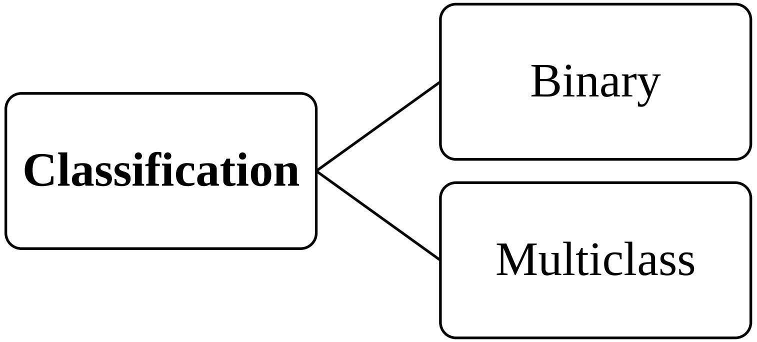
Seventh Session

Alireza Moradi

Classification

Classification

- In Classification problem, Y takes values in a finite, unordered set. In other words, our target variable is Nominal (or Binary).



E.g., default= {yes, no}

E.g., eye color = {Brown, Blue, Green}

- **Ordinal classification** is a form of multiclass classification for which there is an inherent order between the classes, but not a meaningful numeric difference between them. In other words, target variable is Ordinal.

Classification

- The function should predict the class label of an observation based on its features (X). It must put the observation into one of the existing classes.

$$\begin{tikzcd} X \arrow[r, Rightarrow] & \boxed{\text{Function}} \arrow[r, Rightarrow] & Y;\ \text{Class} \end{tikzcd}$$

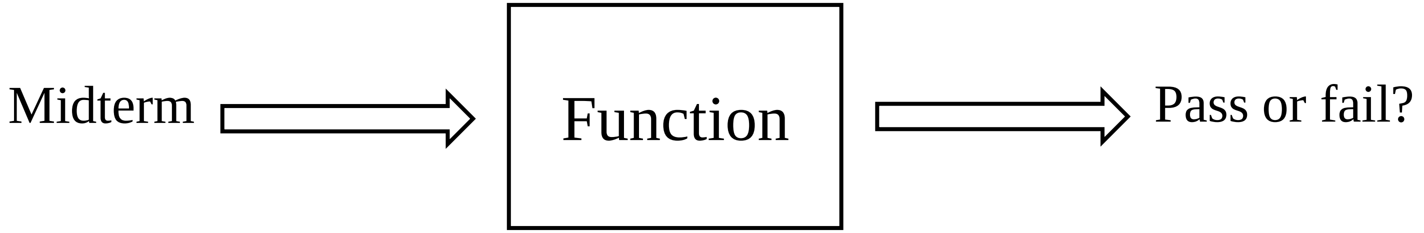
- Classification problems occur often, perhaps even more so than regression problems.

Classification

- Example:

Based on the student's midterm score, can they be expected to pass the term?

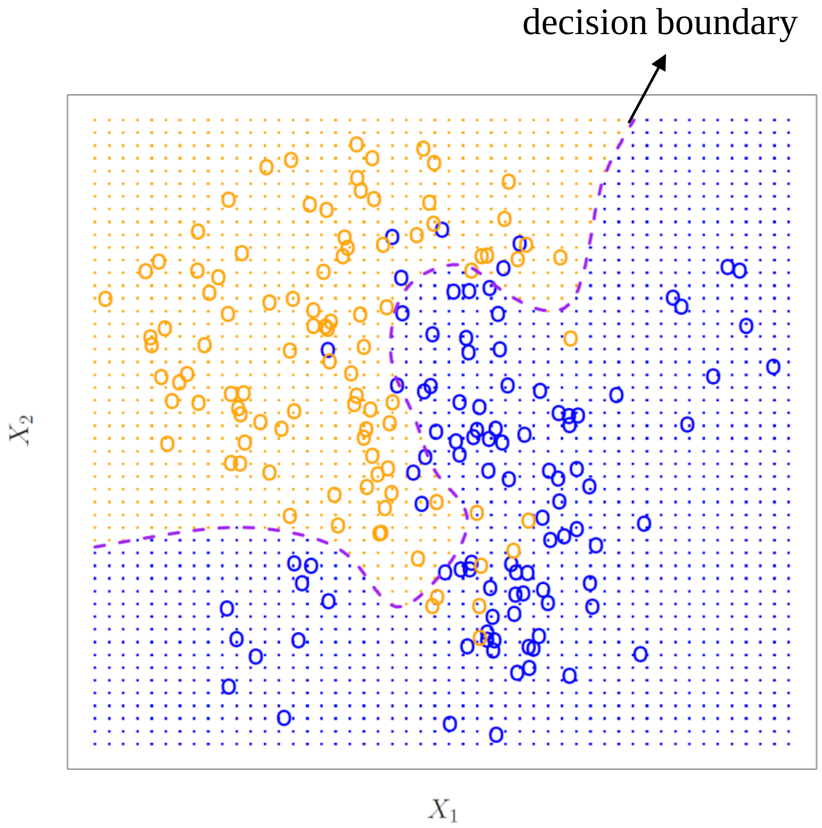
Observation	Midterm	Pass?
1	15	pass
2	13	Fail
3	11	pass
4	3	Fail
5	6	Fail
6	20	pass
7	19	pass
8	20	pass
9	17	pass
10	16	pass



Classification

- A **decision boundary** is a line (in the case of two features), where all (or most) samples of one class are on one side of that line, and all samples of the other class are on the opposite side of the line.

- A **classifier** is the algorithm itself – the rules used by machines to classify data.

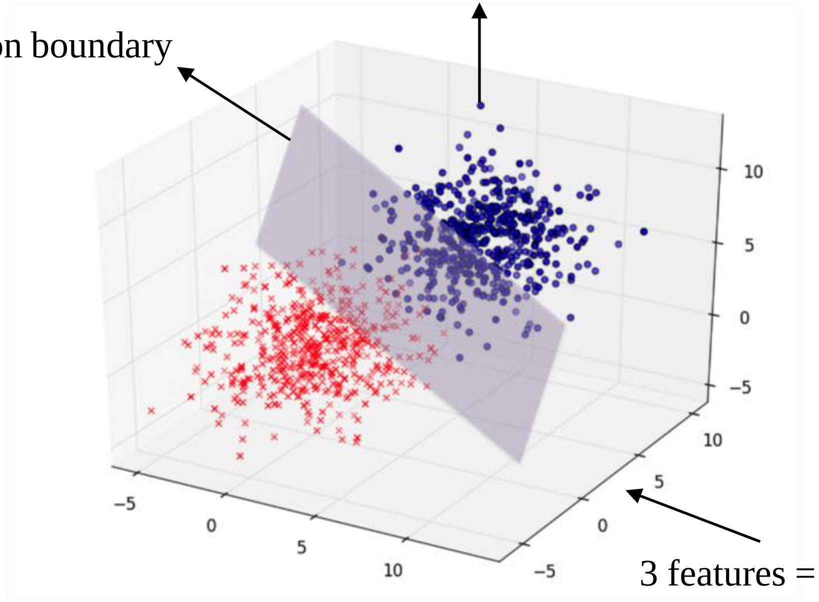


Classification

- A decision boundary is a hypersurface in feature space that separates classes.

