Problem 8

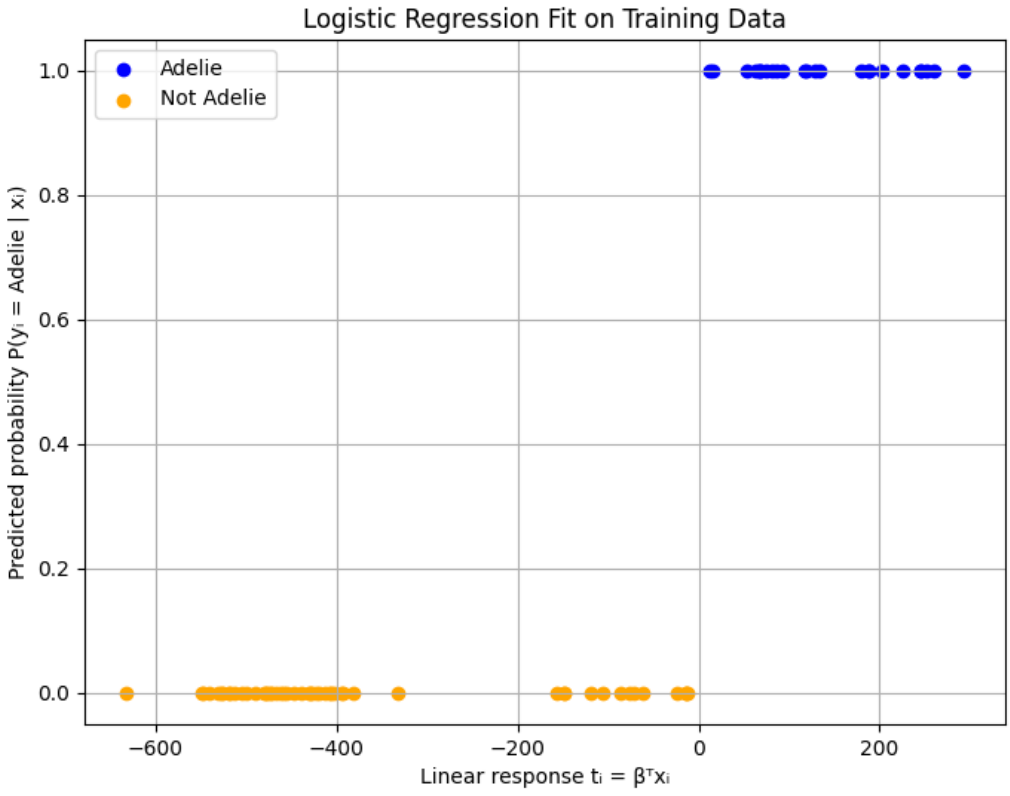
Task a

```python
# fit logistic regression without regularisation(penalty)
model = LogisticRegression(penalty=None, solver="lbfgs", max_iter=1000)

model.fit(X_train, y_train)
```

```
Coefficients (β):
  bill_length_mm: -21.4982
  bill_depth_mm: 81.8189
  flipper_length_mm: -1.9752
  body_mass_g: -0.0479

Training accuracy: 1.0000
Testing accuracy: 0.9333
```



