Chapter 4: Classification

Question 4

(a) Answer

p = 1 (one feature, uniformly distributed on [0, 1])

We're predicting a test observation's response using only those training observations within 10% of the range of X. Since the range is from 0 to 1, 10% = 0.1. So, we look ±0.05 from the test point.

This interval has width 0.1. Because the data is **uniformly distributed**, the proportion of the total data within any subinterval is equal to the length of that interval.

→ Fraction used = 0.1 or 10%

(b) Answer

p = 2 (two features, each uniformly distributed on [0, 1])

Now we use training observations that are within 10% of the range of **both** features, so:

- For X_1: within ±0.05 → width = 0.1
- For X_2: within ±0.05 → width = 0.1

This defines a **square region** of area:
0.1×0.1=0.01

→ Fraction used = 0.01 or 1%

(c) Answer

p = 100 (100 features, all uniformly distributed on [0, 1])

Each feature range is restricted to 10% (i.e., an interval of 0.1). So, we are only keeping observations within a hypercube of volume:

$$0.1^{100} = 10^{-100}$$

This is an **extremely tiny fraction** — essentially zero in practice.

Fraction used effectively **0%**

(d) Answer

Implication of (a)–(c): Curse of Dimensionality

These results show that as the number of dimensions **p increases**, the **fraction of nearby points drops off exponentially**. In high dimensions:

- Almost no points are "close" to any test observation.
- Local methods like KNN **have too few useful neighbors** unless **n (sample size) is enormous**.
- Distance metrics become less meaningful: all points are roughly the same distance apart.

Conclusion: KNN and similar non-parametric methods perform poorly when p is large unless we have a **massive dataset** to compensate.

(e) Answer

Goal: build a p-dimensional hypercube that contains 10% of the data

We want to find the **length of each side** of the hypercube (denoted lll) such that:

$$l^p = 0.1 \Rightarrow l = 0.1^{1/p}$$

For different values of p:

- **p = 1**:

$$l = 0.1^{1/1} = 0.1$$

- **p = 2**:

$$l = 0.1^{1/2} = \sqrt{0.1} \approx 0.316$$

- **p = 100**:

$$l = 0.1^{1/100} \approx 0.977$$

Interpretation:

- For **p = 100**, we need a **hypercube almost as big as the entire space (side length ≈ 0.977)** just to capture **10%** of the data.
- That means, to get enough nearby data points, we end up including **almost the entire dataset**, defeating the purpose of local methods.

Question 5

(a) Answer

If the Bayes decision boundary is linear, do we expect LDA or QDA to perform better:

- **On the training set?**
  QDA will typically perform **better** because it's more flexible and can overfit the training data, even if the true boundary is linear.
  This is because QDA estimates a separate covariance matrix for each class, allowing it to adapt more to the training data.

- **On the test set?**
  LDA will usually perform **better** in this case.
  Since the Bayes boundary is linear, LDA matches the true structure and will generalize better, whereas QDA may overfit, especially with small to moderate sample sizes.

(b) Answer

If the Bayes decision boundary is non-linear, do we expect LDA or QDA to perform better:

- **On the training set?**
  Again, **QDA** will typically perform **better**, because it can model curved (non-linear) boundaries and thus better fit the training data.

- **On the test set?**
  QDA is also expected to perform **better**, assuming **enough data is available** to estimate class-specific covariance matrices accurately.
  If the sample size is **too small**, QDA may overfit, and LDA may generalize better despite being mis-specified.

(c) Answer

As the sample size **n increases**, how does the **test accuracy of QDA relative to LDA** change?

QDA improves relative to LDA as n increases.

Why?

- QDA estimates more parameters (one covariance matrix per class), so it needs more data to be effective.

- As sample size increases, QDA's **variance decreases**, and it can **take advantage of its lower bias** (especially in non-linear settings).
- Therefore, QDA becomes more accurate relative to LDA **as n → large**.

(d) Answer

False

Why?

- QDA can indeed represent a linear boundary **as a special case**, but it estimates **more parameters** than LDA, including separate covariance matrices.
- When the **true boundary is linear**, QDA introduces **unnecessary variance** by estimating extra parameters.
- This can **hurt generalization** and lead to **worse test performance**, especially with small/moderate n.
- In contrast, LDA is **simpler and better matched** to the data-generating process in this case.

Question 6

The logistic model is:

$$\hat{p}(X) = \frac{1}{1 + e^{-(\widehat{\beta_0} + \widehat{\beta_1} X_1 + \widehat{\beta_2} X_2)}}$$

(a) Estimate probability: 40 hours studied, GPA = 3.5

Plug it in or have Python or a calculator do it for you to get: 0.3775

(b) How many hours needed for 50% chance (GPA = 3.5)?

Set p=0.5 and solve for x. After some algebra you get 50 hours

Question 8

Logistic Regression

- **Training error:** 20%
- **Test error:** 30%

This indicates:

- **Low variance**: The model doesn't overfit much (training and test errors aren't too far apart).
- **Some bias**: The 20% training error means the model doesn't perfectly capture the underlying data structure.

1-Nearest Neighbor (K=1)

- **Average error (train + test):** 18%
  (implies training error is very low — almost 0%, and test error is higher)

Because **K=1** memorizes the training set:

- **Training error is ~0%** (since each point is its own nearest neighbor)
- **Test error must be ~36%** to average out to 18% overall
  (i.e., if training error ≈ 0%, then test error≈2×18%=36%

This suggests:

- **Very high variance**: Model overfits to training data
- **Low bias**, but **poor generalization**

Conclusion: Prefer Logistic Regression

Even though 1-NN has a better average error rate, its **test performance is worse** (≈36% vs. 30% for logistic regression), and its tendency to **overfit** makes it unreliable for classifying new observations.

Key takeaway:

- We care most about **test performance**, because that reflects how the model performs on unseen data.
- **Logistic regression generalizes better**, even if its training error is higher.

Use logistic regression, because it strikes a better balance between bias and variance and performs better on new, unseen data.

Question 9

Recall:

$$\textbf{Odds} = \frac{p}{1-p}$$

Probability = $\frac{\text{odds}}{1+\text{odds}}$

(a)

Given: Odds of default = **0.37**
We want to find the **probability** that a person will default.

$$p = \frac{0.37}{1 + 0.37} = \frac{0.37}{1.37} \approx 0.27$$

Answer: ~27% of such people will actually default.

(b)

Given: Probability of default = **16%** or **0.16**
We want to find the **odds**.

$$\text{odds} = \frac{0.16}{1 - 0.16} = \frac{0.16}{0.84} \approx 0.19$$

Answer: The odds of default are approximately **0.19**