourth column shows an asymmetrical matrix.
5. Upper row $\sigma = 5$, lower row $\sigma = 2$.

## Problem 9 (Version C) (2 credits)

0 1 2

(4) and (8) are correct for $\sigma = 5$ and $\sigma = 2$, respectively.

1. First column misses one instance in the lower left cluster and it is located at the center instead $(2.75, 3.5)$.
2. Second column shows an asymmetrical matrix.
3. Third column shows a distance instead of similarity.
4. Fourth column is correct.
5. Upper row $\sigma = 5$, lower row $\sigma = 2$.

## Problem 9 (Version D) (2 credits)

0
1
2

(2) and (6) are correct for $\sigma = 2$ and $\sigma = 5$, respectively.

1. First column misses one instance in the lower left cluster and it is located at the center instead $(2.75, 3.5)$.
2. Second column is correct.
3. Third column shows an asymmetrical matrix.
4. Fourth column shows a distance instead of similarity.
5. Upper row $\sigma = 5$, lower row $\sigma = 2$.

## Problem 10 (All versions) (4 credits)

a)

0 1 2

At the decision boundary we have

$$\begin{aligned} & \|\boldsymbol{x}-\boldsymbol{\mu}_1\|_2 = \|\boldsymbol{x}-\boldsymbol{\mu}_2\|_2 \\ \Leftrightarrow\quad & \|\boldsymbol{x}-\boldsymbol{\mu}_1\|_2^2 = \|\boldsymbol{x}-\boldsymbol{\mu}_2\|_2^2 \\ \Leftrightarrow\quad & (\boldsymbol{x}-\boldsymbol{\mu}_1)^T(\boldsymbol{x}-\boldsymbol{\mu}_1) = (\boldsymbol{x}-\boldsymbol{\mu}_2)^T(\boldsymbol{x}-\boldsymbol{\mu}_2) \\ \Leftrightarrow\quad & \boldsymbol{x}^2 - 2\boldsymbol{\mu}_1^T\boldsymbol{x} + \boldsymbol{\mu}_1^2 = \boldsymbol{x}^2 - 2\boldsymbol{\mu}_2^T\boldsymbol{x} + \boldsymbol{\mu}_2^2 \\ \Leftrightarrow\quad & \boldsymbol{\mu}_1^2 - \boldsymbol{\mu}_2^2 + (2\boldsymbol{\mu}_2 - 2\boldsymbol{\mu}_1)^T\boldsymbol{x} = 0. \end{aligned} \tag{27.1}$$

We can thus define the decision boundary as the hyperplane $x_0 + \boldsymbol{w}^T\boldsymbol{x} = 0$ with $x_0 = \boldsymbol{\mu}_1^2 - \boldsymbol{\mu}_2^2$ and $\boldsymbol{w} = 2\boldsymbol{\mu}_2 - 2\boldsymbol{\mu}_1$.

b)

0 1 2

[This is a full proof, not just the justification students need to give.]
At the decision boundary we have

$$\begin{aligned} & \gamma(\boldsymbol{z}_{i1}) = \gamma(\boldsymbol{z}_{i2}) \\ \Leftrightarrow\quad & \pi_1\mathcal{N}(\boldsymbol{x}_i \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) = \pi_2\mathcal{N}(\boldsymbol{x}_i \mid \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \\ \Leftrightarrow\quad & \log\pi_1 - \frac{1}{2}\log((2\pi)^d|\boldsymbol{\Sigma}_1|) - \frac{1}{2}(\boldsymbol{x}_i-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_2) = \log\pi_1 - \frac{1}{2}\log((2\pi)^d|\boldsymbol{\Sigma}_1|) - \frac{1}{2}(\boldsymbol{x}_i-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_2), \end{aligned} \tag{27.2}$$

where we have used monotonicity of the log function. To obtain a linear decision boundary the quadratic term must drop out, i.e.