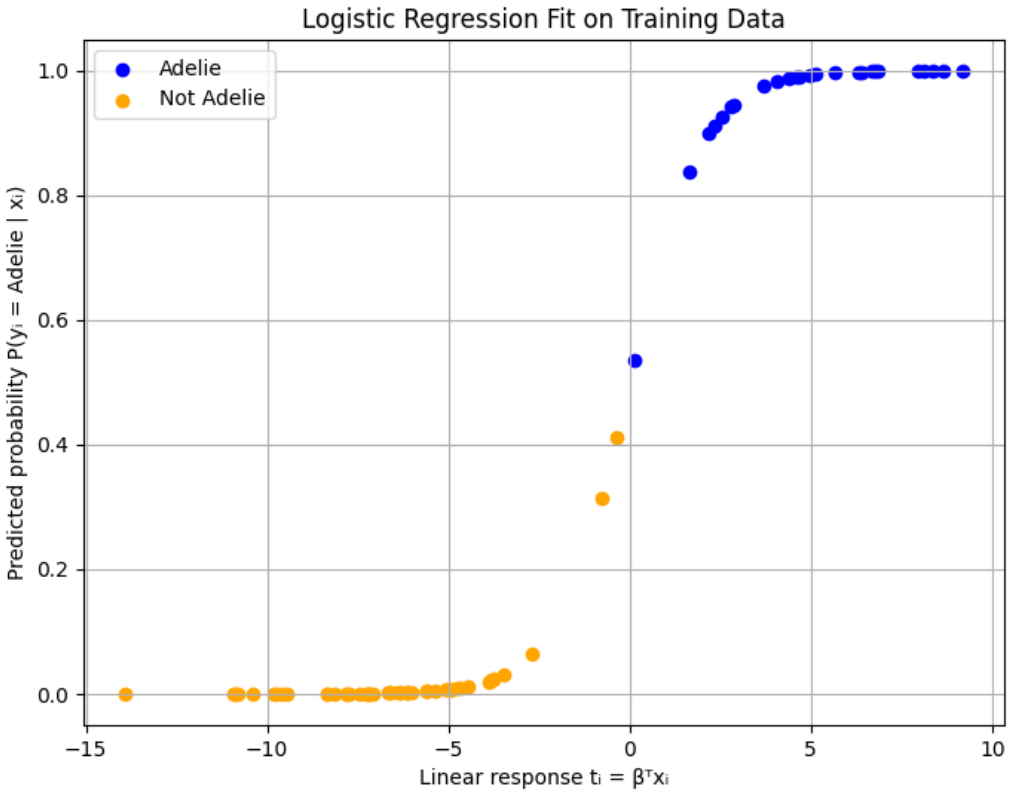
Task b

```python
# fit logistic regression with Lasso regularisation
model = LogisticRegression(penalty='l1',solver='liblinear',C=1.0,
max_iter=1000)

model.fit(X_train, y_train)
```

```
Coefficients (β):
  bill_length_mm: -0.9810
  bill_depth_mm: 1.7242
  flipper_length_mm: 0.0350
  body_mass_g: 0.0012

Training accuracy: 1.0000
Testing accuracy: 0.9733
```



Task c

I use python not R

Problem 9

Proof that QDA is not linear when class variances differ

For a single feature (p=1), the class-conditional distributions are

$$x \mid y = k \sim \mathcal{N}(\mu_k, \sigma_k^2), \quad k = 1, 2 \tag{1}$$

The QDA discriminant functions are

$$\delta_k(x) = -\frac{1}{2}\log \sigma_k^2 - \frac{(x-\mu_k)^2}{2\sigma_k^2} + \log \pi_k \tag{2}$$

The decision boundary is given by

$$\delta_1(x) = \delta_2(x) \tag{3}$$

Expanding, we get

$$-\frac{(x-\mu_1)^2}{\sigma_1^2} + \frac{(x-\mu_2)^2}{\sigma_2^2} = \text{constant} \tag{4}$$

$$\left(\frac{1}{\sigma_2^2} - \frac{1}{\sigma_1^2}\right)x^2 + \text{linear term} + \text{constant} = 0 \tag{5}$$

If

$$\sigma_1^2 \neq \sigma_2^2 \tag{6}$$

, the coefficient of (x^2) is nonzero, so the boundary is **quadratic**, not linear.

Problem 10

Task a

=== Means per class ===

label	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
0	47.818	15.890	211.30	4657.0
1	38.124	18.336	188.88	3576.0

=== Standard deviations per class ===

label	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
0	3.599472	1.965441	11.792855	787.531017
1	2.781528	1.204118	6.320074	461.343148

=== Class probabilities with Laplace smoothing ===
P(y=0 | train) = 0.6623 (notAdelie)
P(y=1 | train) = 0.3377 (Adelie)

Task b

$$\mathcal{N}(x_i \mid \mu_{i,y}, \sigma_{i,y}^2) = \frac{1}{\sqrt{2\pi}\,\sigma_{i,y}} \exp\left(-\frac{(x_i - \mu_{i,y})^2}{2\sigma_{i,y}^2}\right) \tag{7}$$

$$p(x \mid y) = \prod_{i=1}^{4} \mathcal{N}(x_i \mid \mu_{i,y}, \sigma_{i,y}^2). \tag{8}$$

$$\hat{P}(y = \text{Adelie} \mid \mathbf{x}) = \frac{P(y = \text{Adelie}) \prod_{i=1}^{4} p(x_i \mid y = \text{Adelie})}{\sum_{y' \in \{\text{Adelie}, \text{NotAdelie}\}} P(y') \prod_{i=1}^{4} p(x_i \mid y = y')} \tag{9}$$

Task c

Test set classification accuracy: 0.9200

Probabilities P(y=Adelie | x) for first three test penguins:
Penguin 1: 0.9964
Penguin 2: 0.7985
Penguin 3: 0.9988

Problem 11

Task a

According to the authors, discriminative learning is not always better than generative learning. While discriminative classifiers (like logistic regression) have lower asymptotic error and thus perform better with large training sets, generative classifiers (like naive Bayes) can converge faster and achieve close to their best performance even with smaller training sets. Therefore, in small-sample regimes, generative learning may outperform discriminative learning, and only with larger datasets does discriminative learning become superior. This demonstrates that the relative performance depends on the training set size.

