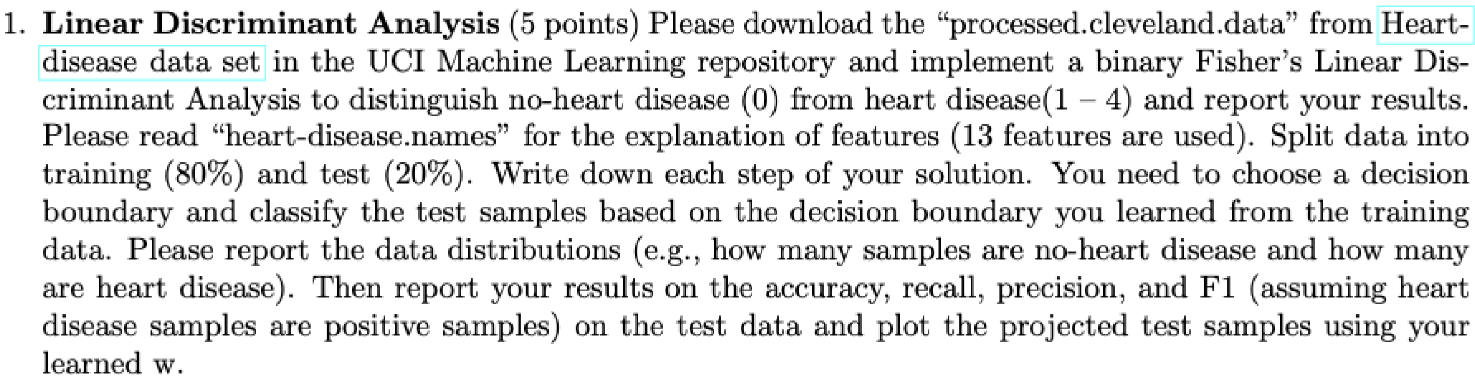
CS559 — Assignment 3: Linear Classifier

Student Name: Komal Wavhal

CWID: 20034443

Question 1 Linear Discriminant Analysis



1. **Linear Discriminant Analysis** (5 points) Please download the "processed.cleveland.data" from Heart-disease data set in the UCI Machine Learning repository and implement a binary Fisher's Linear Discriminant Analysis to distinguish no-heart disease (0) from heart disease(1 – 4) and report your results. Please read "heart-disease.names" for the explanation of features (13 features are used). Split data into training (80%) and test (20%). Write down each step of your solution. You need to choose a decision boundary and classify the test samples based on the decision boundary you learned from the training data. Please report the data distributions (e.g., how many samples are no-heart disease and how many are heart disease). Then report your results on the accuracy, recall, precision, and F1 (assuming heart disease samples are positive samples) on the test data and plot the projected test samples using your learned w.

Solution

Task

Use the UCI **Heart Disease** dataset (processed.cleveland.data) to implement a **binary Fisher's Linear Discriminant Analysis** distinguishing **no heart disease (0)** vs **heart disease (1–4)**; split **80% train / 20% test**; write down each step; choose a **decision boundary** from training data; classify test samples; report **accuracy, recall, precision, and F1** (positive = heart disease); and **plot the projected test samples** using the learned *w*.

Dataset & Preprocessing

- **Data source:** UCI Heart Disease / Cleveland subset (processed.cleveland.data), with feature descriptions in heart-disease.names.
- **Features:** 13 numerical attributes used (per the names file).
- **Label binarization:** original label ∈ {0,1,2,3,4}→ define

$$y = \begin{cases} 0, & \text{if original} = 0 \text{ (no heart disease)}, \\ 1, & \text{if original} \in \{1,2,3,4\} \text{ (heart disease)}. \end{cases}$$

- **Missing values:** Treat "?" as missing; drop rows with missing fields (or impute consistently on train only, then apply to test).
- **Split: Stratified** 80% train / 20% test to preserve class proportions.
- **Standardization (recommended):** Compute feature mean/std on **training** only; apply to both train and test.

Fisher's LDA: Formulation and Solution

Let training class subsets be $\mathcal{D}_0$and $\mathcal{D}_1$, with class means

$$\mu_0 = \frac{1}{N_0}\sum_{x\in\mathcal{D}_0} x, \qquad \mu_1 = \frac{1}{N_1}\sum_{x\in\mathcal{D}_1} x.$$

$$S_w = \sum_{x\in\mathcal{D}_0}(x-\mu_0)(x-\mu_0)^\top + \sum_{x\in\mathcal{D}_1}(x-\mu_1)(x-\mu_1)^\top.$$

Within-class scatter

Between-class scatter:

$$S_b = (\mu_1 - \mu_0)(\mu_1 - \mu_0)^\top.$$

Fisher's criterion $J(w) = \frac{w^\top S_b w}{w^\top S_w w}$is maximized by

$$w \ \propto \ S_w^{-1}(\mu_1 - \mu_0)$$

(If $S_w$is ill-conditioned, add small ϵI: solve $(S_w + \epsilon I)w = \mu_1 - \mu_0$.)