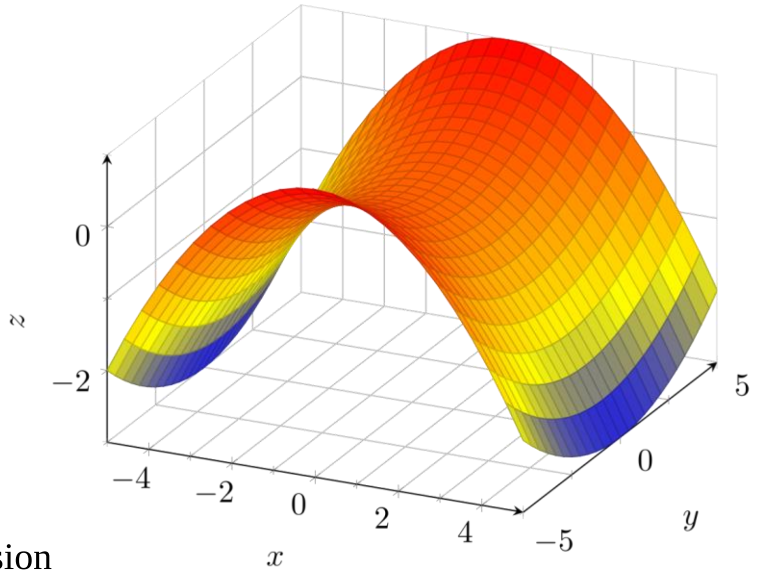
Two colors = binary classification

decision boundary



3 features = 3 dimension

Hypersurface in 3-dimensional space



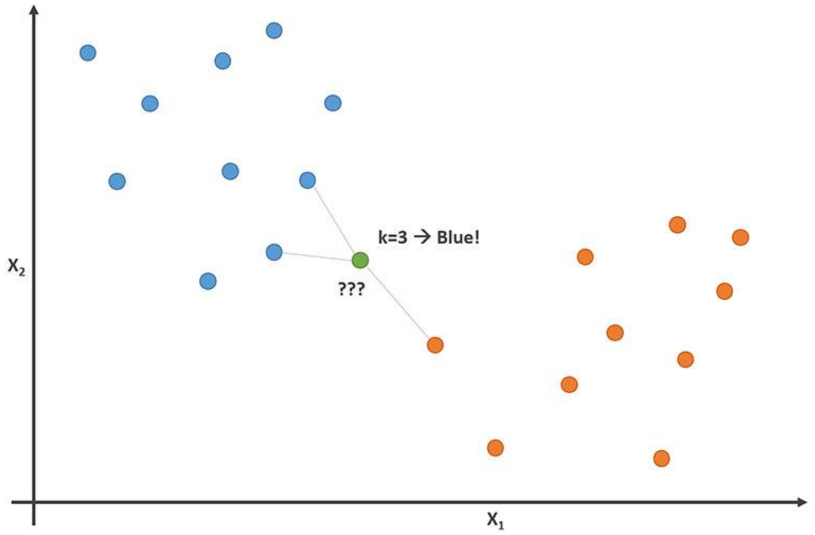
K-nearest neighbors

K-nearest neighbors (KNN)

- The **k-nearest neighbors' algorithm**, also known as **KNN** or **k-NN**, is a classifier which uses proximity to make classifications.

- Main idea:

- The **k value** in the k-NN algorithm defines how many neighbors will be checked to determine the classification of a specific query point.



K-nearest neighbors (KNN)

- The k-NN algorithm relies on a **distance metric** to measure the similarity between data points. Here's the formula for the commonly used **Euclidean distance**:

$$d(a,b) = \sqrt{\sum_{j=1}^{p} \left(a_j - b_j\right)^2}$$

- $\hat{y}$ is the classifier:

$$\hat{y} = \arg\max_{c} \sum_{i=1}^{k} I(y_i = c)$$

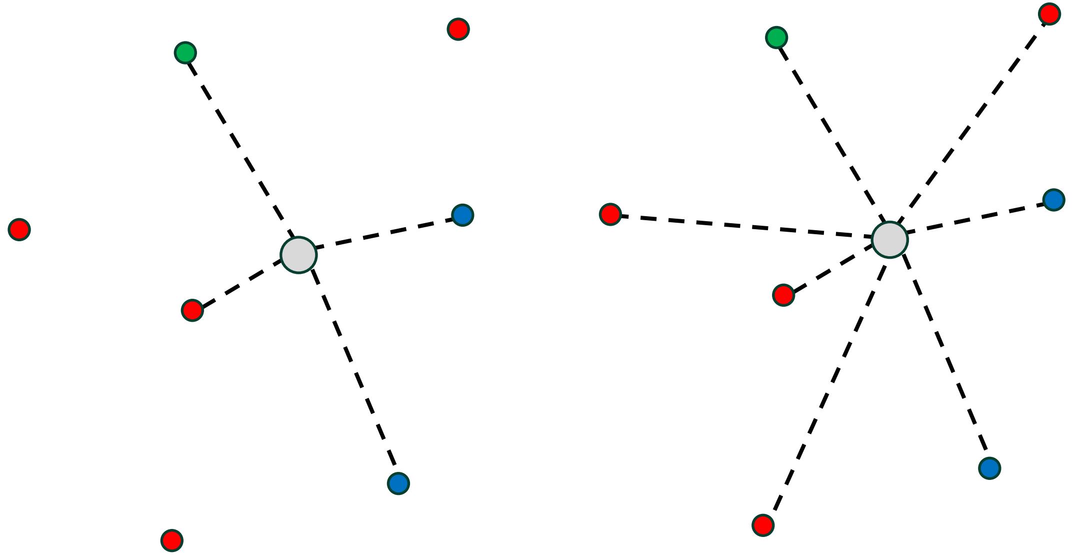
- $I(y_i = c)$ is an indicator function that equals 1 if y_i is equal to class c and 0 otherwise.

K-nearest neighbors (KNN)

- Example:

✓ K = ?