Task b

Ng and Jordan discuss two families of models:

1. **Continuous inputs (Gaussian features)**
    - Generative model (`h_Gen`): Normal Discriminant Analysis (NDA)
    - Discriminative model (`h_Dis`): Logistic Regression
    - Optimization objectives:
        - `h_Gen`: maximize the joint likelihood (p(x, y))
        - `h_Dis`: maximize the conditional likelihood (p(y|x))
2. **Discrete inputs (categorical features)**
    - Generative model (`h_Gen`): Naive Bayes
    - Discriminative model (`h_Dis`): Logistic Regression
    - Optimization objectives:
        - `h_Gen`: maximize the joint likelihood (p(x, y))
        - `h_Dis`: maximize the conditional likelihood (p(y|x))

Task c

The figure suggests that there are two distinct performance regimes:

1. **Small sample regime (small (m))**:
    - Generative models (NDA or Naive Bayes) quickly approach their own asymptotic error.
    - Even though their final error is higher than that of discriminative models, they perform better when the training set is small.
2. **Large sample regime (large (m))**:
    - Discriminative models (Logistic Regression) continue to improve and eventually achieve lower error than generative models.

Implication for the model families in Task b:

- For small training sets, NDA or Naive Bayes may outperform Logistic Regression.
- For large training sets, Logistic Regression is likely to outperform the generative models.
- Therefore, the choice between generative and discriminative approaches depends not only on the data type but also on the size of the training set.

Problem 12

Task a

The Naive Bayes assumption is **not valid** for this toy data set.

Reason: Naive Bayes requires that the features be conditionally independent given the class label (y), i.e.,

$$p(x_1, x_2 \mid y) = p(x_1 \mid y)p(x_2 \mid y) \tag{10}$$

In the toy data, the conditional probability of the class is

$$p(y = 1 \mid x_1, x_2) = \sigma(0.1 - 2x_1 + x_2 + 0.2x_1x_2), \tag{11}$$

which includes the interaction term `0.2 x_1 x_2`. This means that given (y), (x_1) and (x_2) are **not independent**. Therefore, the Naive Bayes conditional independence assumption is violated.

Task b

```
     n  NB_acc  NB_ppx  LR_acc  LR_ppx  LRi_acc  LRi_ppx  OptimalBayes_acc
OptimalBayes_ppx  Dummy_acc  Dummy_ppx
     8  0.6643  3.4408  0.6964  1.7466   0.6779   1.7729             0.793
          1.5465     0.5214     2.0431
    16  0.7635  1.7834  0.7574  1.6243   0.7608   1.6198             0.793
          1.5465     0.5214     2.0431
    32  0.7476  1.7086  0.7854  1.6354   0.7853   1.6354             0.793
          1.5465     0.4786     2.0093
    64  0.7855  1.5979  0.7917  1.5824   0.7917   1.5808             0.793
          1.5465     0.5214     1.9983
   128  0.7889  1.5599  0.7909  1.5608   0.7813   1.5977             0.793
          1.5465     0.5214     2.0000
   256  0.7919  1.5554  0.7912  1.5503   0.7925   1.5488             0.793
          1.5465     0.5214     1.9982
   512  0.7903  1.5674  0.7905  1.5540   0.7930   1.5518             0.793
          1.5465     0.5214     1.9984
  1024  0.7919  1.5689  0.7921  1.5495   0.7917   1.5474             0.793
          1.5465     0.4786     2.0011
  2048  0.7888  1.5656  0.7915  1.5495   0.7921   1.5473             0.793
          1.5465     0.5214     1.9983
      4096  0.7902  1.5648  0.7915  1.5491   0.7924   1.5467
0.793           1.5465     0.5214     1.9986
```

Task c

1. **Model types:**
    - NB: generative, probabilistic
    - LR: discriminative, probabilistic
    - LRi: discriminative, probabilistic
    - OptimalBayes: generative, probabilistic (true model)

- Dummy: baseline, probabilistic

2. **Accuracy and perplexity:**
   - OptimalBayes achieves the highest accuracy and lowest perplexity.
   - LRi closely matches OptimalBayes for large datasets.
   - NB performs worse for large datasets but can be better in very small samples.
   - Dummy performs worst overall.
3. **Relation to previous insights:**
   - Confirms the two performance regimes: generative models converge faster for small data, discriminative models reach lower asymptotic error with large data.
4. **LRi performance:**
   - Including the interaction term allows LRi to capture the true generating process.
   - With sufficient data, coefficients converge to the true values, yielding near-optimal predictions.
5. **Dummy classifier:**
   - Never outperforms OptimalBayes.