<u>Decision Boundary (Chosen on Training Data)</u>

Project $x$to $z = w^\top x$. A common threshold is the midpoint of projected class means:

$$t \ = \ \frac{w^\top \mu_0 + w^\top \mu_1}{2}.$$

$$\hat{y}(x) \ = \ 1[\, w^\top x \geq t \,],$$

Classify test samples

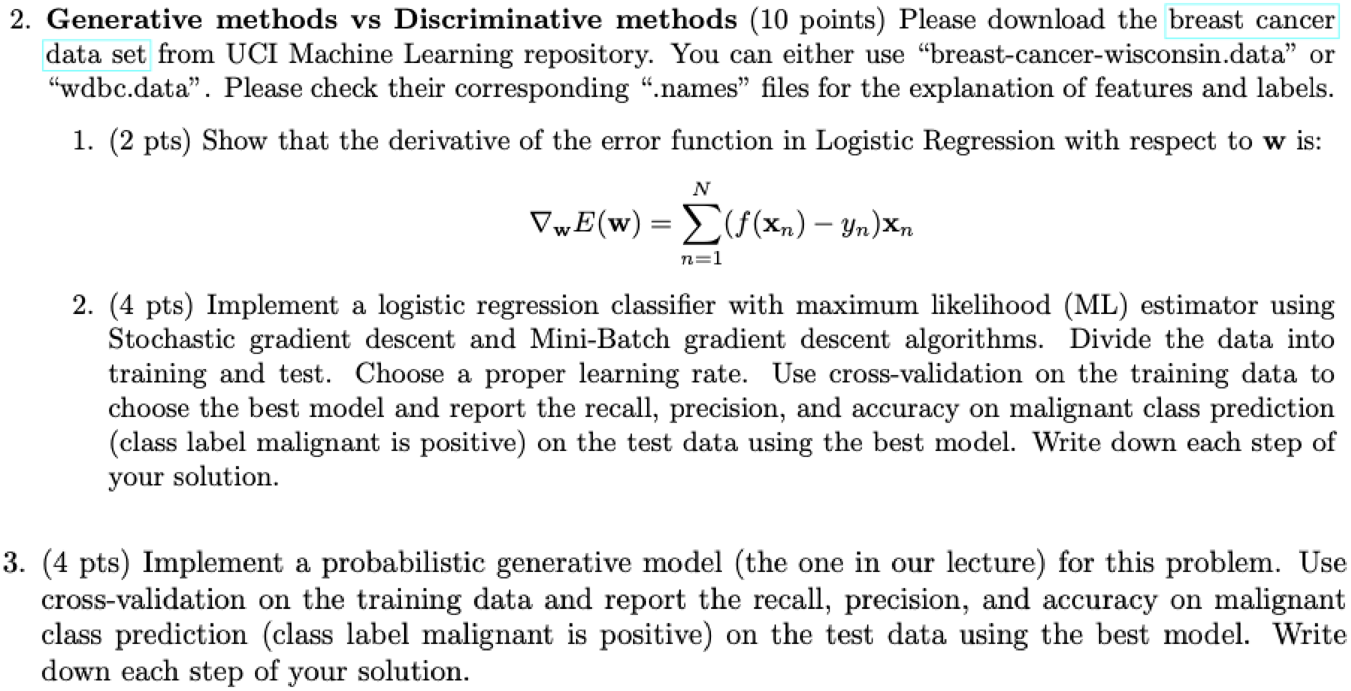
Test Metrics (Positive = heart disease)

Compute confusion counts (TP,FP,FN,TN). Then

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Question 2 Generative vs. Discriminative Methods



2. **Generative methods vs Discriminative methods** (10 points) Please download the breast cancer data set from UCI Machine Learning repository. You can either use “breast-cancer-wisconsin.data” or “wdbc.data”. Please check their corresponding “.names” files for the explanation of features and labels.