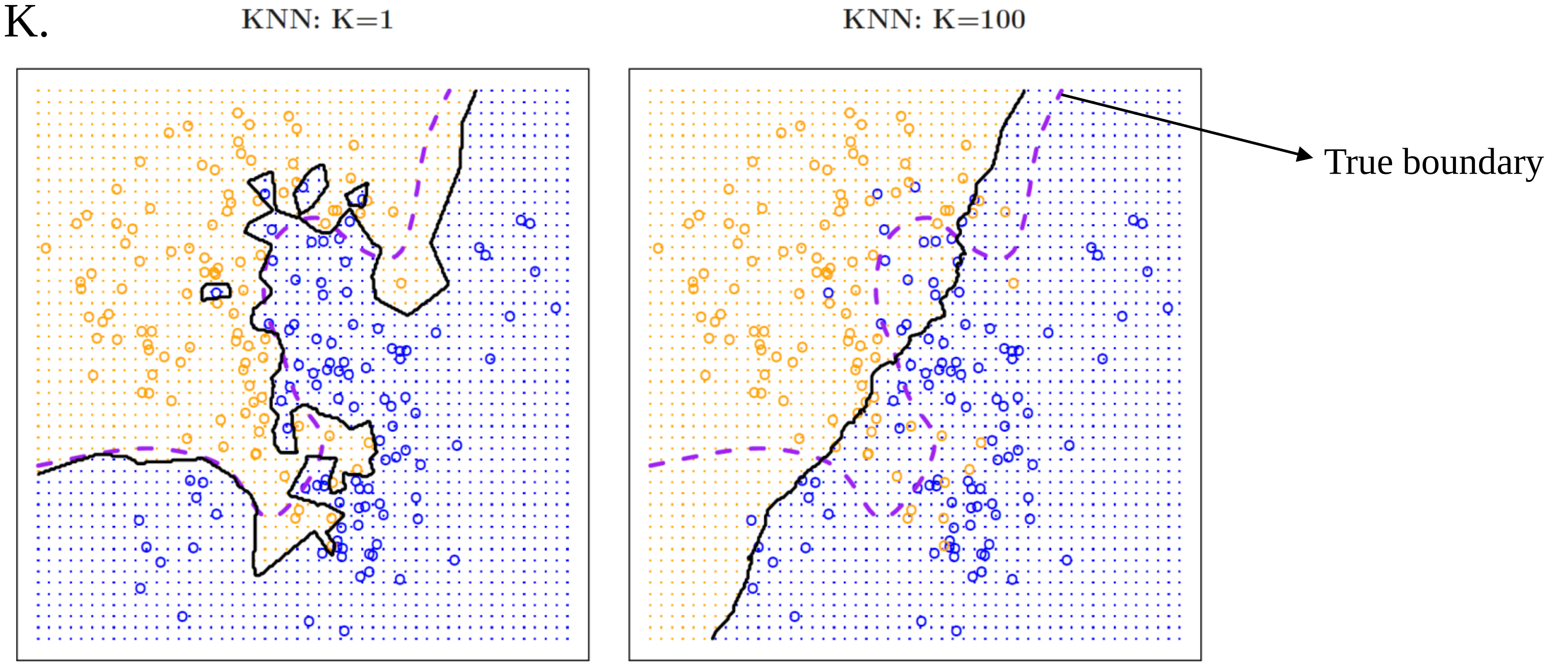
✓ Class prediction = ?



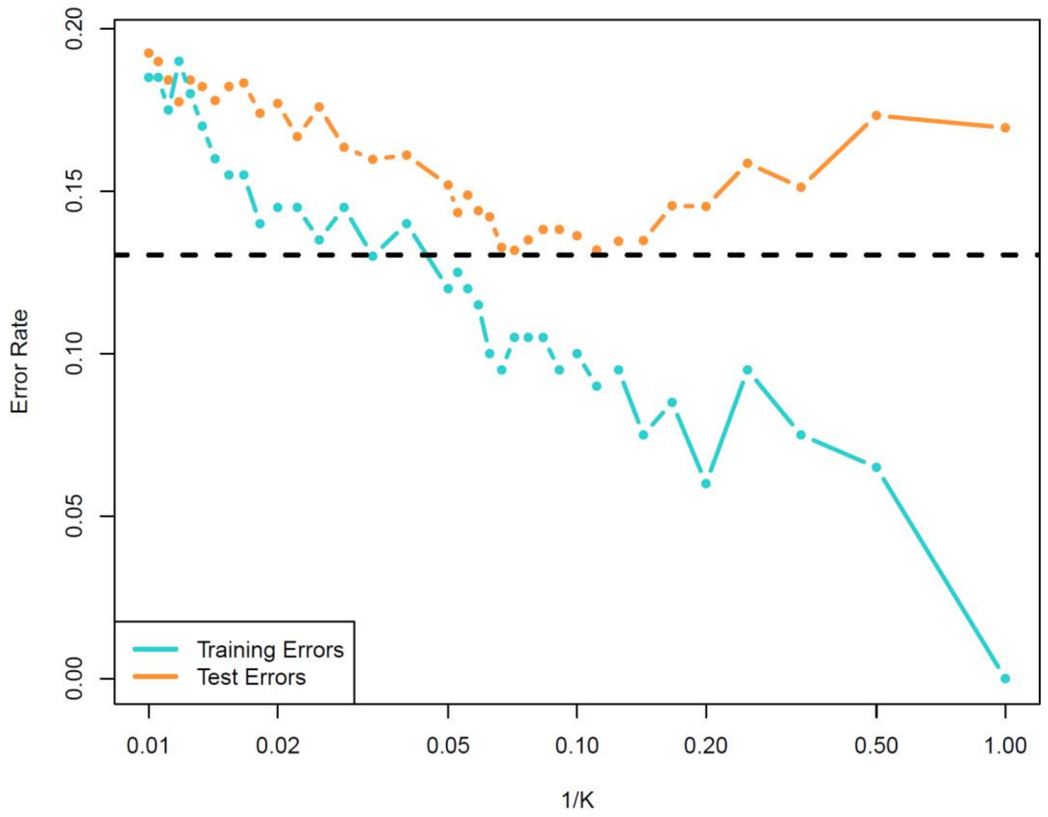
- The choice of K in the K-Nearest Neighbors algorithm can significantly impact the decision boundary.

K-nearest neighbors (KNN)

- Different values of K can lead to varying levels of **smoothness** and **complexity** in the boundary. Here is a comparison of the KNN decision boundaries for different values of K.



