Data Analytics and Machine Learning
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.

Maschinelles Lernen

Exam:	IN2064 / Retake exercise	**Date:**	Wednesday 13th April, 2022
Examiner:	Prof. Dr. Stephan Günnemann	**Time:**	08:00 – 10:00

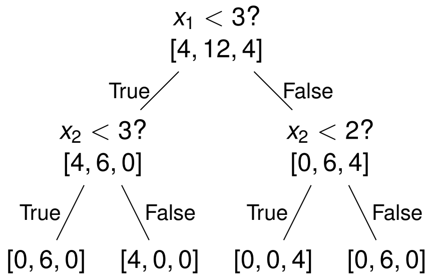
Working instructions

- You have to sign the code of conduct. (Typing your name is fine).
- You have to either print this document, solve the problems on paper and then scan your solutions OR paste scans/pictures of your handwritten, on-paper solutions into the solution boxes in this PDF.
- You must not use any other means of creating a submission (e.g. digital pen on a tablet).
- 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 versions: 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**.
- The last pages (after problem 11) can be used as scratch paper.
- All sheets (save for empty 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.**
- **For problems that say "Justify your answer" you only get points if you provide a valid explanation.**
- **For problems that say "Derive" you only get points if you provide a valid mathematical derivation.**
- **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", "Derive" 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.

Problem 1 (Version A) (8 credits)

0 □
1 □
2 □

a)



b)

□ 0
□ 1
□ 2

Gini index: $i_G(t) = \sum_{i \in C} \pi_i(1 - \pi_i) = 1 - \sum_{i \in C} \pi_i^2$
Root node: $i_G(t) = 1 - \left(\frac{4}{20}\right)^2 - \left(\frac{12}{20}\right)^2 - \left(\frac{4}{20}\right)^2 = \frac{56}{100} = 0.56$
Left child of root: $i_G(t) = 1 - \left(\frac{4}{10}\right)^2 - \left(\frac{6}{10}\right)^2 - \left(\frac{0}{10}\right)^2 = \frac{48}{100} = 0.48$
Right child of root: $i_G(t) = 1 - \left(\frac{0}{10}\right)^2 - \left(\frac{6}{10}\right)^2 - \left(\frac{4}{10}\right)^2 = \frac{48}{100} = 0.48$
Left leaf node: $i_G(t) = 1 - \left(\frac{0}{6}\right)^2 - \left(\frac{6}{6}\right)^2 - \left(\frac{0}{6}\right)^2 = 0$
Left-middle leaf node: $i_G(t) = 1 - \left(\frac{4}{4}\right)^2 - \left(\frac{0}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0$
Right-middle leaf node: $i_G(t) = 1 - \left(\frac{0}{4}\right)^2 - \left(\frac{0}{4}\right)^2 - \left(\frac{4}{4}\right)^2 = 0$
Right leaf node: $i_G(t) = 1 - \left(\frac{0}{6}\right)^2 - \left(\frac{6}{6}\right)^2 - \left(\frac{0}{6}\right)^2 = 0$ (all leaf nodes)

c)

□ 0
□ 1
□ 2

Misclassification rate: $i_E(t) = 1 - \max_c p(y = c|t)$
Root node: $i_E(t) = 1 - \frac{12}{20} = \frac{4}{10}$
Left child of root: $i_E(t) = 1 - \frac{6}{10} = \frac{4}{10}$
Right child of root: $i_E(t) = 1 - \frac{6}{10} = \frac{4}{10}$
Left leaf node: $i_E(t) = 1 - 1 = 0$
Left-middle leaf node: $i_E(t) = 1 - 1 = 0$
Right-middle leaf node: $i_E(t) = 1 - 1 = 0$
Right leaf node: $i_E(t) = 1 - 1 = 0$ (all leaf nodes)

0 □
1 □

d)

Yes, since each node decreases the impurity.

0 □
1 □

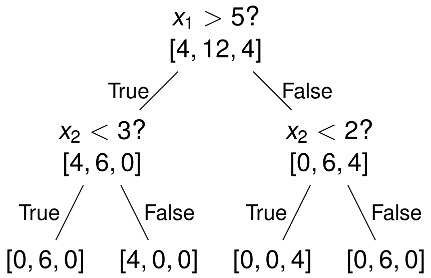
e)

No, because the root node does not decrease impurity, hence it would not have been added by the algorithm.

Problem 1 (Version B) (8 credits)

a)

□ 0
□ 1
□ 2



0 □
1 □
2 □

b)

Gini index: $i_G(t) = \sum_{i \in C} \pi_i(1 - \pi_i) = 1 - \sum_{i \in C} \pi_i^2$

Root node: $i_G(t) = 1 - \left(\frac{4}{20}\right)^2 - \left(\frac{12}{20}\right)^2 - \left(\frac{4}{20}\right)^2 = \frac{56}{100} = 0.56$

Left child of root: $i_G(t) = 1 - \left(\frac{4}{10}\right)^2 - \left(\frac{6}{10}\right)^2 - \left(\frac{0}{10}\right)^2 = \frac{48}{100} = 0.48$

Right child of root: $i_G(t) = 1 - \left(\frac{0}{10}\right)^2 - \left(\frac{6}{10}\right)^2 - \left(\frac{4}{10}\right)^2 = \frac{48}{100} = 0.48$

Left leaf node: $i_G(t) = 1 - \left(\frac{0}{6}\right)^2 - \left(\frac{6}{6}\right)^2 - \left(\frac{0}{6}\right)^2 = 0$

Left-middle leaf node: $i_G(t) = 1 - \left(\frac{4}{4}\right)^2 - \left(\frac{0}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0$

Right-middle leaf node: $i_G(t) = 1 - \left(\frac{0}{4}\right)^2 - \left(\frac{0}{4}\right)^2 - \left(\frac{4}{4}\right)^2 = 0$

Right leaf node: $i_G(t) = 1 - \left(\frac{0}{6}\right)^2 - \left(\frac{6}{6}\right)^2 - \left(\frac{0}{6}\right)^2 = 0$ (all leaf nodes)

0 □
1 □
2 □

c)

Misclassification rate: $i_E(t) = 1 - \max_c p(y = c|t)$

Root node: $i_E(t) = 1 - \frac{12}{20} = \frac{4}{10}$

Left child of root: $i_E(t) = 1 - \frac{6}{10} = \frac{4}{10}$

Right child of root: $i_E(t) = 1 - \frac{6}{10} = \frac{4}{10}$

Left leaf node: $i_E(t) = 1 - 1 = 0$

Left-middle leaf node: $i_E(t) = 1 - 1 = 0$

Right-middle leaf node: $i_E(t) = 1 - 1 = 0$

Right leaf node: $i_E(t) = 1 - 1 = 0$ (all leaf nodes)

d)

□ 0
□ 1

Yes, since each node decreases the impurity.

e)

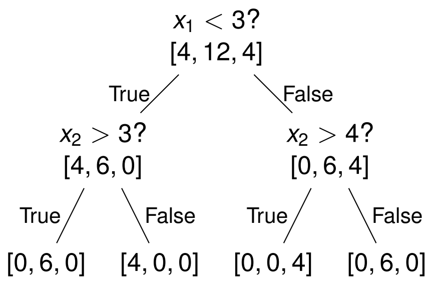
□ 0
□ 1

No, because the root node does not decrease impurity, hence it would not have been added by the algorithm.

Problem 1 (Version C) (8 credits)

0 □
1 □
2 □

a)



b)

□ 0
□ 1
□ 2

Gini index: $i_G(t) = \sum_{i \in C} \pi_i(1 - \pi_i) = 1 - \sum_{i \in C} \pi_i^2$

Root node: $i_G(t) = 1 - \left(\frac{4}{20}\right)^2 - \left(\frac{12}{20}\right)^2 - \left(\frac{4}{20}\right)^2 = \frac{56}{100} = 0.56$

Left child of root: $i_G(t) = 1 - \left(\frac{4}{10}\right)^2 - \left(\frac{6}{10}\right)^2 - \left(\frac{0}{10}\right)^2 = \frac{48}{100} = 0.48$

Right child of root: $i_G(t) = 1 - \left(\frac{0}{10}\right)^2 - \left(\frac{6}{10}\right)^2 - \left(\frac{4}{10}\right)^2 = \frac{48}{100} = 0.48$

Left leaf node: $i_G(t) = 1 - \left(\frac{0}{6}\right)^2 - \left(\frac{6}{6}\right)^2 - \left(\frac{0}{6}\right)^2 = 0$

Left-middle leaf node: $i_G(t) = 1 - \left(\frac{4}{4}\right)^2 - \left(\frac{0}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0$

Right-middle leaf node: $i_G(t) = 1 - \left(\frac{0}{4}\right)^2 - \left(\frac{0}{4}\right)^2 - \left(\frac{4}{4}\right)^2 = 0$

Right leaf node: $i_G(t) = 1 - \left(\frac{0}{6}\right)^2 - \left(\frac{6}{6}\right)^2 - \left(\frac{0}{6}\right)^2 = 0$ (all leaf nodes)

c)

□ 0
□ 1
□ 2

Misclassification rate: $i_E(t) = 1 - \max_c p(y = c|t)$

Root node: $i_E(t) = 1 - \frac{12}{20} = \frac{4}{10}$

Left child of root: $i_E(t) = 1 - \frac{6}{10} = \frac{4}{10}$

Right child of root: $i_E(t) = 1 - \frac{6}{10} = \frac{4}{10}$

Left leaf node: $i_E(t) = 1 - 1 = 0$

Left-middle leaf node: $i_E(t) = 1 - 1 = 0$

Right-middle leaf node: $i_E(t) = 1 - 1 = 0$

Right leaf node: $i_E(t) = 1 - 1 = 0$ (all leaf nodes)

0
1 □
□

d)

Yes, since each node decreases the impurity.