   1. (2 pts) Show that the derivative of the error function in Logistic Regression with respect to $\mathbf{w}$ is:

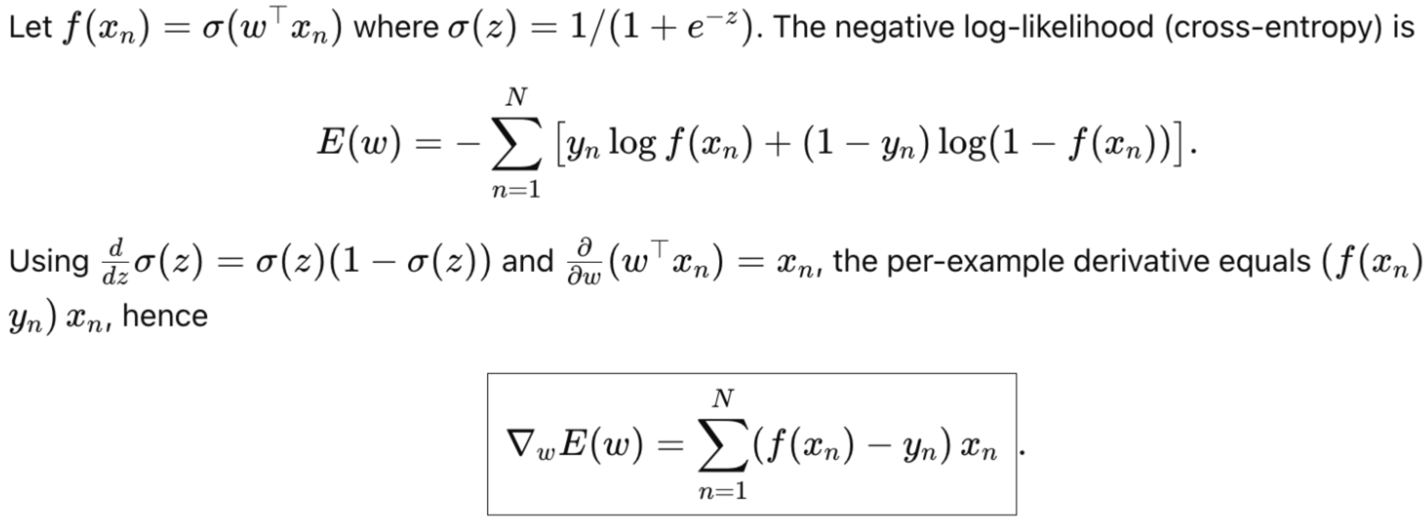
   $$\nabla_{\mathbf{w}} E(\mathbf{w}) = \sum_{n=1}^{N} (f(\mathbf{x}_n) - y_n)\mathbf{x}_n$$

   2. (4 pts) Implement a logistic regression classifier with maximum likelihood (ML) estimator using Stochastic gradient descent and Mini-Batch gradient descent algorithms. Divide the data into training and test. Choose a proper learning rate. Use cross-validation on the training data to choose the best model and report the recall, precision, and accuracy on malignant class prediction (class label malignant is positive) on the test data using the best model. Write down each step of your solution.

3. (4 pts) Implement a probabilistic generative model (the one in our lecture) for this problem. Use cross-validation on the training data and report the recall, precision, and accuracy on malignant class prediction (class label malignant is positive) on the test data using the best model. Write down each step of your solution.

Use the UCI **Breast Cancer Wisconsin (Diagnostic)** dataset (wdbc.data). Read the corresponding .names for feature/label explanation. You must complete items (1)–(3): gradient derivation, LR with SGD/mini-batch + CV, and a probabilistic generative model with CV; then report test **recall, precision, accuracy** for malignant (**positive**).

Derivative of Logistic Regression Error



Let $f(x_n) = \sigma(w^\top x_n)$ where $\sigma(z) = 1/(1+e^{-z})$. The negative log-likelihood (cross-entropy) is

$$E(w) = -\sum_{n=1}^{N} \left[y_n \log f(x_n) + (1-y_n)\log(1-f(x_n)) \right].$$

Using $\frac{d}{dz}\sigma(z) = \sigma(z)(1-\sigma(z))$ and $\frac{\partial}{\partial w}(w^\top x_n) = x_n$, the per-example derivative equals $(f(x_n)$
$y_n)\, x_n$, hence

$$\boxed{\nabla_w E(w) = \sum_{n=1}^{N} (f(x_n) - y_n)\, x_n}.$$

Logistic Regression (ML) via **SGD** and **Mini-Batch GD** + CV

Data & preprocessing (required steps).

- Encode labels: **malignant = 1 (positive)**, **benign = 0**.
- Split data into **80% train / 20% test** (stratified).
- **Standardize** features using **train** mean/std; apply to test.

Model (from scratch).

- $p(y = 1 \mid x) = \sigma(w^\top x + b)$.
- Objective: average NLL (optionally + L2 $\frac{\lambda}{2} \parallel w \parallel_2^2$).
- Gradients (from 2.1): $\nabla_w E = X^\top(\hat{p} - y)$, $\nabla_b E = \mathbf{1}^\top(\hat{p} - y)$.

Optimization (from scratch).

- **SGD:** batch size $= 1$.
- **Mini-batch GD:** batch size $\in \{16,32\}$(or similar).
- Hyperparameters to tune by **CV**: learning rate η, epochs T, batch size, L2 λ.