K-nearest neighbors (KNN)



Curse of dimensionality

- Nearest neighbor methods can be lousy when p is large.
  Reason: the **curse of dimensionality**

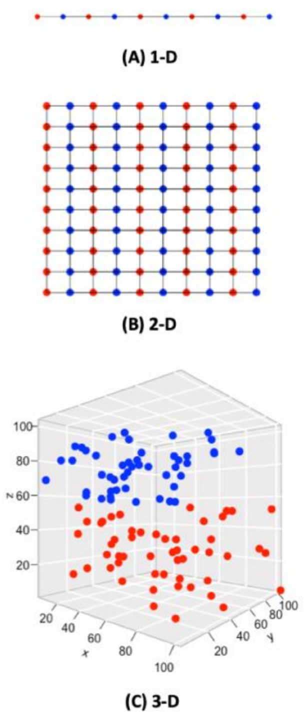
- The **curse of dimensionality** refers to the various challenges and complications that arise when analyzing and organizing data in high-dimensional spaces (often hundreds or thousands of dimensions).

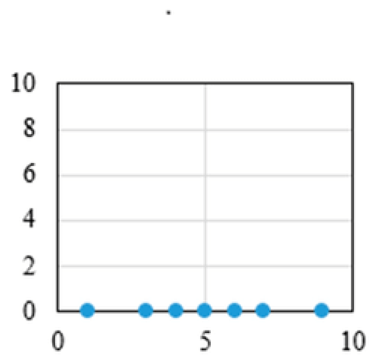
- In KNN, increasing the value of k helps reduce variance. (E.g., 10%) However, in high dimensions, a 10% neighborhood may no longer be considered truly local.

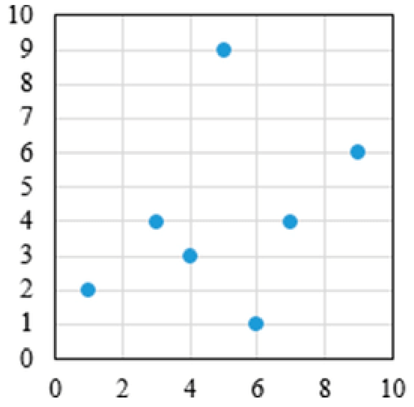
- Nearest neighbors tend to be far away in high dimensions.

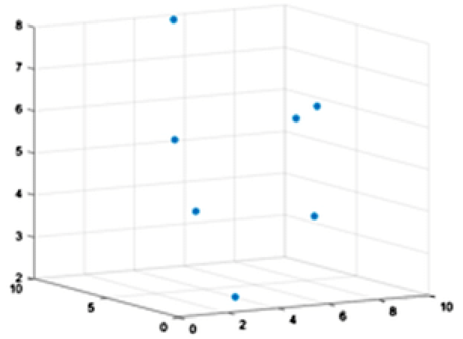
Curse of dimensionality

- Nearest neighbors tend to be far away in high dimensions.

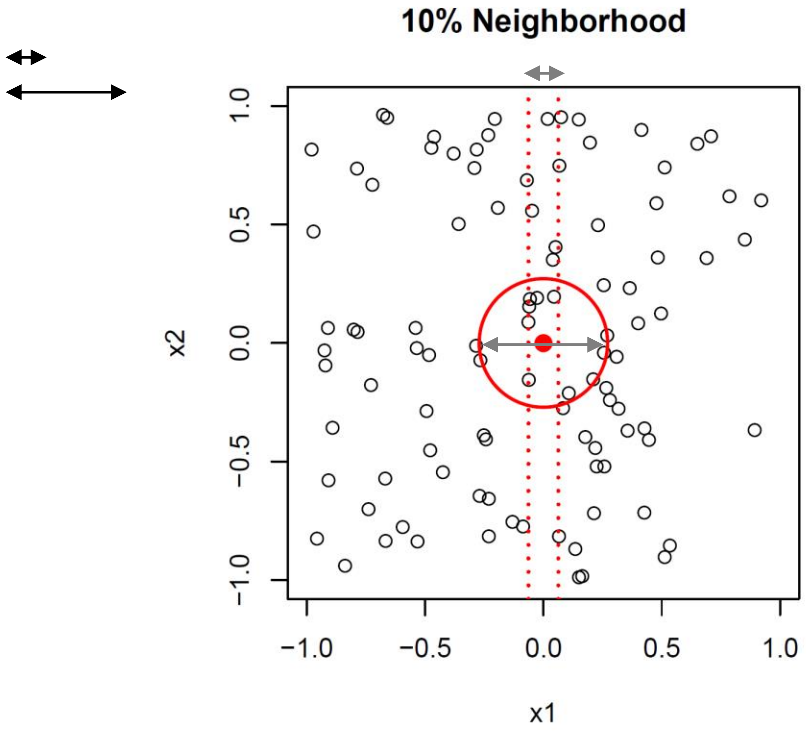




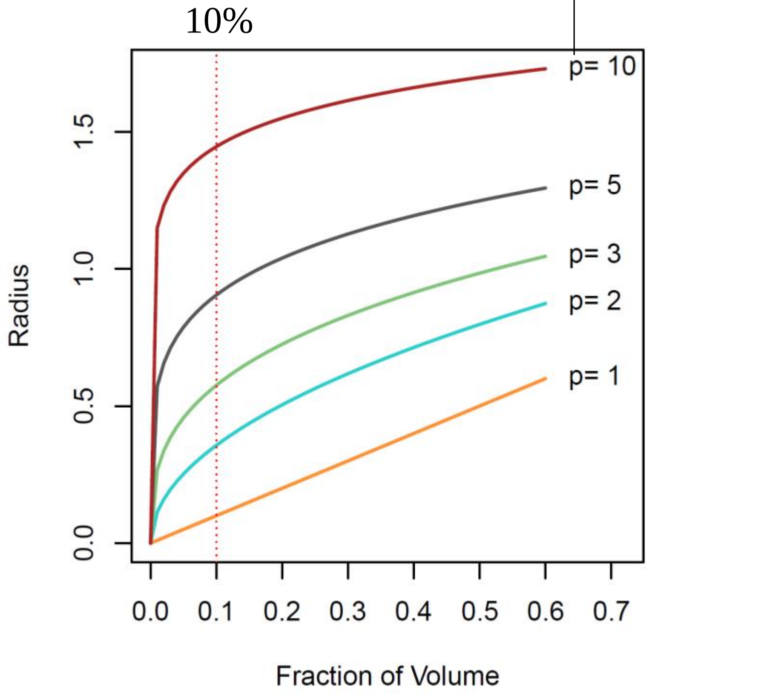




Curse of dimensionality



P = Dimensions = features



Curse of dimensionality

- Major problems that the curse of dimensionality cause:

  1. **Data sparsity:** most of the high-dimensional space is empty.

  2. **Increased computation:** More dimensions mean more computational resources and time to process the data.