0
1 □
□

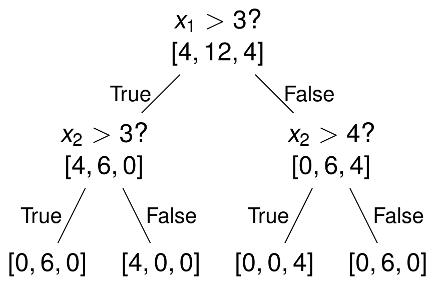
e)

No, because the root node does not decrease impurity, hence it would not have been added by the algorithm.

Problem 1 (Version D) (8 credits)

a)

□ 0
□ 1
□ 2



0 □
1 □
2 □

b)

Gini index: $i_G(t) = \sum_{i \in C} \pi_i(1 - \pi_i) = 1 - \sum_{i \in C} \pi_i^2$

Root node: $i_G(t) = 1 - \left(\frac{4}{20}\right)^2 - \left(\frac{12}{20}\right)^2 - \left(\frac{4}{20}\right)^2 = \frac{56}{100} = 0.56$

Left child of root: $i_G(t) = 1 - \left(\frac{4}{10}\right)^2 - \left(\frac{6}{10}\right)^2 - \left(\frac{0}{10}\right)^2 = \frac{48}{100} = 0.48$

Right child of root: $i_G(t) = 1 - \left(\frac{0}{10}\right)^2 - \left(\frac{6}{10}\right)^2 - \left(\frac{4}{10}\right)^2 = \frac{48}{100} = 0.48$

Left leaf node: $i_G(t) = 1 - \left(\frac{0}{6}\right)^2 - \left(\frac{6}{6}\right)^2 - \left(\frac{0}{6}\right)^2 = 0$

Left-middle leaf node: $i_G(t) = 1 - \left(\frac{4}{4}\right)^2 - \left(\frac{0}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0$

Right-middle leaf node: $i_G(t) = 1 - \left(\frac{0}{4}\right)^2 - \left(\frac{0}{4}\right)^2 - \left(\frac{4}{4}\right)^2 = 0$

Right leaf node: $i_G(t) = 1 - \left(\frac{0}{6}\right)^2 - \left(\frac{6}{6}\right)^2 - \left(\frac{0}{6}\right)^2 = 0$ (all leaf nodes)

0 □
1 □
2 □

c)

Misclassification rate: $i_E(t) = 1 - \max_c p(y = c|t)$

Root node: $i_E(t) = 1 - \frac{12}{20} = \frac{4}{10}$

Left child of root: $i_E(t) = 1 - \frac{6}{10} = \frac{4}{10}$

Right child of root: $i_E(t) = 1 - \frac{6}{10} = \frac{4}{10}$

Left leaf node: $i_E(t) = 1 - 1 = 0$

Left-middle leaf node: $i_E(t) = 1 - 1 = 0$

Right-middle leaf node: $i_E(t) = 1 - 1 = 0$

Right leaf node: $i_E(t) = 1 - 1 = 0$ (all leaf nodes)

d)

☐ 0
☐ 1

Yes, since each node decreases the impurity.

e)

☐ 0
☐ 1

No, because the root node does not decrease impurity, hence it would not have been added by the algorithm.

Problem 2 (Version A) (8 credits)

0 □
1 □
2 □
3 □
4 □
5 □
6 □
7 □
8 □

We compute the logarithm of the posterior distibution:

$$
\begin{aligned}
\log \mathbb{P}(\theta \mid \{x_1, ..., x_N\}, \alpha) &= \sum_{i=1}^{N} \log \mathbb{P}(x_i \mid \theta) + \log \mathbb{P}(\theta \mid \alpha) + D \\
&= \sum_{i=1}^{N} \sum_{c=1}^{C} \mathbb{I}(x_i = c) \log \theta_c + \sum_{c=1}^{C} (\alpha_c - 1) \log \theta_c - \log B(\alpha) + D \\
&= \sum_{c=1}^{C} N_c \log \theta_c + \sum_{c=1}^{C} (\alpha_c - 1) \log \theta_c - \log B(\alpha) + D \\
&= \sum_{c=1}^{C} (N_c + \alpha_c - 1) \log \theta_c + D',
\end{aligned}
$$

where $B, D, D' \in \mathbb{R}$ are constant w.r.t. θ and can be ignored in our optimization problem. We use the normalization constraint $\sum_{c=1}^{C} \theta_c = 1$ to compute the Lagrangian:

$$
L = \sum_{c=1}^{C} (N_c + \alpha_c - 1) \log \theta_c + \lambda(1 - \sum_{c=1}^{C} \theta_c)
$$

with $\lambda \in \mathbb{R}$.
We set the derivative to 0:

$$
\begin{aligned}
\frac{\partial L}{\partial \theta_c} = \frac{(N_c + \alpha_c - 1)}{\theta_c} - \lambda = 0 \\
\frac{(N_c + \alpha_c - 1)}{\lambda} = \theta_c
\end{aligned}
$$

By using again $\sum_{c=1}^{C} \theta_c = 1$, we get $\lambda = \sum_c (N_c + \alpha_c - 1)$ and finally $\theta_{\text{MAP},c} = \frac{N_c + \alpha_c - 1}{\sum_{c'} (N_{c'} + \alpha_{c'} - 1)}$.

Problem 2 (Version B) (8 credits)

□ 0
□ 1
□ 2
□ 3
□ 4
□ 5
□ 6
□ 7
□ 8

We compute the logarithm of the posterior distibution:

$$\begin{aligned}
\log \mathbb{P}(\theta \mid \{x_1, ..., x_N\}, \alpha) &= \sum_{i=1}^{N} \log \mathbb{P}(x_i \mid \theta) + \log \mathbb{P}(\theta \mid \alpha) + D \\
&= \sum_{i=1}^{N} \sum_{c=1}^{C} \mathbb{I}(x_i = c) \log \theta_c + \sum_{c=1}^{C} (\alpha_c - 1) \log \theta_c - \log B(\alpha) + D \\
&= \sum_{c=1}^{C} N_c \log \theta_c + \sum_{c=1}^{C} (\alpha_c - 1) \log \theta_c - \log B(\alpha) + D \\
&= \sum_{c=1}^{C} (N_c + \alpha_c - 1) \log \theta_c + D',
\end{aligned}$$

where $B, D, D' \in \mathbb{R}$ are constant w.r.t. θ and can be ignored in our optimization problem.
We use the normalization constraint $\sum_{c=1}^{C} \theta_c = 1$ to compute the Lagrangian:

$$L = \sum_{c=1}^{C} (N_c + \alpha_c - 1) \log \theta_c + \lambda(1 - \sum_{c=1}^{C} \theta_c)$$

with $\lambda \in \mathbb{R}$.
We set the derivative to 0:

$$\begin{aligned}
\frac{\partial L}{\partial \theta_c} = \frac{(N_c + \alpha_c - 1)}{\theta_c} - \lambda = 0 \\
\frac{(N_c + \alpha_c - 1)}{\lambda} = \theta_c
\end{aligned}$$

By using again $\sum_{c=1}^{C} \theta_c = 1$, we get $\lambda = \sum_c (N_c + \alpha_c - 1)$ and finally $\theta_{\text{MAP},c} = \frac{N_c + \alpha_c - 1}{\sum_{c'} (N_{c'} + \alpha_{c'} - 1)}$.

Problem 2 (Version C) (8 credits)

0 □
1 □
2 □
3 □
4 □
5 □
6 □
7 □
8 □

We compute the logarithm of the posterior distibution:

$$
\begin{aligned}
\log \mathbb{P}(\theta \mid \{x_1, ..., x_N\}, \alpha) &= \sum_{i=1}^{N} \log \mathbb{P}(x_i \mid \theta) + \log \mathbb{P}(\theta \mid \alpha) + D \\
&= \sum_{i=1}^{N} \sum_{c=1}^{C} \mathbb{I}(x_i = c) \log \theta_c + \sum_{c=1}^{C} (\alpha_c - 1) \log \theta_c - \log B(\alpha) + D \\
&= \sum_{c=1}^{C} N_c \log \theta_c + \sum_{c=1}^{C} (\alpha_c - 1) \log \theta_c - \log B(\alpha) + D \\
&= \sum_{c=1}^{C} (N_c + \alpha_c - 1) \log \theta_c + D',
\end{aligned}
$$

where $B, D, D' \in \mathbb{R}$ are constant w.r.t. θ and can be ignored in our optimization problem.
We use the normalization constraint $\sum_{c=1}^{C} \theta_c = 1$ to compute the Lagrangian:

$$
L = \sum_{c=1}^{C} (N_c + \alpha_c - 1) \log \theta_c + \lambda (1 - \sum_{c=1}^{C} \theta_c)
$$

with $\lambda \in \mathbb{R}$.
We set the derivative to 0:

$$
\begin{aligned}
\frac{\partial L}{\partial \theta_c} = \frac{(N_c + \alpha_c - 1)}{\theta_c} - \lambda = 0 \\
\frac{(N_c + \alpha_c - 1)}{\lambda} = \theta_c
\end{aligned}
$$

By using again $\sum_{c=1}^{C} \theta_c = 1$, we get $\lambda = \sum_c (N_c + \alpha_c - 1)$ and finally $\theta_{\text{MAP},c} = \frac{N_c + \alpha_c - 1}{\sum_{c'} (N_{c'} + \alpha_{c'} - 1)}$.