Problem 13

1. get Gini

$$G = 1 - \sum_{k=1}^{2} p_k^2 = 1 - \frac{8}{23}^2 - \frac{15}{23}^2 = \frac{240}{529} \approx 0.455 \tag{12}$$

2. minimize G_split

$$G_{split} = \frac{N_{left}}{N} G_{left} + \frac{N_{right}}{N} G_{right} \tag{13}$$

3. generate tree

split data point 1: x1 = 2

left: 2(0) + 8(1) = 10

right: 6(0) + 7(1) = 13

G = 0.42

left split data point 2: x1 = 1 G = 0

right split data point 2: x2 = 2 G = 0

4. tree

```
x1 < 2 ?  (G = 0.42)
├─ Yes: x1 < 1 ?  (G = 0)
|    ├─ Yes: Class 1
|    └─ No:  Class 0
└─ No: x2 < 2 ?  (G = 0)
     ├─ Yes: Class 1
     └─ No:  Class 0
```

Problem 14

Task a

1-NN

x=2.5,4.0,10.5,14.0,15.5

error = 0

3-NN

x=10.5

error point 3 ,15

Task b

1. Small k
    - The classifier is very sensitive to individual points.
    - The decision boundary wiggles a lot, trying to go exactly between points of different classes.
    - Can overfit the training data.
    - 1-NN boundary will be exactly at midpoints between each +1 and -1 pair
2. Large k
    - Each prediction is based on many neighbors, so majority class dominates.
    - The boundary becomes very smooth, possibly linear or almost flat in 1D.
    - Can underfit the data because local variations are ignored.
    - ∞-NN all points are predicted as majority class

Problem 15

Task a

Choose normal vector w=[−1, 0] and bias b=3. The hyperplane equation:

$$w^T x + b = 0 => -x_1 + 3 = 0 \tag{14}$$

The nearest positives lie at x1=2.0 and the nearest negatives at x1=4.0, so the margin

$$\gamma = 1 \tag{15}$$

Support vectors, Points touching the margin are:

Positive support: D = (2.0, 0.0)

Negative supports: E = (4.0, 1.0) and F = (4.0, -1.0)

Task b

Task c

Blue (0,0) (2,2) (3,8)

Red (-1,1)

Task d

$$(1 + X_1)^2 + (2 - X_2)^2 = 4 \tag{16}$$

$$(X_1^2 + 2X_1 + 1) + (X_2^2 - 4X_2 + 4) - 4 = 0 \tag{17}$$

$$X_1^2 + X_2^2 + 2X_1 - 4X_2 + 1 = 0 \tag{18}$$

So the decision boundary is linear in X1 X2 X1^2 X2^2

Problem 16

Task a

In the past weeks, I learned several fundamental and advanced concepts in machine learning, especially the differences between probabilistic and algorithmic approaches. From Lecture 5, shrinkage methods such as ridge and lasso clarified how regularisation helps prevent overfitting by controlling model complexity. I also understood how linear discriminative classifiers such as logistic regression directly model P(y | x), and why focusing on the decision boundary often leads to better predictive performance. The exercises helped reinforce how regularisation affects coefficient estimates and model interpretability.

Lectures 6 and 7 shifted the focus to generative learning and algorithmic methods. I learned how generative classifiers like LDA and QDA model P(x | y) and why this can be advantageous, especially when dealing with missing data. We also covered ROC curves, which gave me a more principled way to compare classifiers. Later, exploring k-NN and decision trees introduced an entirely different philosophy of learning: instead of estimating distributions or parametric functions, these models rely on instance-based reasoning or hierarchical partitioning. Working through problems 9–14 made these differences very concrete, particularly how k affects bias–variance tradeoffs and how tree depth controls model flexibility.

Finally, the advanced topics of Lecture 8 covered SVMs and ensemble methods. I found SVMs particularly interesting due to the geometric intuition of margins and support vectors. Understanding hinge loss and the connection to linear models made SVMs feel less mysterious. Ensemble methods, especially random forests and gradient boosting, were eye-opening: instead of designing a single strong model, combining many weak learners can significantly improve performance. This felt highly relevant to practical machine learning workflows, where ensemble models are often state of the art. The exercises and the final problems strengthened my understanding of margin-based learning and aggregation methods.

Task b

17h