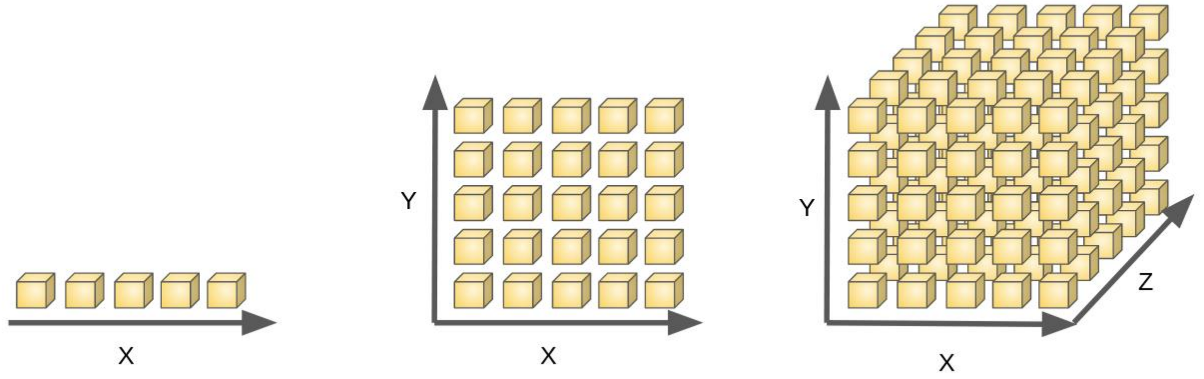
  3. **Overfitting:** With higher dimensions, models can become overly complex, fitting to the noise rather than the underlying pattern.

  4. **Distances lose meaning:** In high dimensions, the difference in distances between data points tends to become negligible, making measures like Euclidean distance less meaningful.

Curse of dimensionality

5. **Performance degradation:** Algorithms, especially those relying on distance measurements like k-nearest neighbors, can see a drop in performance.

6. **Visualization challenges:** High-dimensional data is hard to visualize, making exploratory data analysis more difficult.

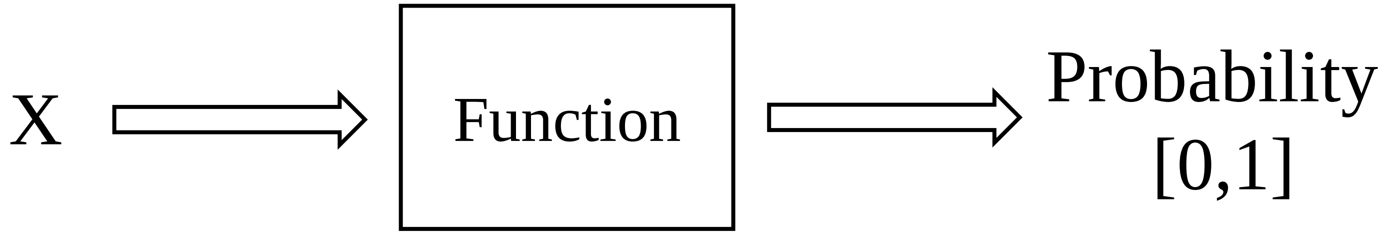


Logistic Regression

Classification; probabilities

- In classification, we are more interested in **estimating the probabilities** that observation belongs to each of the categories. In this sense classification methods also behave like regression methods.(predicting a continuous number)

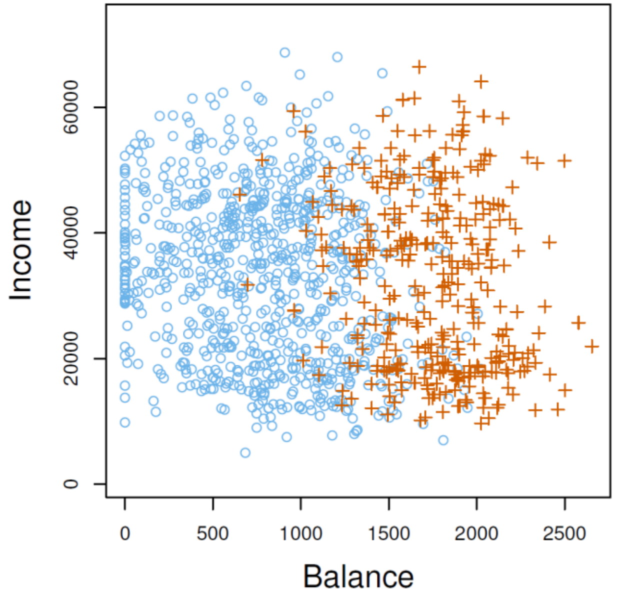
- For example, it is more valuable to have an estimate of the probability that an insurance claim is fraudulent, than a classification fraudulent or not.

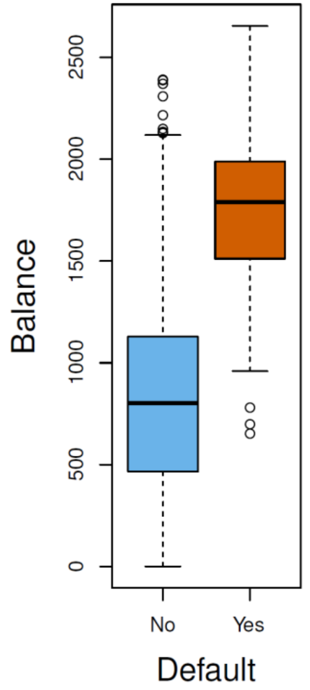


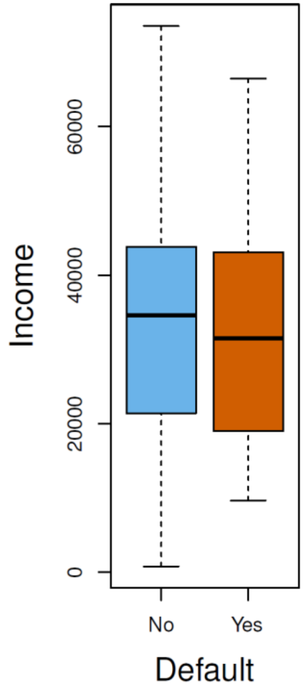
- Probabilities in binary classification => *P* and *1-P*