Problem 2 (Version D) (8 credits)

□ 0
□ 1
□ 2
□ 3
□ 4
□ 5
□ 6
□ 7
□ 8

We compute the logarithm of the posterior distibution:

$$
\begin{aligned}
\log \mathbb{P}(\theta \mid \{x_1, ..., x_N\}, \alpha) &= \sum_{i=1}^{N} \log \mathbb{P}(x_i \mid \theta) + \log \mathbb{P}(\theta \mid \alpha) + D \\
&= \sum_{i=1}^{N} \sum_{c=1}^{C} \mathbb{I}(x_i = c) \log \theta_c + \sum_{c=1}^{C} (\alpha_c - 1) \log \theta_c - \log B(\alpha) + D \\
&= \sum_{c=1}^{C} N_c \log \theta_c + \sum_{c=1}^{C} (\alpha_c - 1) \log \theta_c - \log B(\alpha) + D \\
&= \sum_{c=1}^{C} (N_c + \alpha_c - 1) \log \theta_c + D',
\end{aligned}
$$

where $B, D, D' \in \mathbb{R}$ are constant w.r.t. θ and can be ignored in our optimization problem.
We use the normalization constraint $\sum_{c=1}^{C} \theta_c = 1$ to compute the Lagrangian:

$$
L = \sum_{c=1}^{C} (N_c + \alpha_c - 1) \log \theta_c + \lambda (1 - \sum_{c=1}^{C} \theta_c)
$$

with $\lambda \in \mathbb{R}$.
We set the derivative to 0:

$$
\begin{aligned}
\frac{\partial L}{\partial \theta_c} = \frac{(N_c + \alpha_c - 1)}{\theta_c} - \lambda = 0 \\
\frac{(N_c + \alpha_c - 1)}{\lambda} = \theta_c
\end{aligned}
$$

By using again $\sum_{c=1}^{C} \theta_c = 1$, we get $\lambda = \sum_c (N_c + \alpha_c - 1)$ and finally $\theta_{\text{MAP},c} = \frac{N_c + \alpha_c - 1}{\sum_{c'} (N_{c'} + \alpha_{c'} - 1)}$.

Problem 3 (Version A) (8 credits)

0 □
1 □
2 □
3 □
4 □

a)

Yes, the problem is linear wrt the weights $\mathbf{W}^{(L)}$.By setting $\Phi = \sigma(\mathbf{W}^{(L-1)} \dots (\sigma(\mathbf{W}^{(0)}\mathbf{x})))$,the optimal solution is $\mathbf{w}_L = (\Phi^T\Phi)^{-1}\Phi\mathbf{y}$.

0 □
1 □
2 □
3 □
4 □

b)

No, the problem is highly non-linear wrt the weights $\mathbf{W}^{(0)}$.Note that we cannot invert the network either, since $D > 1$ and the weight matrices are non-zero. We could approximate the optimal solution by optimizing with SGD.

Problem 3 (Version B) (8 credits)

a)

□ 0
□ 1
□ 2
□ 3
□ 4

Yes, the problem is linear wrt the weights $\mathbf{W}^{(L)}$.By setting $\Phi = \sigma(\mathbf{W}^{(L-1)} \ldots (\sigma(\mathbf{W}^{(0)}\mathbf{x})))$,the optimal solution is $\mathbf{w}_L = (\Phi^T\Phi)^{-1}\Phi\mathbf{y}$.

b)

□ 0
□ 1
□ 2
□ 3
□ 4

No, the problem is highly non-linear wrt the weights $\mathbf{W}^{(0)}$.Note that we cannot invert the network either, since $D > 1$ and the weight matrices are non-zero. We could approximate the optimal solution by optimizing with SGD.

Problem 3 (Version C) (8 credits)

0 □
1 □
2 □
3 □
4 □

a)

Yes, the problem is linear wrt the weights $\mathbf{W}^{(L)}$.By setting $\Phi = \sigma(\mathbf{W}^{(L-1)} \dots (\sigma(\mathbf{W}^{(0)}\mathbf{x})))$,the optimal solution is $\mathbf{w}_L = (\Phi^T\Phi)^{-1}\Phi\mathbf{y}$.

0 □
1 □
2 □
3 □
4 □

b)

No, the problem is highly non-linear wrt the weights $\mathbf{W}^{(0)}$.Note that we cannot invert the network either, since $D > 1$ and the weight matrices are non-zero. We could approximate the optimal solution by optimizing with SGD.

Problem 3 (Version D) (8 credits)

a)

□ 0
□ 1
□ 2
□ 3
□ 4

Yes, the problem is linear wrt the weights $\mathbf{W}^{(L)}$.By setting $\Phi = \sigma(\mathbf{W}^{(L-1)} \ldots (\sigma(\mathbf{W}^{(0)}\mathbf{x})))$,the optimal solution is $\mathbf{w}_L = (\Phi^T\Phi)^{-1}\Phi\mathbf{y}$.

b)

□ 0
□ 1
□ 2
□ 3
□ 4

No, the problem is highly non-linear wrt the weights $\mathbf{W}^{(0)}$.Note that we cannot invert the network either, since $D > 1$ and the weight matrices are non-zero. We could approximate the optimal solution by optimizing with SGD.

Problem 4 (Version A) (9 credits)

0 □
1 □
2 □

a)

(1) We have missing data and (2) logistic regression treats the data as $\mathbb{R}^5$ and Gaussian distributed.

0 □
1 □
2 □
3 □
4 □

b)

1. (d) because we model the density of a probability in [0, 1].
2. (b) because the values are continuous (and likely normal distributed throughout the summer/year).
3. (c) because we have integer values (and are considering a limited region as well as time span).
4. (a) because the data is binary.
5. (a) because the data is binary.

0 □
1 □
2 □
3 □

c)

Possibilities:

- The features are likely correlated (e.g. “# Visitors at 9 am”, “Public holiday”, and “Weekday”). However, Naïve Bayes assumes conditional independence. In other words, the same information is taken into consideration for each feature independently / multiple times. Hence, the very low temperature is overruled by the other features.
- The data is not “sampled i.i.d.” E.g. the model does not incorporate the preference of your friend not going to the lake on a rainy day.
- Data distribution shift. The weather changes and potentially how many potential buyers are at the lake at certain conditions.

Problem 4 (Version B) (9 credits)

a)

□ 0
□ 1
□ 2

(1) We have missing data and (2) logistic regression treats the data as $\mathbb{R}^5$ and Gaussian distributed.

b)

□ 0
□ 1
□ 2
□ 3
□ 4

1. (d) because we model the density of a probability in $[0, 1]$.
2. (b) because the values are continuous (and likely normal distributed throughout the summer/year).
3. (c) because we have integer values (and are considering a limited region as well as time span).
4. (a) because the data is binary.
5. (a) because the data is binary.

c)

□ 0
□ 1
□ 2
□ 3

Possibilities:

- The features are likely correlated (e.g. “# Visitors at 9 am”, “Public holiday”, and “Weekday”). However, Naïve Bayes assumes conditional independence. In other words, the same information is taken into consideration for each feature independently / multiple times. Hence, the very low temperature is overruled by the other features.
- The data is not “sampled i.i.d.” E.g. the model does not incorporate the preference of your friend not going to the lake on a rainy day.
- Data distribution shift. The weather changes and potentially how many potential buyers are at the lake at certain conditions.

Problem 4 (Version C) (9 credits)

0 □
1 □
2 □

a)

(1) We have missing data and (2) logistic regression treats the data as $\mathbb{R}^5$ and Gaussian distributed.

0 □
1 □
2 □
3 □
4 □

b)

1. (d) because we model the density of a probability in [0, 1].
2. (b) because the values are continuous (and likely normal distributed throughout the summer/year).
3. (c) because we have integer values (and are considering a limited region as well as time span).
4. (a) because the data is binary.
5. (a) because the data is binary.