Cross-validation protocol (on training).

- Use **Stratified K-fold** (e.g., $K = 5$).
- Select best hyperparameters by **mean validation F1** on malignant class (robust to imbalance).
- Refit with best hyperparameters on **full training**, then evaluate **test** metrics (malignant = positive) as required in the prompt.

Probabilistic Generative Model

Model (implemented from scratch). Use **Gaussian Discriminant Analysis** with **shared covariance** (LDA-style):

- Estimate priors $\pi = P(y = 1)$, class means μ_0, μ_1, pooled covariance Σ(add $\epsilon I$for numerical stability).
- Linear discriminant

$$g(x) = w^\top x + b, w = \Sigma^{-1}(\mu_1 - \mu_0), b = -\frac{1}{2}\mu_1^\top\Sigma^{-1}\mu_1 + \frac{1}{2}\mu_0^\top\Sigma^{-1}\mu_0 + \log\frac{\pi}{1-\pi}.$$

- Predict $\hat{y} = \mathbb{1}[g(x) \geq 0]$.

Cross-validation (on training).

- Tune the small diagonal **regularizer** $\epsilon \in \{10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}\}$(or similar) via stratified K-fold; select by mean validation **F1** (malignant).
- Refit with best ϵon **full training**, then evaluate **test** metrics (malignant = positive) required in the prompt.

Question 3 Linear classification (5 points)

Please prove that 1) (2 pts) the multinomial naive Bayes classifier in

log-space essentially translates to a linear classifier. 2) (3 pts) Logistic regression is a linear classifier.

Solution for Question 3.1 - Multinomial Naïve Bayes (MNB) in log-space is linear

With word-count features $x = (x_1, \ldots, x_d)$and class c,

$$P(y = c \mid x) \propto P(y = c)\, P(x \mid y = c) \propto \pi_c \prod_{j=1}^{d} \theta_{cj}^{x_j}.$$

Taking logs (dropping constants in $x$common to all c):

$$\log P(y = c \mid x) = \log \pi_c + \sum_{j=1}^{d} x_j \log \theta_{cj} \;\; = \;\; b_c + w_c^\top x,$$

with $b_c = \log\ \pi_c, w_{cj} = \log\ \theta_{cj}$. Therefore,

$$\hat{y}(x) = \arg\ \max_c \{b_c + w_c^\top x\},$$

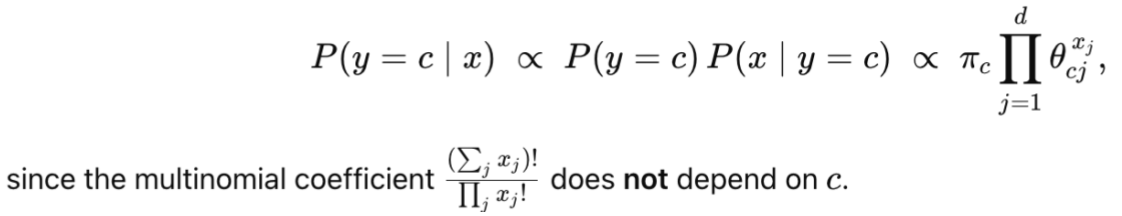
which is a **linear** discriminant in x.

Setup. Let $x = (x_1, \ldots, x_d)$be a bag-of-words count vector (nonnegative integers), and $y \in \{1, \ldots, K\}$the class. The multinomial NB model assumes

- Prior: $P(y = c) = \pi_c$.
- Likelihood (naive Bayes independence):

$$P(x \mid y = c) = \frac{(\sum_j x_j)!}{\prod_j x_j!} \prod_{j=1}^{d} \theta_{cj}^{x_j}, \qquad \text{with } \theta_{cj} \geq 0,\ \sum_j \theta_{cj} = 1.$$