Classification

Example: Credit card default







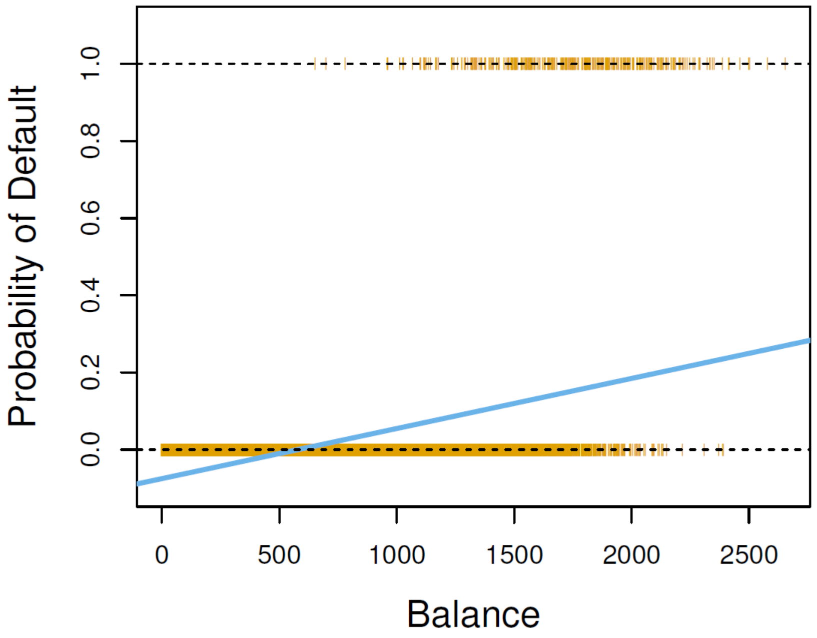
✓ Which variable is more important?

Classification

- Suppose for the Default classification task that we code:

$$Y = \begin{cases} 0 \text{ if No} \\ 1 \text{ if Yes} \end{cases}$$

✓ Can we simply perform a linear regression of Y on X and classify a new observation as Yes if $\hat{y} > 0.5$?



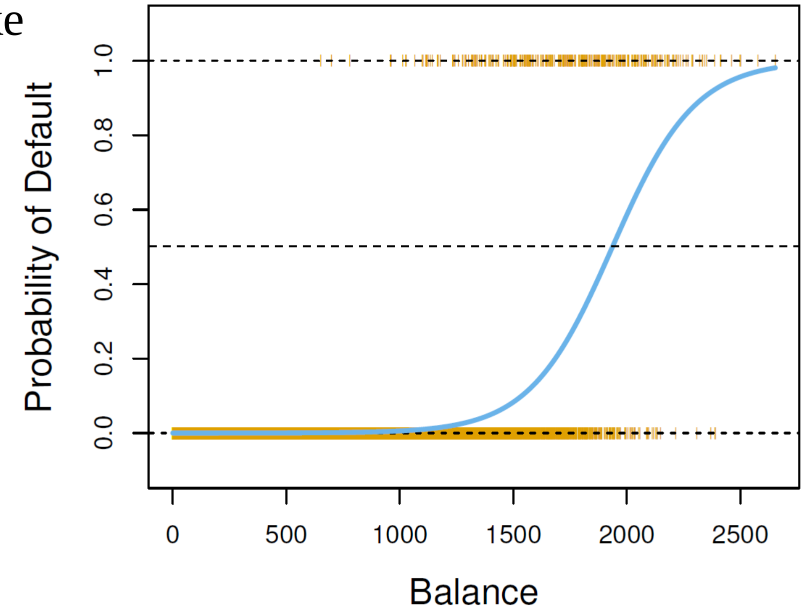
Logistic Regression

- **Logistic Regression**, despite its name, is a widely used machine learning algorithm for binary classification tasks.
- **Logistic regression** is a statistical method for binary classification. The model is like below:

$$P(X) = \frac{e^{\beta_0+\beta_1.X}}{1+e^{\beta_0+\beta_1.X}}$$

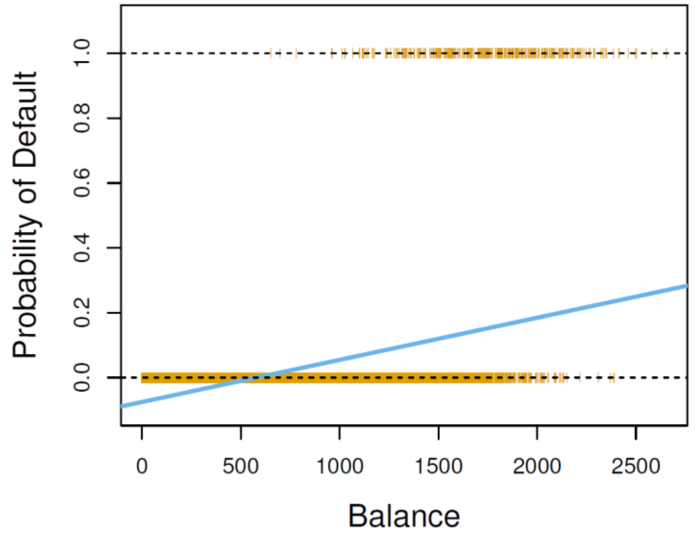
- How we classify observations?

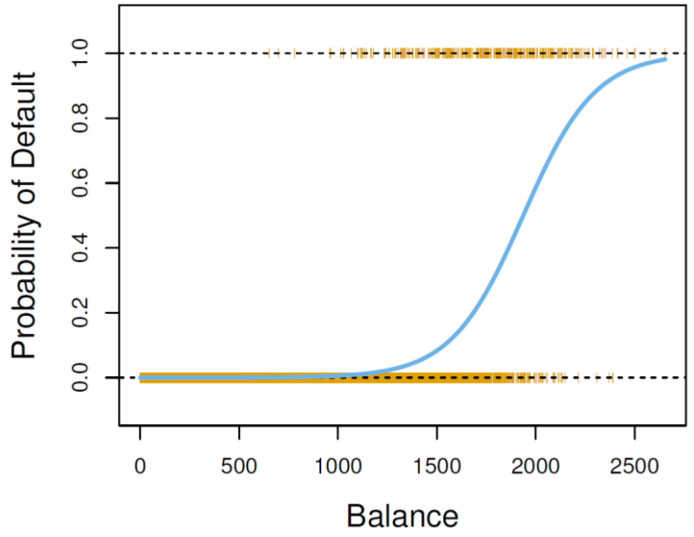
$$if\ P(X) \geq 0{,}5 \rightarrow \hat{y} = 1$$
$$if\ P(X) < 0{,}5 \rightarrow \hat{y} = 0$$



Logistic Regression

- Logistic regression ensures that our estimate for P(X) lies between 0 and 1.





$$P(X) = \beta_0 + \beta_1 . X$$

$$P(X) = \frac{e^{\beta_0 + \beta_1 . X}}{1 + e^{\beta_0 + \beta_1 . X}}$$

Logistic Regression; Multiple features

- Model with 2 features:

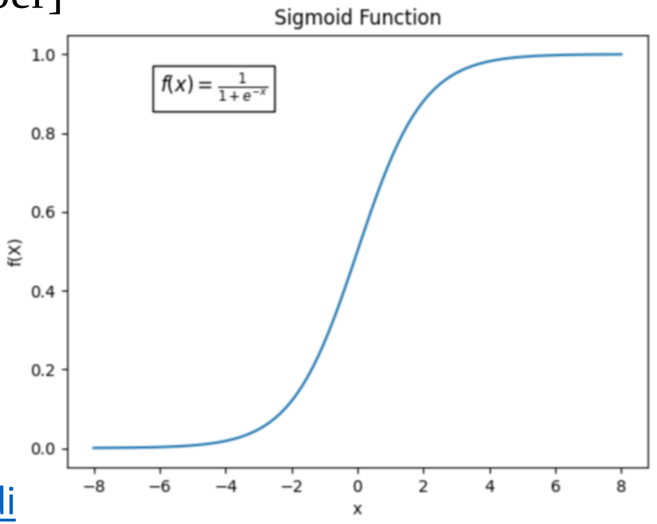
$$P(X) = \frac{e^{\beta_0+\beta_1.x_1+\beta_2.x_2}}{1+e^{\beta_0+\beta_1.x_1+\beta_2.x_2}} = \frac{1}{1+e^{-(\beta_0+\beta_1.x_1+\beta_2.x_2)}}$$

- e 2.71828 is a mathematical constant [Euler's number]

- Logistic regression can also be written as:

$$g(z) = \frac{1}{1+e^{-z}}$$

$$z = \beta_0 + \beta_1 . x_1 + \beta_2 . x_2 \longrightarrow \text{Line(where is it?)}$$



Logistic Regression

- Summarized process:

$$\begin{tikzcd} z = \beta_0 + \beta_1 . x_1 + \beta_2 . x_2 \arrow[r] & \text{Sigmoid function} \arrow[r] & P(y = 1) \ \epsilon \ [0,1] \end{tikzcd}$$

- P(y = 0) + P(y = 1) = 1

Logistic regression; Decision boundary

- In Logistic Regression, Decision Boundary is a linear line, which separates class A and class B.

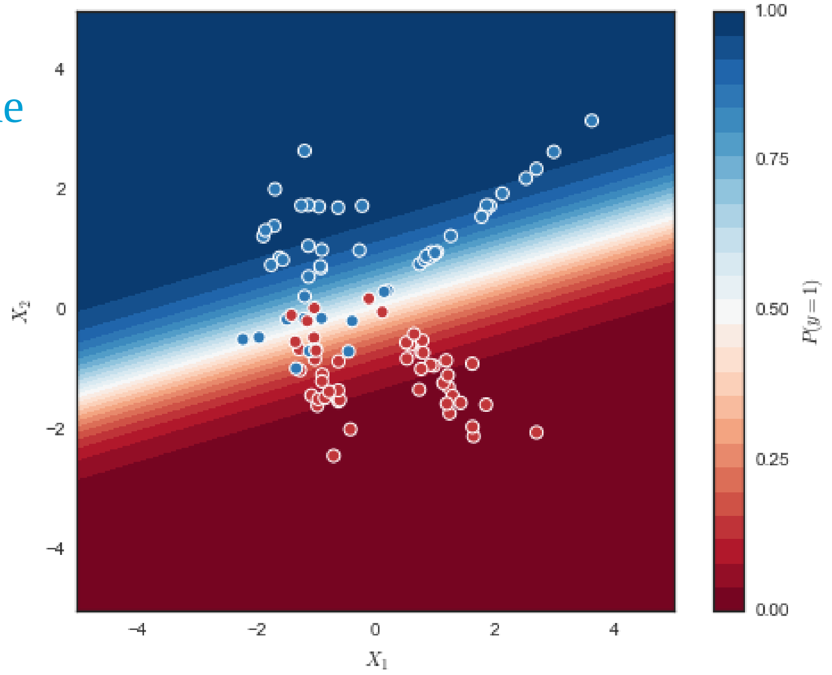
- In fact, the decision boundary is a hyperplane in the features space. **z = 0 is the boundary**.

- For our example (2 feature):

$$z = 0$$

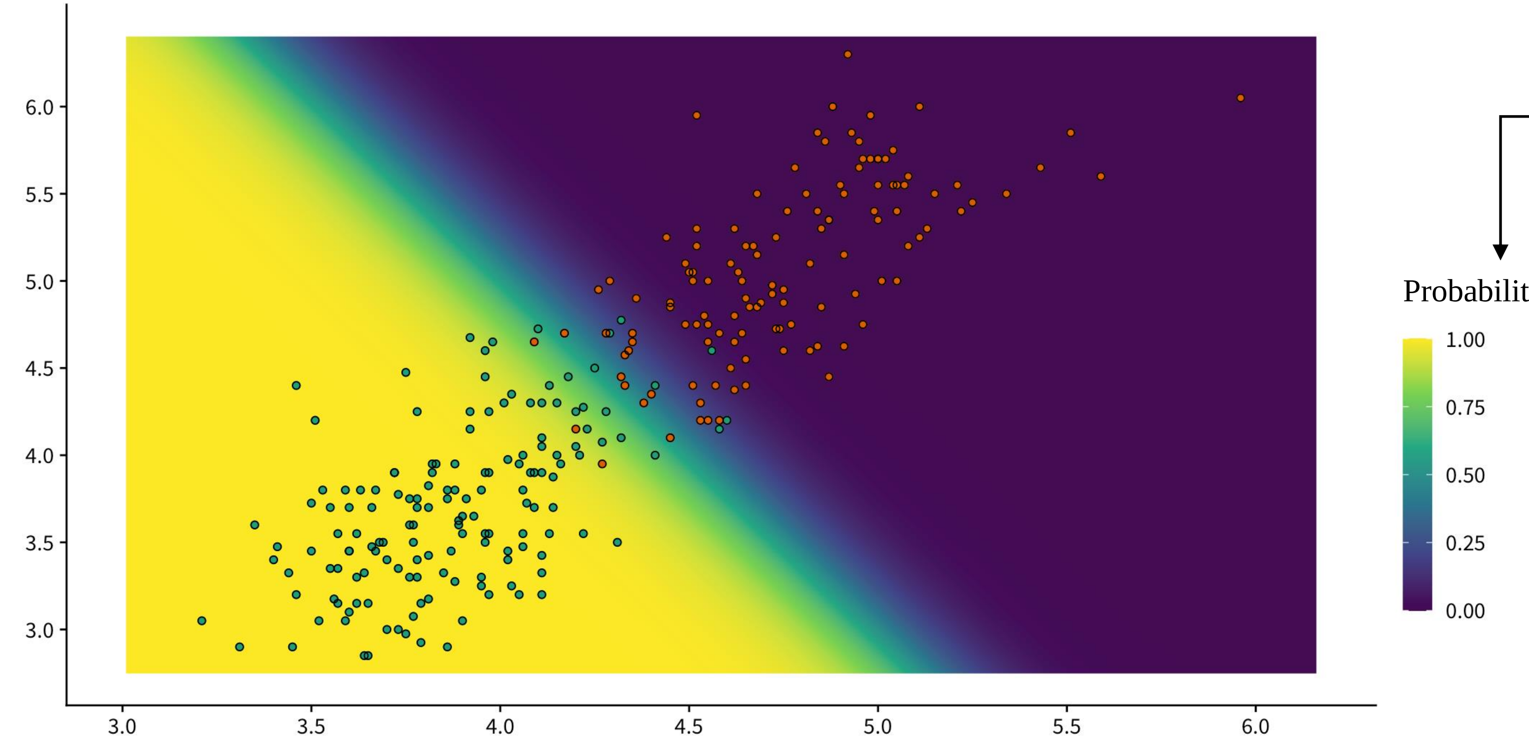
$$\beta_0 + \beta_1 . x_1 + \beta_2 . x_2 = 0$$

Decision boundary



Logistic regression; Decision boundary

class green or red?