0 □
1 □
2 □
3 □

c)

Possibilities:

- The features are likely correlated (e.g. “# Visitors at 9 am”, “Public holiday”, and “Weekday”). However, Naïve Bayes assumes conditional independence. In other words, the same information is taken into consideration for each feature independently / multiple times. Hence, the very low temperature is overruled by the other features.
- The data is not “sampled i.i.d.” E.g. the model does not incorporate the preference of your friend not going to the lake on a rainy day.
- Data distribution shift. The weather changes and potentially how many potential buyers are at the lake at certain conditions.

Problem 4 (Version D) (9 credits)

a)

□ 0
□ 1
□ 2

(1) We have missing data and (2) logistic regression treats the data as $\mathbb{R}^5$ and Gaussian distributed.

b)

□ 0
□ 1
□ 2
□ 3
□ 4

1. (d) because we model the density of a probability in $[0, 1]$.
2. (b) because the values are continuous (and likely normal distributed throughout the summer/year).
3. (c) because we have integer values (and are considering a limited region as well as time span).
4. (a) because the data is binary.
5. (a) because the data is binary.

c)

□ 0
□ 1
□ 2
□ 3

Possibilities:

- The features are likely correlated (e.g. "# Visitors at 9 am", "Public holiday", and "Weekday"). However, Naïve Bayes assumes conditional independence. In other words, the same information is taken into consideration for each feature independently / multiple times. Hence, the very low temperature is overruled by the other features.
- The data is not "sampled i.i.d." E.g. the model does not incorporate the preference of your friend not going to the lake on a rainy day.
- Data distribution shift. The weather changes and potentially how many potential buyers are at the lake at certain conditions.

Problem 5 (Version A) (8 credits)

0 □
1 □
2 □
3 □
4 □

a)

$f_a(\mathbf{x})$ is convex if $(\mathbf{x}-\mathbf{a})^\top\mathbf{A}(\mathbf{x}-\mathbf{a})$ is convex(rule 2).
$(\mathbf{x}-\mathbf{a})^\top\mathbf{A}(\mathbf{x}-\mathbf{a}) = \mathbf{x}^\top\mathbf{A}\mathbf{x} - 2\mathbf{x}^\top\mathbf{A}\mathbf{a} + \mathbf{a}^\top\mathbf{A}\mathbf{a}$.
$-2\mathbf{x}^\top\mathbf{A}\mathbf{a} + \mathbf{a}^\top\mathbf{A}\mathbf{a}$ is linear and, therefore, convex in $\mathbf{x}$.
$\mathbf{x}^\top\mathbf{A}\mathbf{x}$ is convex since $\nabla^2_{\mathbf{x}}[\mathbf{x}^\top\mathbf{A}\mathbf{x}] = 2\mathbf{A}$ and the fact that $\mathbf{A}$ is positive semidefinite.
Last, $\mathbf{x}^\top\mathbf{A}\mathbf{x} - 2\mathbf{x}^\top\mathbf{A}\mathbf{a} + \mathbf{a}^\top\mathbf{A}\mathbf{a}$ is a sum of convex functions and, thus, also convex(rule 1).
In conclusion, $f_a(\mathbf{x})$ is convex.(if the argumentation is correct)

0 □
1 □
2 □
3 □
4 □

b)

Since 2^x is a convex and nondecreasing function, $f_b((\mathbf{x})$ is convex if $h(\mathbf{x}) = \max_i |\mathbf{x}_i - \mathbf{a}_i|$ is convex(rule 6).
Consider the function $e_i(\mathbf{x}) = x_i$ which is clearly convex in $\mathbf{x}$ since it is constant in all dimensions but the i-th, in which it is linear. From the definition of convexity, it is easy to see that $e_i(\lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \lambda e_i(\boldsymbol{x}) + (1-\lambda)e_i(\boldsymbol{y})$ holds.
Moreover, $|x|$ is convex but *not nondecreasing* (i.e. rule 6 does not apply). However, from rule 5 it follows that $|x-a|$ is convex in $x \in \mathbb{R}$for any $a \in \mathbb{R}$.
Using rule 2, we can conclude that also the maximum in $\max_i |\mathbf{x}_i - \mathbf{a}_i|$ is convex.
We conclude that $f_b((\mathbf{x})$ is convex.(if the argumentation is correct)

Problem 5 (Version B) (8 credits)

a)

□ 0
□ 1
□ 2
□ 3
□ 4

$f_a(\mathbf{x})$ is convex if $(\mathbf{x}-\mathbf{a})^\top\mathbf{A}(\mathbf{x}-\mathbf{a})$ is convex(rule 2).
$(\mathbf{x}-\mathbf{a})^\top\mathbf{A}(\mathbf{x}-\mathbf{a}) = \mathbf{x}^\top\mathbf{A}\mathbf{x} - 2\mathbf{x}^\top\mathbf{A}\mathbf{a} + \mathbf{a}^\top\mathbf{A}\mathbf{a}$.
$-2\mathbf{x}^\top\mathbf{A}\mathbf{a} + \mathbf{a}^\top\mathbf{A}\mathbf{a}$ is linear and, therefore, convex in $\mathbf{x}$.
$\mathbf{x}^\top\mathbf{A}\mathbf{x}$ is convex since $\nabla^2_{\mathbf{x}}[\mathbf{x}^\top\mathbf{A}\mathbf{x}] = 2\mathbf{A}$ and the fact that $\mathbf{A}$ is positive semidefinite.
Last, $\mathbf{x}^\top\mathbf{A}\mathbf{x} - 2\mathbf{x}^\top\mathbf{A}\mathbf{a} + \mathbf{a}^\top\mathbf{A}\mathbf{a}$ is a sum of convex functions and, thus, also convex(rule 1).
In conclusion, $f_a(\mathbf{x})$ is convex.(if the argumentation is correct)

b)

□ 0
□ 1
□ 2
□ 3
□ 4

Since 2^x is a convex and nondecreasing function, $f_b((\mathbf{x})$ is convex if $h(\mathbf{x}) = \max_i |\mathbf{x}_i - \mathbf{a}_i|$ is convex(rule 6).
Consider the function $e_i(\boldsymbol{x}) = x_i$ which is clearly convex in $\mathbf{x}$ since it is constant in all dimensions but the i-th, in which it is linear. From the definition of convexity, it is easy to see that $e_i(\lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \lambda e_i(\boldsymbol{x}) + (1-\lambda)e_i(\boldsymbol{y})$ holds.
Moreover, $|x|$ is convex but *not nondecreasing* (i.e. rule 6 does not apply). However, from rule 5 it follows that $|x-a|$ is convex in $x \in \mathbb{R}$for any $a \in \mathbb{R}$.
Using rule 2, we can conclude that also the maximum in $\max_i |\mathbf{x}_i - \mathbf{a}_i|$ is convex.
We conclude that $f_b((\mathbf{x})$ is convex.(if the argumentation is correct)

Problem 5 (Version C) (8 credits)

0 □ 1 □ 2 □ 3 □ 4 □

a)

$f_a(\mathbf{x})$ is convex if $(\mathbf{x}-\mathbf{a})^\top \mathbf{A}(\mathbf{x}-\mathbf{a})$ is convex(rule 2).

$(\mathbf{x}-\mathbf{a})^\top \mathbf{A}(\mathbf{x}-\mathbf{a}) = \mathbf{x}^\top \mathbf{A}\mathbf{x} - 2\mathbf{x}^\top \mathbf{A}\mathbf{a} + \mathbf{a}^\top \mathbf{A}\mathbf{a}$.

$-2\mathbf{x}^\top \mathbf{A}\mathbf{a} + \mathbf{a}^\top \mathbf{A}\mathbf{a}$ is linear and, therefore, convex in **x**.

$\mathbf{x}^\top \mathbf{A}\mathbf{x}$ is convex since $\nabla^2_{\mathbf{x}}[\mathbf{x}^\top \mathbf{A}\mathbf{x}] = 2\mathbf{A}$ and the fact that **A** is positive semidefinite.

Last, $\mathbf{x}^\top \mathbf{A}\mathbf{x} - 2\mathbf{x}^\top \mathbf{A}\mathbf{a} + \mathbf{a}^\top \mathbf{A}\mathbf{a}$ is a sum of convex functions and, thus, also convex(rule 1).

In conclusion, $f_a(\mathbf{x})$ is convex.(if the argumentation is correct)

0 □ 1 □ 2 □ 3 □ 4 □

b)

Since 2^x is a convex and nondecreasing function, $f_b((\mathbf{x})$ is convex if $h(\mathbf{x}) = \max_i |\mathbf{x}_i - \mathbf{a}_i|$ is convex(rule 6).

Consider the function $e_i(\mathbf{x}) = x_i$ which is clearly convex in **x** since it is constant in all dimensions but the i-th, in which it is linear. From the definition of convexity, it is easy to see that $e_i(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \lambda e_i(\boldsymbol{x}) + (1-\lambda)e_i(\boldsymbol{y})$ holds.

Moreover, $|x|$ is convex but *not nondecreasing* (i.e. rule 6 does not apply). However, from rule 5 it follows that $|x-a|$ is convex in $x \in \mathbb{R}$for any $a \in \mathbb{R}$.

Using rule 2, we can conclude that also the maximum in $\max_i |\mathbf{x}_i - \mathbf{a}_i|$ is convex.