$$\boldsymbol{x}^T\boldsymbol{\Sigma}_1^{-1}\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{\Sigma}_2^{-1}\boldsymbol{x} \quad \forall\boldsymbol{x}. \tag{27.3}$$

Since $\boldsymbol{\Sigma}$ is a covariance matrix and invertible, it must be positive definite. The same holds for its inverse, which must thus have a decomposition $\boldsymbol{\Sigma}^{-1} = \boldsymbol{V}^T\boldsymbol{V}$, where $\boldsymbol{V}$ is invertible as well. Using this, we have

$$\begin{aligned} & \boldsymbol{x}^T\boldsymbol{V}_1^T\boldsymbol{V}_1\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{V}_2^T\boldsymbol{V}_2\boldsymbol{x} \quad \forall\boldsymbol{x} \\ \Leftrightarrow\quad & \|\boldsymbol{V}_1\boldsymbol{x}\|_2 = \|\boldsymbol{V}_2\boldsymbol{x}\|_2 \quad \forall\boldsymbol{x} \\ \Leftrightarrow\quad & \|\boldsymbol{y}\|_2 = \|\boldsymbol{V}_2\boldsymbol{V}_1^{-1}\boldsymbol{y}\|_2 \quad \forall\boldsymbol{y}, \end{aligned} \tag{27.4}$$

where we have substituted $\boldsymbol{y} = \boldsymbol{V}_1\boldsymbol{x}$. We can do this due to the full rank (invertibility) of $\boldsymbol{V}_1$. $\boldsymbol{U} = \boldsymbol{V}_2\boldsymbol{V}_1^{-1}$ must therefore be an isometry with respect to the $L_2$ norm, i.e. a unitary matrix with $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$. Due to invertibility $\boldsymbol{U}\boldsymbol{V}_1 = \boldsymbol{V}_2$ and, finally,

$$\boldsymbol{\Sigma}_2^{-1} = \boldsymbol{V}_2^T\boldsymbol{V}_2 = \boldsymbol{V}_1^T\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{V}_1 = \boldsymbol{V}_1^T\boldsymbol{V}_1 = \boldsymbol{\Sigma}_1^{-1}. \tag{27.5}$$

The decision boundary is therefore linear if and only if $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$, which is precisely linear discriminant analysis (LDA).

## Problem 11 (Version A) (4 credits)

a)

No. We cannot conclude anything because "Fairness through Unawareness" does not work. There can be many highly correlated features that are proxies of the sensitive attribute.

b)

First we compute the predictions $R$ to obtain:

| ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| $X$ | 0.5 | -1.0 | -0.5 | 2.0 | 0.5 | 1.5 | 0.1 |
| $A$ | a | b | b | a | b | a | b |
| $R$ | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| $Y$ | 1 | 1 | 0 | 0 | 0 | 0 | 1 |

We see that $1/3$ instances in group $a$ have $R = 1$ vs. $3/4$ instances in group $b$. *Independence* is not satisfied.

For group $a$ we have TP=$1/1$ and FP=$0/2$.
For group $b$ we have TP=$2/2$ and FP=$1/2$.

Since only TP matches for both groups, *Equality of Opportunity* is satisfied and *Separation* is not satisfied.



c)

We modify the instance with ID 1, changing the non-sensitive feature from $X = 0.5$ to $X = 1.5$. Now the prediction changes from $R = 1$ to $R = 0$.

Now we have $0/3$ instances with $R = 1$ within its group, vs. $3/4$ in the other group so *Independence* is still not satisfied. The TP rate has changed from $1/1$ to $0/1$ compared to $2/2$ in the other group, which means that neither *Equality of Opportunity* nor *Separation* are satisfied.

## Problem 11 (Version B) (4 credits)

a)

No. We cannot conclude anything because "Fairness through Unawareness" does not work. There can be many highly correlated features that are proxies of the sensitive attribute.