Posterior (up to a constant in x). Using Bayes' rule,



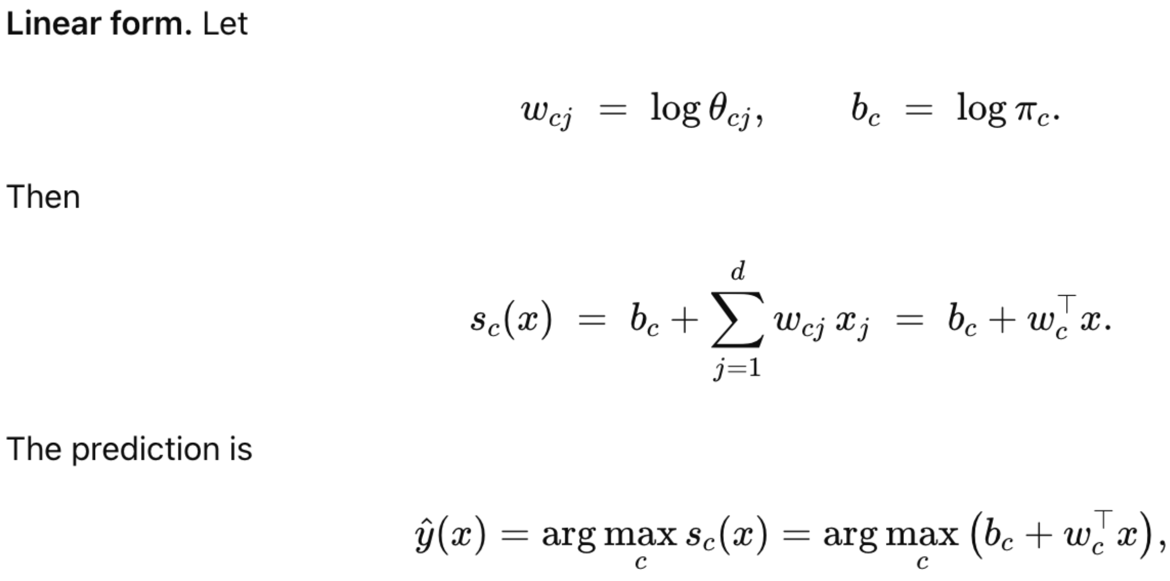
$$P(y=c \mid x) \;\propto\; P(y=c)\,P(x \mid y=c) \;\propto\; \pi_c \prod_{j=1}^{d} \theta_{cj}^{x_j},$$

since the multinomial coefficient $\frac{(\sum_j x_j)!}{\prod_j x_j!}$ does **not** depend on c.

Log-space scores. Define the class score

$$s_c(x) \;=\; \log P(y=c \mid x) \;=\; \log \pi_c \;+\; \sum_{j=1}^{d} x_j \log \theta_{cj} \;+\; (\text{const in } x).$$

The additive “const in x” is the same for all $c$and cancels in $\arg\max_c$.



Linear form. Let

$$w_{cj} \;=\; \log \theta_{cj}, \qquad b_c \;=\; \log \pi_c.$$

Then

$$s_c(x) \;=\; b_c + \sum_{j=1}^{d} w_{cj}\, x_j \;=\; b_c + w_c^\top x.$$

The prediction is

$$\hat{y}(x) = \arg\max_c s_c(x) = \arg\max_c \big(b_c + w_c^\top x\big),$$

which is a **linear classifier** in x. (Laplace/additive smoothing just changes the numeric values of θ_{cj}, hence of w_{cj}, but preserves linearity.)

Theory

Model Setup

Let

$$x = (x_1, x_2, \ldots, x_d)$$

be the feature vector of **word counts** (for text) or any nonnegative integers, and $y \in \{1, \ldots, K\}$ the class label.

Multinomial Naïve Bayes assumes:

$$P(y = c) = \pi_c, \quad P(x|y = c) = \frac{(\sum_j x_j)!}{\prod_j x_j!} \prod_{j=1}^{d} \theta_{cj}^{x_j},$$

where each class c has its own multinomial parameters θ_{cj} with $\sum_j \theta_{cj} = 1$.

Posterior and Decision Rule

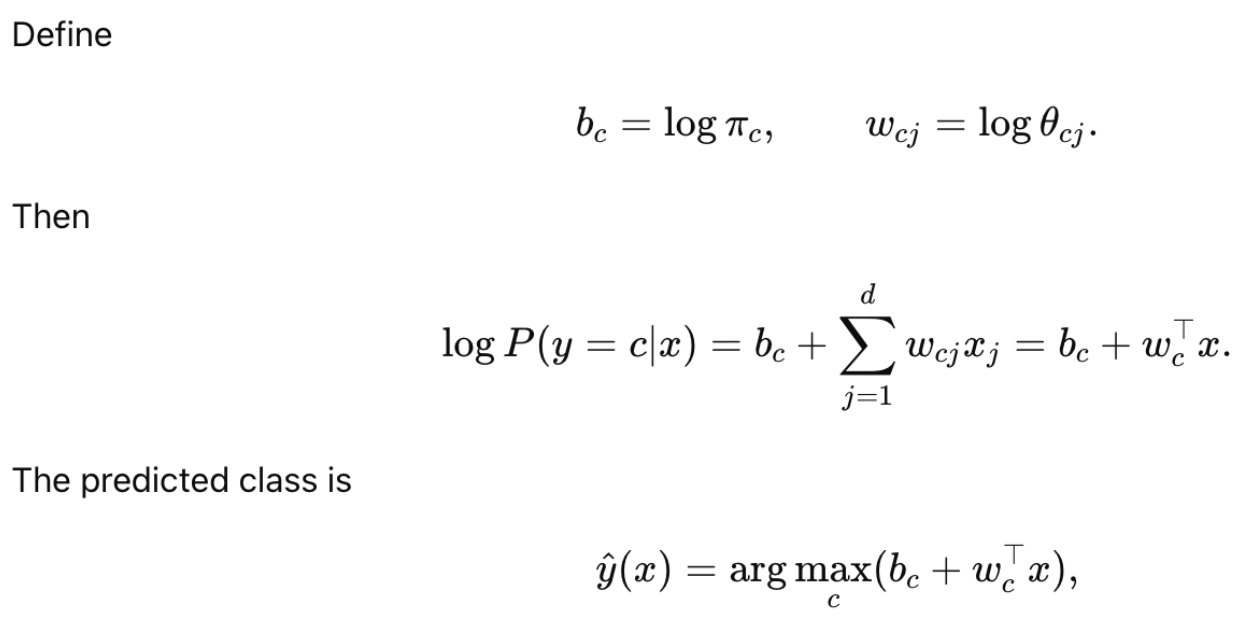
Bayes′ theorem:

$$P(y = c|x) \propto P(y = c)\, P(x|y = c) \propto \pi_c \prod_{j=1}^{d} \theta_{cj}^{x_j}.$$

Taking logs (since the denominator is constant across c):

$$\log P(y = c|x) = \log \pi_c + \sum_{j=1}^{d} x_j \log \theta_{cj} + \text{constant (in x)}.$$

Linear Form



Define

$$b_c = \log \pi_c, \qquad w_{cj} = \log \theta_{cj}.$$

Then

$$\log P(y = c|x) = b_c + \sum_{j=1}^{d} w_{cj} x_j = b_c + w_c^\top x.$$

The predicted class is

$$\hat{y}(x) = \arg\max_c (b_c + w_c^\top x),$$

which is a **linear discriminant function**.

Thus, the **Multinomial Naïve Bayes** classifier is **linear in the input x** when expressed in log-space.

Solution for Question 3.2 - Logistic regression is linear

Binary logistic regression:

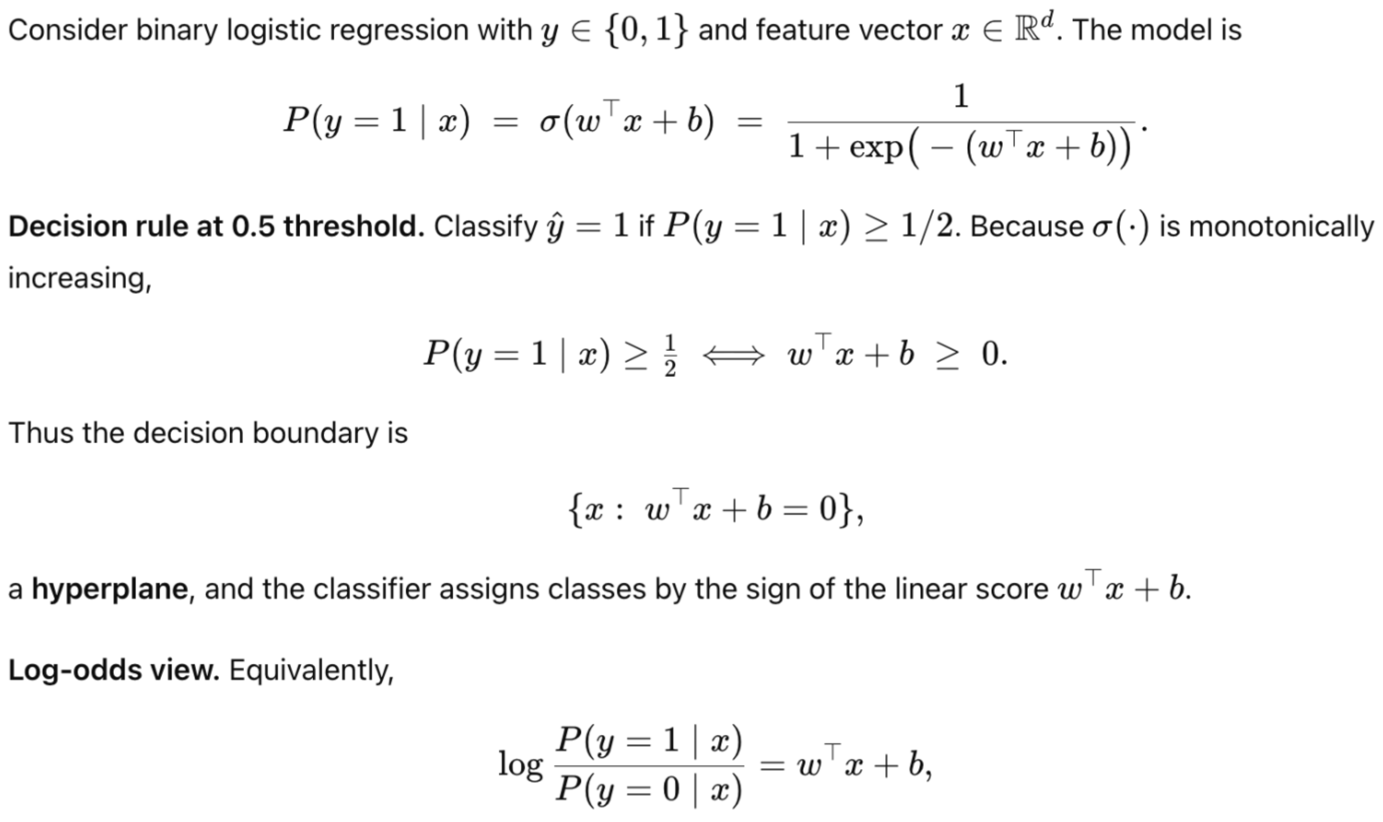
$$P(y = 1 \mid x) = \sigma(w^\top x + b) = \frac{1}{1 + \exp\left(-(w^\top x + b)\right)}.$$