We conclude that $f_b((\mathbf{x})$ is convex.(if the argumentation is correct)

Problem 5 (Version D) (8 credits)

a)

□ 0
□ 1
□ 2
□ 3
□ 4

$f_a(\mathbf{x})$ is convex if $(\mathbf{x}-\mathbf{a})^\top\mathbf{A}(\mathbf{x}-\mathbf{a})$ is convex(rule 2).
$(\mathbf{x}-\mathbf{a})^\top\mathbf{A}(\mathbf{x}-\mathbf{a}) = \mathbf{x}^\top\mathbf{A}\mathbf{x} - 2\mathbf{x}^\top\mathbf{A}\mathbf{a} + \mathbf{a}^\top\mathbf{A}\mathbf{a}$.
$-2\mathbf{x}^\top\mathbf{A}\mathbf{a} + \mathbf{a}^\top\mathbf{A}\mathbf{a}$ is linear and, therefore, convex in $\mathbf{x}$.
$\mathbf{x}^\top\mathbf{A}\mathbf{x}$ is convex since $\nabla^2_{\mathbf{x}}[\mathbf{x}^\top\mathbf{A}\mathbf{x}] = 2\mathbf{A}$ and the fact that $\mathbf{A}$ is positive semidefinite.
Last, $\mathbf{x}^\top\mathbf{A}\mathbf{x} - 2\mathbf{x}^\top\mathbf{A}\mathbf{a} + \mathbf{a}^\top\mathbf{A}\mathbf{a}$ is a sum of convex functions and, thus, also convex(rule 1).
In conclusion, $f_a(\mathbf{x})$ is convex.(if the argumentation is correct)

b)

□ 0
□ 1
□ 2
□ 3
□ 4

Since 2^x is a convex and nondecreasing function, $f_b((\mathbf{x})$ is convex if $h(\mathbf{x}) = \max_i |\mathbf{x}_i - \mathbf{a}_i|$ is convex(rule 6).
Consider the function $e_i(\mathbf{x}) = x_i$ which is clearly convex in $\mathbf{x}$ since it is constant in all dimensions but the i-th, in which it is linear. From the definition of convexity, it is easy to see that $e_i(\lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \lambda e_i(\boldsymbol{x}) + (1-\lambda)e_i(\boldsymbol{y})$ holds.
Moreover, $|x|$ is convex but *not nondecreasing* (i.e. rule 6 does not apply). However, from rule 5 it follows that $|x-a|$ is convex in $x \in \mathbb{R}$for any $a \in \mathbb{R}$.
Using rule 2, we can conclude that also the maximum in $\max_i |\mathbf{x}_i - \mathbf{a}_i|$ is convex.
We conclude that $f_b((\mathbf{x})$ is convex.(if the argumentation is correct)

Problem 6 (Version A) (8 credits)

0 □
1 □
2 □
3 □
4 □
5 □
6 □
7 □
8 □

1. (c)
2. (b)
3. (a)
4. (d)

Plot (c) is the only one where the error doesn't go to zero. The minimum is 1. This corresponds to the $f(x) = \sigma(\alpha x + \beta)$ model since here $f(x)$ can be at most 1 due to the sigmoid. Given the true value $y = 2$, this exactly corresponds to the error $E = 1$.

Plot (a) is symmetric and has minimum when α is large and β small, and vice versa. We can already rule out $f(x) = \sigma(\alpha x) + \beta$ model because of symmetry. We can visually see that the minimum is around $(3, -1)$ which would correspond to $f(x) = \alpha x + \beta$.

Plot (b) is again symmetric so it corresponds to the remaining model without sigmoid $f(x) = \alpha\beta x$. We can also check the contour that goes through $(2, 1)$ and $(1, 2)$ minimizes the function. The value at $(0, 0)$ has some positive error as expected. The error drops again as we go to negative values which makes sense given we multiply α and β.

Plot (d) is the only one remaining so it's $f(x) = \sigma(\alpha x) + \beta$ model. We can arrive to this conclusion by seeing there is not symmetry *and* we unlike (c) it reaches 0 error. We can also check some of the values and the contour lines to convince ourselves.

Problem 6 (Version B) (8 credits)

□ 0
□ 1
□ 2
□ 3
□ 4
□ 5
□ 6
□ 7
□ 8

1. (d)
2. (b)
3. (a)
4. (c)

Plot (c) is the only one where the error doesn't go to zero. The minimum is 1. This corresponds to the $f(x) = \sigma(\alpha x + \beta)$ model since here $f(x)$ can be at most 1 due to the sigmoid. Given the true value $y = 2$, this exactly corresponds to the error $E = 1$.

Plot (a) is symmetric and has minimum when α is large and β small, and vice versa. We can already rule out $f(x) = \sigma(\alpha x) + \beta$ model because of symmetry. We can visually see that the minimum is around $(3, -1)$ which would correspond to $f(x) = \alpha x + \beta$.

Plot (b) is again symmetric so it corresponds to the remaining model without sigmoid $f(x) = \alpha\beta x$. We can also check the contour that goes through $(2, 1)$ and $(1, 2)$ minimizes the function. The value at $(0, 0)$ has some positive error as expected. The error drops again as we go to negative values which makes sense given we multiply α and β.

Plot (d) is the only one remaining so it's $f(x) = \sigma(\alpha x) + \beta$ model. We can arrive to this conclusion by seeing there is not symmetry *and* we unlike (c) it reaches 0 error. We can also check some of the values and the contour lines to convince ourselves.

Problem 6 (Version C) (8 credits)

0 □
1 □
2 □
3 □
4 □
5 □
6 □
7 □
8 □

1. (c)
2. (d)
3. (b)
4. (a)

Plot (c) is the only one where the error doesn't go to zero. The minimum is 1. This corresponds to the $f(x) = \sigma(\alpha x + \beta)$ model since here $f(x)$ can be at most 1 due to the sigmoid. Given the true value $y = 2$, this exactly corresponds to the error $E = 1$.
Plot (a) is symmetric and has minimum when α is large and β small, and vice versa. We can already rule out $f(x) = \sigma(\alpha x) + \beta$ model because of symmetry. We can visually see that the minimum is around $(3, -1)$ which would correspond to $f(x) = \alpha x + \beta$.
Plot (b) is again symmetric so it corresponds to the remaining model without sigmoid $f(x) = \alpha\beta x$. We can also check the contour that goes through $(2, 1)$ and $(1, 2)$ minimizes the function. The value at $(0, 0)$ has some positive error as expected. The error drops again as we go to negative values which makes sense given we multiply α and β.
Plot (d) is the only one remaining so it's $f(x) = \sigma(\alpha x) + \beta$ model. We can arrive to this conclusion by seeing there is not symmetry *and* we unlike (c) it reaches 0 error. We can also check some of the values and the contour lines to convince ourselves.

Problem 6 (Version D) (8 credits)

□ 0
□ 1
□ 2
□ 3
□ 4
□ 5
□ 6
□ 7
□ 8

1. (c)
2. (a)
3. (d)
4. (b)

Plot (c) is the only one where the error doesn't go to zero. The minimum is 1. This corresponds to the $f(x) = \sigma(\alpha x + \beta)$ model since here $f(x)$ can be at most 1 due to the sigmoid. Given the true value $y = 2$, this exactly corresponds to the error $E = 1$.

Plot (a) is symmetric and has minimum when α is large and β small, and vice versa. We can already rule out $f(x) = \sigma(\alpha x) + \beta$ model because of symmetry. We can visually see that the minimum is around $(3, -1)$ which would correspond to $f(x) = \alpha x + \beta$.

Plot (b) is again symmetric so it corresponds to the remaining model without sigmoid $f(x) = \alpha\beta x$. We can also check the contour that goes through $(2, 1)$ and $(1, 2)$ minimizes the function. The value at $(0, 0)$ has some positive error as expected. The error drops again as we go to negative values which makes sense given we multiply α and β.

Plot (d) is the only one remaining so it's $f(x) = \sigma(\alpha x) + \beta$ model. We can arrive to this conclusion by seeing there is not symmetry *and* we unlike (c) it reaches 0 error. We can also check some of the values and the contour lines to convince ourselves.

Problem 7 (Version A) (4 credits)

0 □
1 □
2 □
3 □
4 □

1. (a)
2. (c)
3. (b)
4. (d)

Plots in (a) and (b) clearly belong to the logistic regression and linear SVM model since the boundary is linear. Plot in (a) separates more points than the plot in (b) and it has smaller margin meaning (a) belongs to `LogisticRegression(lambda=1.0, penalty=’l2’)` which is the same as using linear SVM with C=1, and (b) is `SVM(C=0.01, kernel=’linear’)`. Same reasoning can be applied to rbf SVM models. The one with higher C will try to classify more points correctly. Thus, (c) is `SVM(C=0.01, kernel=’rbf’)` and (d) `SVM(C=1000, kernel=’rbf’)`.