0
1

b)

First we compute the predictions $R$ to obtain:

| ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| $X$ | 0.5 | -1.0 | -0.5 | 2.0 | 0.5 | 1.5 | 0.1 |
| $A$ | b | a | a | b | a | b | a |
| $R$ | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| $Y$ | 1 | 1 | 0 | 0 | 0 | 0 | 1 |

We see that $3/4$ instances in group $a$ have $R = 1$ vs. $1/3$ instances in group $b$. *Independence* is not satisfied.

For group $a$ we have TP=$2/2$ and FP=$1/2$.
For group $b$ we have TP=$1/1$ and FP=$0/2$.

Since only TP matches for both groups, *Equality of Opportunity* is satisfied and *Separation* is not satisfied.

0
1
2

c)

We modify the instance with ID 1, changing the non-sensitive feature from $X = 0.5$ to $X = 1.5$. Now the prediction changes from $R = 1$ to $R = 0$.

Now we have $0/3$ instances with $R = 1$ within its group, vs. $3/4$ in the other group so *Independence* is still not satisfied. The TP rate has changed from $1/1$ to $0/1$ compared to $2/2$ in the other group, which means that neither *Equality of Opportunity* nor *Separation* are satisfied.

0
1

## Problem 11 (Version C) (4 credits)

0
1

a)

No. We cannot conclude anything because "Fairness through Unawareness" does not work. There can be many highly correlated features that are proxies of the sensitive attribute.

0
1
2

b)

First we compute the predictions $R$ to obtain:

| ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| $X$ | 0.5 | -1.0 | -0.5 | 2.0 | 0.5 | 1.5 | 0.1 |
| $A$ | a | b | b | a | b | a | b |
| $R$ | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| $Y$ | 1 | 1 | 0 | 0 | 0 | 0 | 1 |

We see that $1/3$ instances in group $a$ have $R = 1$ vs. $3/4$ instances in group $b$. *Independence* is not satisfied.

For group $a$ we have TP=$1/1$ and FP=$0/2$.
For group $b$ we have TP=$2/2$ and FP=$1/2$.

Since only TP matches for both groups, *Equality of Opportunity* is satisfied and *Separation* is not satisfied.



0
1

c)

We modify the instance with ID 1, changing the non-sensitive feature from $X = 0.5$ to $X = 1.5$. Now the prediction changes from $R = 1$ to $R = 0$.

Now we have $0/3$ instances with $R = 1$ within its group, vs. $3/4$ in the other group so *Independence* is still not satisfied. The TP rate has changed from $1/1$ to $0/1$ compared to $2/2$ in the other group, which means that neither *Equality of Opportunity* nor *Separation* are satisfied.

## Problem 11 (Version D) (4 credits)

a)

0
1

No. We cannot conclude anything because "Fairness through Unawareness" does not work. There can be many highly correlated features that are proxies of the sensitive attribute.

b)

0
1
2

First we compute the predictions $R$ to obtain:

| ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| $X$ | 0.5 | -1.0 | -0.5 | 2.0 | 0.5 | 1.5 | 0.1 |
| $A$ | b | a | a | b | a | b | a |
| $R$ | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| $Y$ | 1 | 1 | 0 | 0 | 0 | 0 | 1 |

We see that $3/4$ instances in group $a$ have $R = 1$ vs. $1/3$ instances in group $b$. *Independence* is not satisfied.

For group $a$ we have TP=$2/2$ and FP=$1/2$.
For group $b$ we have TP=$1/1$ and FP=$0/2$.

Since only TP matches for both groups, *Equality of Opportunity* is satisfied and *Separation* is not satisfied.

c)

0
1

We modify the instance with ID 1, changing the non-sensitive feature from $X = 0.5$ to $X = 1.5$. Now the prediction changes from $R = 1$ to $R = 0$.

Now we have $0/3$ instances with $R = 1$ within its group, vs. $3/4$ in the other group so *Independence* is still not satisfied. The TP rate has changed from $1/1$ to $0/1$ compared to $2/2$ in the other group, which means that neither *Equality of Opportunity* nor *Separation* are satisfied.

**Additional space for solutions–clearly mark the (sub)problem your answers are related to and strike out invalid solutions.**