Using the 0.5threshold: $$P(y = 1 \mid x) \geq \tfrac{1}{2} \iff w^\top x + b \geq 0,$$

so the decision boundary $w^\top x + b = 0$is a **hyperplane**. Equivalently, the log-odds is linear:

$$\log \frac{P(y = 1 \mid x)}{P(y = 0 \mid x)} = w^\top x + b.$$

Thus logistic regression is a **linear classifier** (and multiclass softmax uses linear class scores $w_c^\top x + b_c$with decision $\arg\max_c$, also linear).



Consider binary logistic regression with $y \in \{0,1\}$ and feature vector $x \in \mathbb{R}^d$. The model is

$$P(y=1 \mid x) \;=\; \sigma(w^\top x + b) \;=\; \frac{1}{1+\exp\big(-(w^\top x + b)\big)}.$$

Decision rule at 0.5 threshold. Classify $\hat{y}=1$ if $P(y=1 \mid x) \geq 1/2$. Because $\sigma(\cdot)$ is monotonically increasing,

$$P(y=1 \mid x) \geq \tfrac{1}{2} \iff w^\top x + b \;\geq\; 0.$$

Thus the decision boundary is

$$\{x:\; w^\top x + b = 0\},$$

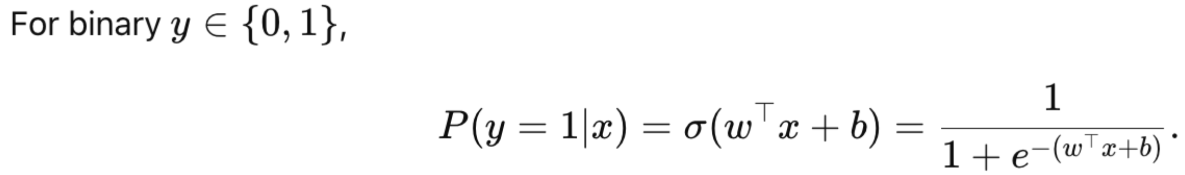
a **hyperplane**, and the classifier assigns classes by the sign of the linear score $w^\top x + b$.