Problem 7 (Version B) (4 credits)

□ 0
□ 1
□ 2
□ 3
□ 4

1. (c)
2. (a)
3. (d)
4. (b)

Plots in (a) and (b) clearly belong to the logistic regression and linear SVM model since the boundary is linear. Plot in (a) separates more points than the plot in (b) and it has smaller margin meaning (a) belongs to `LogisticRegression(lambda=1.0, penalty='l2')` which is the same as using linear SVM with C=1, and (b) is `SVM(C=0.01, kernel='linear')`. Same reasoning can be applied to rbf SVM models. The one with higher C will try to classify more points correctly. Thus, (c) is `SVM(C=0.01, kernel='rbf')` and (d) `SVM(C=1000, kernel='rbf')`.

Problem 7 (Version C) (4 credits)

0 □
1 □
2 □
3 □
4 □

1. (d)
2. (b)
3. (a)
4. (c)

Plots in (a) and (b) clearly belong to the logistic regression and linear SVM model since the boundary is linear. Plot in (a) separates more points than the plot in (b) and it has smaller margin meaning (a) belongs to `LogisticRegression(lambda=1.0, penalty=’l2’)` which is the same as using linear SVM with C=1, and (b) is `SVM(C=0.01, kernel=’linear’)`. Same reasoning can be applied to rbf SVM models. The one with higher C will try to classify more points correctly. Thus, (c) is `SVM(C=0.01, kernel=’rbf’)` and (d) `SVM(C=1000, kernel=’rbf’)`.

Problem 7 (Version D) (4 credits)

□ 0
□ 1
□ 2
□ 3
□ 4

1. (a)
2. (c)
3. (d)
4. (b)

Plots in (a) and (b) clearly belong to the logistic regression and linear SVM model since the boundary is linear. Plot in (a) separates more points than the plot in (b) and it has smaller margin meaning (a) belongs to `LogisticRegression(lambda=1.0, penalty=’l2’)` which is the same as using linear SVM with C=1, and (b) is `SVM(C=0.01, kernel=’linear’)`. Same reasoning can be applied to rbf SVM models. The one with higher C will try to classify more points correctly. Thus, (c) is `SVM(C=0.01, kernel=’rbf’)` and (d) `SVM(C=1000, kernel=’rbf’)`.

Problem 8 (Version A) (8 credits)

0 □
1 □
2 □
3 □
4 □

a)

$k(\mathbf{x},\mathbf{y}) = \sum_i^M f(\mathbf{x})_i + \sum_i^M f(\mathbf{y})_i$
Function f together with the summation $\sum$ can be viewed as a feature map. Then we just need to check if $k(x, y) = x + y$ is a valid kernel. The kernel matrix has to be positive semidefinite. If we take the 2×2 matrix for two data points x and y, it is easy to show that the kernel matrix is not positive semidefinite since the determinant is always negative or zero.
$\det\left(\begin{bmatrix} x+x & x+y \\ x+y & y+y \end{bmatrix}\right) = 4xy - (x+y)^2 = -(x-y)^2 \leq 0$

0 □
1 □
2 □
3 □
4 □

b)

$k(\mathbf{x},\mathbf{y}) = \log p(\mathbf{x}) \log p(\mathbf{y})$
The probability density function is a feature map $p : \mathbb{R}^N \to \mathbb{R}$.
Same goes for the logarithm $\log : \mathbb{R} \to \mathbb{R}$.
That leaves us with the multiplication between two scalars which is a valid kernel.

Problem 8 (Version B) (8 credits)

a)

□ 0
□ 1
□ 2
□ 3
□ 4

$k(\mathbf{x},\mathbf{y}) = \left(\sum_i^M f(\mathbf{x})_i\right)\left(\sum_i^M f(\mathbf{y})_i\right)$
Function f together with the summation $\sum$ can be viewed as a feature map.
That leaves us with the multiplication between two scalars which is a valid kernel.
0 points for psd matrix (unless actually proved)

b)

□ 0
□ 1
□ 2
□ 3
□ 4

$k(\mathbf{x},\mathbf{y}) = \sum_z p(\mathbf{x}|z)p(\mathbf{y}|z)p(z)$, where $z \in \{1,\dots,Z\}$
The probability density function is a feature map $p : \mathbb{R}^N \to \mathbb{R}$. Same is true for the conditional density function.
So the function $p(\mathbf{x}|z)p(\mathbf{y}|z)$ is a valid kernel.
Multiplying the kernel with a constant $p(z)$ is a kernel.
The sum of kernels is a kernel.

Problem 8 (Version C) (8 credits)

0 □
1 □
2 □
3 □
4 □

a)

$k(\mathbf{x},\mathbf{y}) = \sum_i^M f(\mathbf{x})_i + \sum_i^M f(\mathbf{y})_i$
Function f together with the summation $\sum$ can be viewed as a feature map. Then we just need to check if $k(x, y) = x + y$ is a valid kernel. The kernel matrix has to be positive semidefinite. If we take the 2×2 matrix for two data points x and y, it is easy to show that the kernel matrix is not positive semidefinite since the determinant is always negative or zero.

$$\det\left(\begin{bmatrix} x+x & x+y \\ x+y & y+y \end{bmatrix}\right) = 4xy - (x+y)^2 = -(x-y)^2 \leq 0$$

0 □
1 □
2 □
3 □
4 □

b)

$k(\mathbf{x},\mathbf{y}) = \sum_z p(\mathbf{x}|z)p(\mathbf{y}|z)p(z)$, where $z \in \{1, ..., Z\}$
The probability density function is a feature map $p : \mathbb{R}^N \to \mathbb{R}$. Same is true for the conditional density function.
So the function $p(\mathbf{x}|z)p(\mathbf{y}|z)$ is a valid kernel.
Multiplying the kernel with a constant $p(z)$ is a kernel.
The sum of kernels is a kernel.

Problem 8 (Version D) (8 credits)

a)

□ 0
□ 1
□ 2
□ 3
□ 4

$k(\mathbf{x}, \mathbf{y}) = \log p(\mathbf{x}) \log p(\mathbf{y})$
The probability density function is a feature map $p : \mathbb{R}^N \rightarrow \mathbb{R}$.
Same goes for the logarithm $\log : \mathbb{R} \rightarrow \mathbb{R}$.
That leaves us with the multiplication between two scalars which is a valid kernel.

b)

□ 0
□ 1
□ 2
□ 3
□ 4

$k(\mathbf{x}, \mathbf{y}) = \sum_i^M f(\mathbf{x})_i + \sum_i^M f(\mathbf{y})_i$
Function f together with the summation $\sum$ can be viewed as a feature map. Then we just need to check if $k(x, y) = x + y$ is a valid kernel. The kernel matrix has to be positive semidefinite. If we take the 2×2 matrix for two data points x and y, it is easy to show that the kernel matrix is not positive semidefinite since the determinant is always negative or zero.

$$\det\left(\begin{bmatrix} x+x & x+y \\ x+y & y+y \end{bmatrix}\right) = 4xy - (x+y)^2 = -(x-y)^2 \leq 0$$

Problem 9 (Version A) (6 credits)

0 □
1 □
2 □
3 □

a)

We obtain the coordinates of the projected points by multiplying the data matrix by the first two eigenvectors of the covariance matrix $\mathbf{\Gamma}_{1:2}$, i.e.

$$\tilde{\mathbf{X}} = \mathbf{X}\mathbf{\Gamma}_{1:2}$$

$$\tilde{\mathbf{X}} = \begin{pmatrix} -2 & -5 \\ 6 & 0 \\ -2 & 1 \\ 6 & 0 \\ -2 & 1 \\ -2 & 1 \\ -2 & 1 \\ 0 & -1 \\ -2 & 1 \\ 0 & 1 \end{pmatrix}$$

0 □
1 □
2 □
3 □

b)

Two possible solutions:

1. We know that the variance when projected onto k-th component equals to the k-th largest eigenvalue of the covariance matrix.
2. We can first project the data onto the 3rd eigenvector $\mathbf{\Gamma}_3$ as $\tilde{\mathbf{x}} = \mathbf{X}\mathbf{\Gamma}_3$ and then directly compute the variance using the formula

$$\mathrm{Var}(\tilde{\mathbf{x}}) = \frac{1}{N}\sum_{i=1}^{N}(\tilde{x}_i - \mathrm{mean}(\tilde{\mathbf{x}}))^2$$

Normalizing by $N - 1$ instead of N in the variance computation is also considered correct.

In both cases we arrive at the same result

$$\mathrm{Var}(\tilde{\mathbf{x}}) = 1.4$$

Problem 9 (Version B) (6 credits)

a)

□ 0
□ 1
□ 2
□ 3

We obtain the coordinates of the projected points by multiplying the data matrix by the first two eigenvectors of the covariance matrix $\mathbf{\Gamma}_{1:2}$, i.e.

$$\tilde{\mathbf{X}} = \mathbf{X}\mathbf{\Gamma}_{1:2}$$

$$\tilde{\mathbf{X}} = \begin{pmatrix} 0.5 & 0 \\ -1 & 2 \\ -1 & 0 \\ 3.5 & 0 \\ 0 & -2 \\ 0 & 1 \\ -2 & -2 \\ 1 & -1 \\ 0 & 1 \\ -1 & 1 \end{pmatrix}$$

b)

□ 0
□ 1
□ 2
□ 3

Two possible solutions:

1. We know that the variance when projected onto k-th component equals to the k-th largest eigenvalue of the covariance matrix.
2. We can first project the data onto the 3rd eigenvector $\mathbf{\Gamma}_3$ as $\tilde{\mathbf{x}} = \mathbf{X}\mathbf{\Gamma}_3$ and then directly compute the variance using the formula

$$\mathrm{Var}(\tilde{\mathbf{x}}) = \frac{1}{N}\sum_{i=1}^{N}(\tilde{x}_i - \mathrm{mean}(\tilde{\mathbf{x}}))^2$$

Normalizing by $N-1$ instead of N in the variance computation is also considered correct.

In both cases we arrive at the same result

$$\mathrm{Var}(\tilde{\mathbf{x}}) = 1.4$$

Problem 9 (Version C) (6 credits)

0 □
1 □
2 □
3 □

a)

We obtain the coordinates of the projected points by multiplying the data matrix by the first two eigenvectors of the covariance matrix $\mathbf{\Gamma}_{1:2}$, i.e.

$$\tilde{\mathbf{X}} = \mathbf{X}\mathbf{\Gamma}_{1:2}$$

$$\tilde{\mathbf{X}} = \begin{pmatrix} 1 & -4 \\ 2 & 2 \\ -4.5 & 1 \\ 1.5 & 1 \\ -2 & -1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & -2 \\ 1 & 1 \end{pmatrix}$$

0 □
1 □
2 □
3 □

b)

Two possible solutions:

1. We know that the variance when projected onto k-th component equals to the k-th largest eigenvalue of the covariance matrix.
2. We can first project the data onto the 3rd eigenvector $\mathbf{\Gamma}_3$ as $\tilde{\mathbf{x}} = \mathbf{X}\mathbf{\Gamma}_3$ and then directly compute the variance using the formula

$$\mathrm{Var}(\tilde{\mathbf{x}}) = \frac{1}{N}\sum_{i=1}^{N}(\tilde{x}_i - \mathrm{mean}(\tilde{\mathbf{x}}))^2$$

Normalizing by $N-1$ instead of N in the variance computation is also considered correct.

In both cases we arrive at the same result

$$\mathrm{Var}(\tilde{\mathbf{x}}) = 1.4$$

Problem 9 (Version D) (6 credits)

a)

□ 0
□ 1
□ 2
□ 3

We obtain the coordinates of the projected points by multiplying the data matrix by the first two eigenvectors of the covariance matrix $\mathbf{\Gamma}_{1:2}$, i.e.

$$\tilde{\mathbf{X}} = \mathbf{X}\mathbf{\Gamma}_{1:2}$$

$$\tilde{\mathbf{X}} = \begin{pmatrix} 3 & 0 \\ 1 & 1 \\ 3 & 4 \\ -7 & 1 \\ -2 & 1 \\ 1 & -2 \\ 1 & -1 \\ 1 & -2 \\ -1 & -1 \\ 0 & -1 \end{pmatrix}$$

b)

□ 0
□ 1
□ 2
□ 3

Two possible solutions:

1. We know that the variance when projected onto k-th component equals to the k-th largest eigenvalue of the covariance matrix.
2. We can first project the data onto the 3rd eigenvector $\mathbf{\Gamma}_3$ as $\tilde{\mathbf{x}} = \mathbf{X}\mathbf{\Gamma}_3$ and then directly compute the variance using the formula

$$\operatorname{Var}(\tilde{\mathbf{x}}) = \frac{1}{N}\sum_{i=1}^{N}(\tilde{x}_i - \operatorname{mean}(\tilde{\mathbf{x}}))^2$$

Normalizing by $N - 1$ instead of N in the variance computation is also considered correct.

In both cases we arrive at the same result

$$\operatorname{Var}(\tilde{\mathbf{x}}) = 2$$

Problem 10 (Version A) (8 credits)

0 □
1 □
2 □

a)

No, because k-means with 2 clusters has linear decision boundary.

0 □
1 □
2 □
3 □

b)

No, since GMM with shared covariance matrix again produces a linear decision boundary, similar to Linear Discriminant Analysis.

0 □
1 □
2 □
3 □

c)

Yes, since GMM with separate diagonal covariance for each cluster can produce a quadratic decision boundary, similar to Naive Bayes / Quadratic Discriminant Analysis.

Another possible explanation: Yes, if both clusters have the same mean but different covariances, the decision boundary will be a circle.

Problem 10 (Version B) (8 credits)

a)

□ 0
□ 1
□ 2

No, because k-means with 2 clusters has linear decision boundary.

b)

□ 0
□ 1
□ 2
□ 3

No, since GMM with shared covariance matrix again produces a linear decision boundary, similar to Linear Discriminant Analysis.

c)

□ 0
□ 1
□ 2
□ 3

Yes, since GMM with separate diagonal covariance for each cluster can produce a quadratic decision boundary, similar to Naive Bayes / Quadratic Discriminant Analysis.

Another possible explanation: Yes, if both clusters have the same mean but different covariances, the decision boundary will be a circle.

Problem 10 (Version C) (8 credits)

0 □
1 □
2 □

a)

No, because k-means with 2 clusters has linear decision boundary.

0 □
1 □
2 □
3 □

b)

No, since GMM with shared covariance matrix again produces a linear decision boundary, similar to Linear Discriminant Analysis.

0 □
1 □
2 □
3 □

c)

Yes, since GMM with separate diagonal covariance for each cluster can produce a quadratic decision boundary, similar to Naive Bayes / Quadratic Discriminant Analysis.

Another possible explanation: Yes, if both clusters have the same mean but different covariances, the decision boundary will be a circle.

Problem 10 (Version D) (8 credits)

a)

□ 0
□ 1
□ 2

No, because k-means with 2 clusters has linear decision boundary.

b)

□ 0
□ 1
□ 2
□ 3

No, since GMM with shared covariance matrix again produces a linear decision boundary, similar to Linear Discriminant Analysis.

c)

□ 0
□ 1
□ 2
□ 3

Yes, since GMM with separate diagonal covariance for each cluster can produce a quadratic decision boundary, similar to Naive Bayes / Quadratic Discriminant Analysis.

Another possible explanation: Yes, if both clusters have the same mean but different covariances, the decision boundary will be a circle.

Problem 11 (Version A) (7 credits)

0 □ 1 □ 2 □

a)

Equality of opportunity requires that $\Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = 1\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = 1\right]$. Based on our generative model and the hint, we know that this is equivalent to

$$\Phi\left(\frac{w^T\boldsymbol{\mu}_a^1 + d}{||w||_2}\right) = \Phi\left(\frac{w^T\boldsymbol{\mu}_b^1 + d}{||w||_2}\right).$$

Both terms are the same if

$$w^T\boldsymbol{\mu}_a^1 = w^T\boldsymbol{\mu}_b^1 \iff w^T\left(\boldsymbol{\mu}_a^1 - \boldsymbol{\mu}_b^1\right) = 0 \iff w^T\left(\begin{bmatrix}0\\1\end{bmatrix} - \begin{bmatrix}1\\0\end{bmatrix}\right) = 0,$$

i.e. $\mathbf{w}^T$ is orthogonal to the connecting line of $\boldsymbol{\mu}_a^1$ and $\boldsymbol{\mu}_b^1$ and d can be an arbitrary value.

In particular, we can choose $\mathbf{w} = \begin{bmatrix}1\\1\end{bmatrix}, d = 0$.

0 □ 1 □ 2 □

b)

Separation requires that

$$\Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = 1\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = 1\right]$$

$$\Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = 0\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = 0\right]$$

Using the same argument as in b.), we can conclude that a linear classifier fulfilling independence would need to have weight vector $\mathbf{w}$ that is orthogonal to both $\mu_a^1 - \mu_b^1$ and $\mu_a^0 - \mu_b^0$. Since both vector differences are not colinear, this is not possible.

c)

□ 0
□ 1
□ 2
□ 3

Separation requires that

$$\Pr\left[r(\mathbf{x},A)=1|A=a\right]=\Pr\left[r(\mathbf{x},A)=1|A=b\right].$$

Due to the law of total probability, this is equivalent to

$$\sum_{c\in\{0,1\}}\Pr\left[r(\mathbf{x},A)=1|A=a,Y=c\right]\cdot\Pr\left[Y=c|A=a\right]=\sum_{c\in\{0,1\}}\Pr\left[r(\mathbf{x},A)=1|A=b,Y=c\right]\Pr\left[Y=c|A=b\right].$$

Since $\Pr\left[Y=c|A=a\right]=\Pr\left[Y=c|A=b\right]=\frac{1}{2}$, the above holds if

$$\Pr\left[r(\mathbf{x},A)=1|A=a,Y=1\right]=\Pr\left[r(\mathbf{x},A)=1|A=b,Y=0\right]$$

and

$$\Pr\left[r(\mathbf{x},A)=1|A=a,Y=0\right]=\Pr\left[r(\mathbf{x},A)=1|A=b,Y=1\right].$$

Using the same argument as in b.), the above is fulfilled by any $\mathbf{w}$ and d such that $\mathbf{w}$ is orthogonal to $\mu_a^1-\mu_b^0=\begin{bmatrix}0\\1\end{bmatrix}-\begin{bmatrix}1\\1\end{bmatrix}$ and $\mu_a^0-\mu_b^1=\begin{bmatrix}0\\0\end{bmatrix}-\begin{bmatrix}1\\0\end{bmatrix}$.