Log-odds view. Equivalently,

$$\log \frac{P(y=1 \mid x)}{P(y=0 \mid x)} = w^\top x + b,$$

Theory

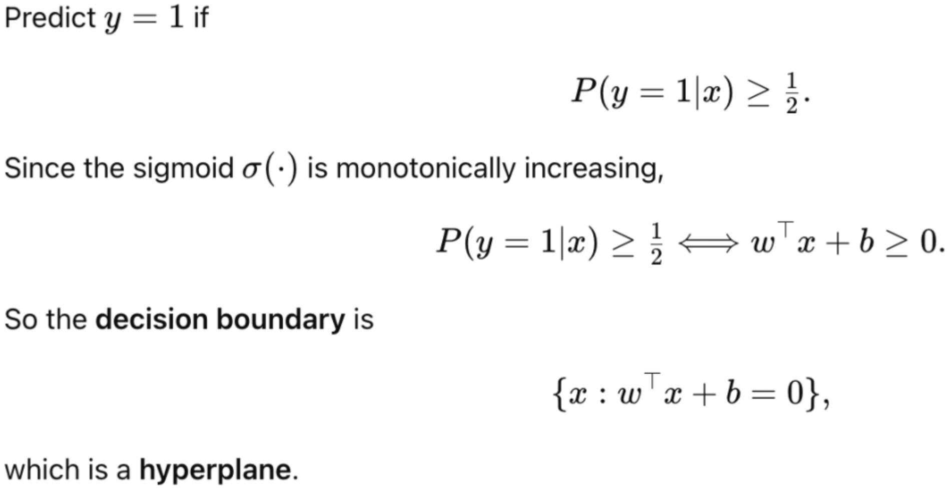
Model Setup



For binary $y \in \{0,1\}$,

$$P(y=1|x) = \sigma(w^\top x + b) = \frac{1}{1+e^{-(w^\top x+b)}}.$$

Decision Rule



Predict $y=1$ if

$$P(y=1|x) \geq \tfrac{1}{2}.$$

Since the sigmoid $\sigma(\cdot)$ is monotonically increasing,

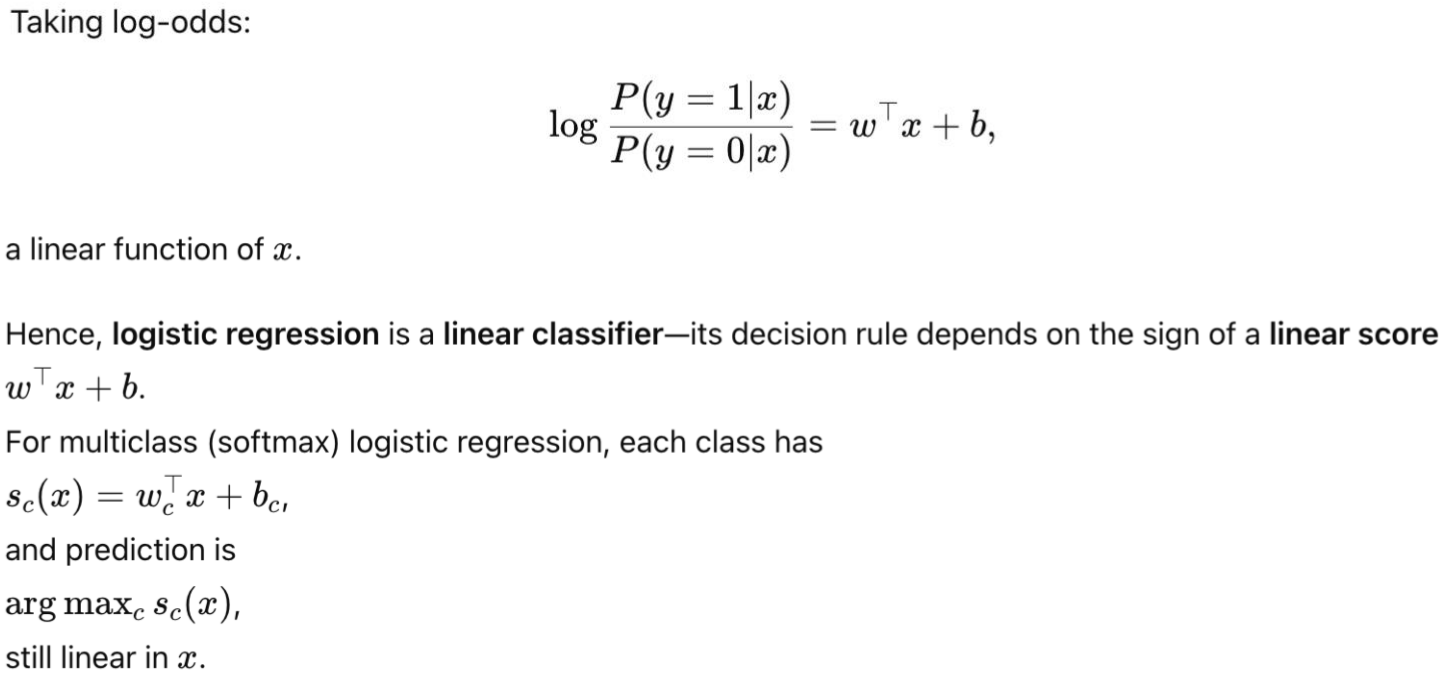
$$P(y=1|x) \geq \tfrac{1}{2} \iff w^\top x + b \geq 0.$$

So the **decision boundary** is

$$\{x : w^\top x + b = 0\},$$

which is a **hyperplane**.

Log-Odds Form



Taking log-odds:

$$\log \frac{P(y=1|x)}{P(y=0|x)} = w^\top x + b,$$

a linear function of x.

Hence, **logistic regression** is a **linear classifier**—its decision rule depends on the sign of a **linear score** $w^\top x + b$.
For multiclass (softmax) logistic regression, each class has
$s_c(x) = w_c^\top x + b_c$,
and prediction is
$\arg\max_c s_c(x)$,
still linear in x.

Conclusion:
Both binary and multiclass logistic regression use linear decision boundaries,
making them linear classifiers in the input space.