In particular, we can choose $\mathbf{w}=\begin{bmatrix}0\\1\end{bmatrix}$, $d=0$.

Problem 11 (Version B) (7 credits)

0 □
1 □
2 □

a)

Equality of opportunity requires that $\Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = 1\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = 1\right]$. Based on our generative model and the hint, we know that this is equivalent to

$$\Phi\left(\frac{w^T\boldsymbol{\mu}_a^1 + d}{||w||_2}\right) = \Phi\left(\frac{w^T\boldsymbol{\mu}_b^1 + d}{||w||_2}\right).$$

Both terms are the same if

$$w^T\boldsymbol{\mu}_a^1 = w^T\boldsymbol{\mu}_b^1 \iff w^T\left(\boldsymbol{\mu}_a^1 - \boldsymbol{\mu}_b^1\right) = 0 \iff w^T\left(\begin{bmatrix}0\\0\end{bmatrix} - \begin{bmatrix}1\\1\end{bmatrix}\right) = 0,$$

i.e. $\mathbf{w}^T$ is orthogonal to the connecting line of $\boldsymbol{\mu}_a^1$ and $\boldsymbol{\mu}_b^1$ and d can be an arbitrary value. In particular, we can choose $\mathbf{w} = \begin{bmatrix}-1\\1\end{bmatrix}$, $d = 0$.

0 □
1 □
2 □

b)

Separation requires that

$$\Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = 1\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = 1\right]$$

$$\Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = 0\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = 0\right]$$

Using the same argument as in b.), we can conclude that a linear classifier fulfilling independence would need to have weight vector $\mathbf{w}$ that is orthogonal to both $\mu_a^1 - \mu_b^1$ and $\mu_a^0 - \mu_b^0$. Since both vector differences are not colinear, this is not possible.

c)

□ 0
□ 1
□ 2
□ 3

Separation requires that

$$\Pr\left[r(\mathbf{x},A)=1|A=a\right]=\Pr\left[r(\mathbf{x},A)=1|A=b\right].$$

Due to the law of total probability, this is equivalent to

$$\sum_{c\in\{0,1\}}\Pr\left[r(\mathbf{x},A)=1|A=a,Y=c\right]\cdot\Pr\left[Y=c|A=a\right]=\sum_{c\in\{0,1\}}\Pr\left[r(\mathbf{x},A)=1|A=b,Y=c\right]\Pr\left[Y=c|A=b\right].$$

Since $\Pr\left[Y=c|A=a\right]=\Pr\left[Y=c|A=b\right]=\frac{1}{2}$, the above holds if

$$\Pr\left[r(\mathbf{x},A)=1|A=a,Y=1\right]=\Pr\left[r(\mathbf{x},A)=1|A=b,Y=0\right]$$

and

$$\Pr\left[r(\mathbf{x},A)=1|A=a,Y=0\right]=\Pr\left[r(\mathbf{x},A)=1|A=b,Y=1\right].$$

Using the same argument as in b.), the above is fulfilled by any $\mathbf{w}$ and d such that $\mathbf{w}$ is orthogonal to $\mu_a^1-\mu_b^0=\begin{bmatrix}0\\0\end{bmatrix}-\begin{bmatrix}1\\0\end{bmatrix}$ and $\mu_a^0-\mu_b^1=\begin{bmatrix}0\\1\end{bmatrix}-\begin{bmatrix}1\\1\end{bmatrix}$.

In particular, we can choose $\mathbf{w}=\begin{bmatrix}0\\1\end{bmatrix}$, $d=0$.

Problem 11 (Version C) (7 credits)

0 □
1 □
2 □

a)

Equality of opportunity requires that $\Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = 1\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = 1\right]$. Based on our generative model and the hint, we know that this is equivalent to

$$\Phi\left(\frac{w^T \boldsymbol{\mu}_a^1 + d}{||w||_2}\right) = \Phi\left(\frac{w^T \boldsymbol{\mu}_b^1 + d}{||w||_2}\right).$$

Both terms are the same if

$$w^T \boldsymbol{\mu}_a^1 = w^T \boldsymbol{\mu}_b^1 \iff w^T\left(\boldsymbol{\mu}_a^1 - \boldsymbol{\mu}_b^1\right) = 0 \iff w^T\left(\begin{bmatrix}0\\0\end{bmatrix} - \begin{bmatrix}1\\1\end{bmatrix}\right) = 0,$$

i.e. $\mathbf{w}^T$ is orthogonal to the connecting line of $\boldsymbol{\mu}_a^1$ and $\boldsymbol{\mu}_b^1$ and d can be an arbitrary value. In particular, we can choose $\mathbf{w} = \begin{bmatrix}-1\\1\end{bmatrix}$, $d = 0$.

0 □
1 □
2 □

b)

Separation requires that

$$\Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = 1\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = 1\right]$$

$$\Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = 0\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = 0\right]$$

Using the same argument as in b.), we can conclude that a linear classifier fulfilling independence would need to have weight vector $\mathbf{w}$ that is orthogonal to both $\mu_a^1 - \mu_b^1$ and $\mu_a^0 - \mu_b^0$. Since both vector differences are not colinear, this is not possible.

c)

□ 0
□ 1
□ 2
□ 3

Separation requires that

$$\Pr\left[r(\mathbf{x}, A) = 1 | A = a\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b\right].$$

Due to the law of total probability, this is equivalent to

$$\sum_{c\in\{0,1\}} \Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = c\right] \cdot \Pr\left[Y = c | A = a\right] = \sum_{c\in\{0,1\}} \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = c\right] \Pr\left[Y = c | A = b\right].$$

Since $\Pr\left[Y = c | A = a\right] = \Pr\left[Y = c | A = b\right] = \frac{1}{2}$, the above holds if

$$\Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = 1\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = 0\right]$$

and

$$\Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = 0\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = 1\right].$$

Using the same argument as in b.), the above is fulfilled by any $\mathbf{w}$ and d such that $\mathbf{w}$ is orthogonal to $\mu_a^1 - \mu_b^0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and $\mu_a^0 - \mu_b^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

In particular, we can choose $\mathbf{w} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $d = 0$.

Problem 11 (Version D) (7 credits)

0 □
1 □
2 □

a)

Equality of opportunity requires that $\Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = 1\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = 1\right]$. Based on our generative model and the hint, we know that this is equivalent to

$$\Phi\left(\frac{w^T \boldsymbol{\mu}_a^1 + d}{||w||_2}\right) = \Phi\left(\frac{w^T \boldsymbol{\mu}_b^1 + d}{||w||_2}\right).$$

Both terms are the same if

$$w^T \boldsymbol{\mu}_a^1 = w^T \boldsymbol{\mu}_b^1 \iff w^T\left(\boldsymbol{\mu}_a^1 - \boldsymbol{\mu}_b^1\right) = 0 \iff w^T\left(\begin{bmatrix}1\\0\end{bmatrix} - \begin{bmatrix}0\\1\end{bmatrix}\right) = 0,$$

i.e. $\mathbf{w}^T$ is orthogonal to the connecting line of $\boldsymbol{\mu}_a^1$ and $\boldsymbol{\mu}_b^1$ and d can be an arbitrary value.

In particular, we can choose $\mathbf{w} = \begin{bmatrix}1\\1\end{bmatrix}, d = 0$.

0 □
1 □
2 □

b)

Separation requires that

$$\Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = 1\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = 1\right]$$

$$\Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = 0\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = 0\right]$$

Using the same argument as in b.), we can conclude that a linear classifier fulfilling independence would need to have weight vector $\mathbf{w}$ that is orthogonal to both $\mu_a^1 - \mu_b^1$ and $\mu_a^0 - \mu_b^0$. Since both vector differences are not colinear, this is not possible.

c)

□ 0
□ 1
□ 2
□ 3

Separation requires that

$$\Pr\left[r(\mathbf{x}, A) = 1 | A = a\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b\right].$$

Due to the law of total probability, this is equivalent to

$$\sum_{c \in \{0,1\}} \Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = c\right] \cdot \Pr\left[Y = c | A = a\right] = \sum_{c \in \{0,1\}} \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = c\right] \Pr\left[Y = c | A = b\right].$$

Since $\Pr\left[Y = c | A = a\right] = \Pr\left[Y = c | A = b\right] = \frac{1}{2}$, the above holds if

$$\Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = 1\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = 0\right]$$

and

$$\Pr\left[r(\mathbf{x}, A) = 1 | A = a, Y = 0\right] = \Pr\left[r(\mathbf{x}, A) = 1 | A = b, Y = 1\right].$$

Using the same argument as in b.), the above is fulfilled by any $\mathbf{w}$ and d such that $\mathbf{w}$ is orthogonal to $\mu_a^1 - \mu_b^0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\mu_a^0 - \mu_b^1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

In particular, we can choose $\mathbf{w} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $d = 0$.

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