Logistic regression with multiple features

- General model for logistic regression with multiple features: (p denotes number of features.)

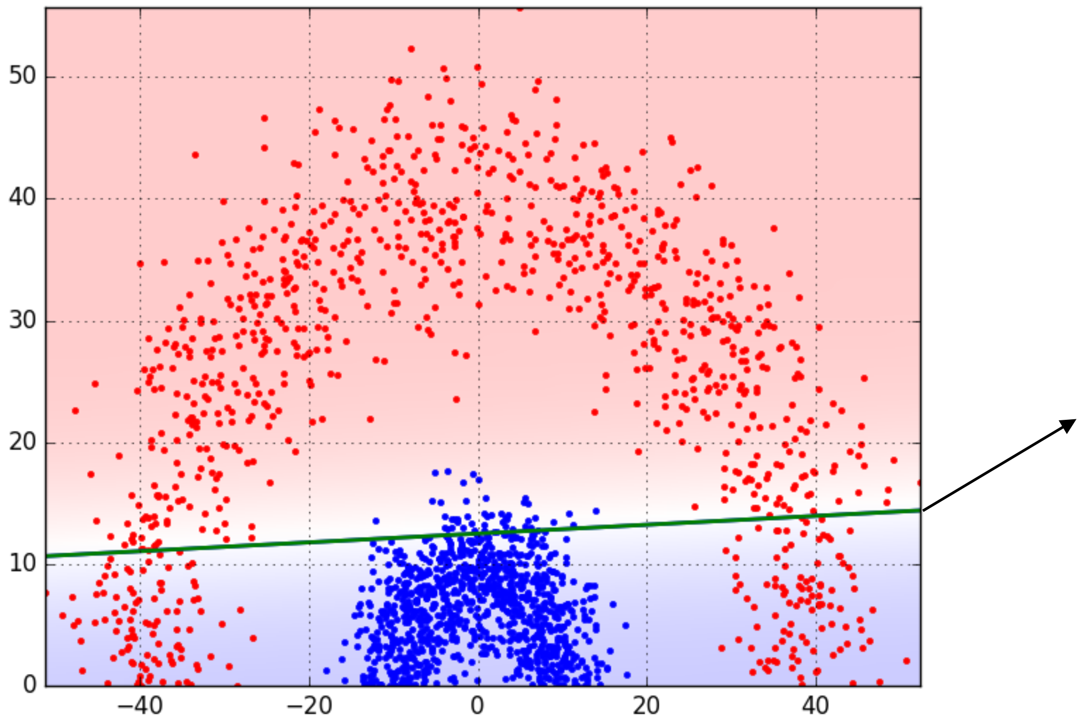
$$z = \beta_0 + \beta_1 . x_1 + \beta_2 . x_2 + \ \dots + \beta_p . x_p$$

$$P(X) = \frac{1}{1 + e^{-(z)}}$$

Observation	Midterm	Statistics score	Number of absence	Pass?
1	15	16	2	pass
2	13	11	7	Fail
3	11	15	0	pass
4	3	15	6	Fail
5	6	12	5	Fail
6	20	18	3	pass
7	19	15	2	pass
8	20	20	1	pass
9	17	16	2	pass
10	16	12	0	pass

Logistic regression; Decision boundary

- ✓ How the decision boundary must be?



Created by Alireza Moradi

Logistic regression; Decision boundary

- Recall the trick in last session. We can do the exact thing here:

- We can introduce non-linearity into z:

$$z = \beta_0 + \beta_1 . x_1 + \beta_2 . x_2 + \beta_3 . x_1^2 + \beta_4 . x_2^2$$

$$P(X) = \frac{1}{1 + e^{-(z)}}$$

Logistic regression; Cost function

- How do we find the best decision boundary?

- This leads us to the question: what are the optimal values for the parameters β_0 to β_p? These values are determined using a **cost function**.

- The cost function for logistic regression, also known as the **binary cross-entropy loss:**

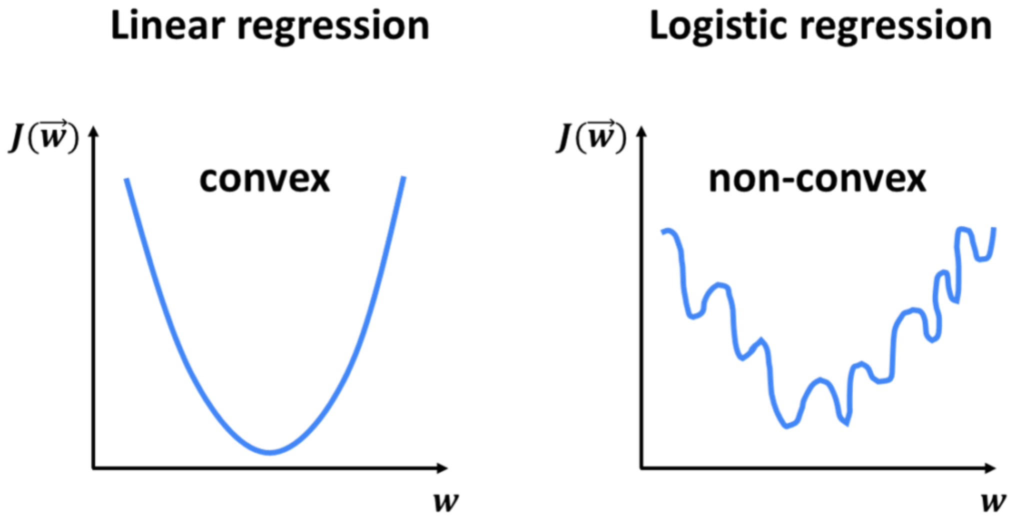
$$J(\beta) = -\frac{1}{n}\sum_{i=1}^{n}\left[y^{(i)}\log\left(P_\beta\left(X^{(i)}\right)\right) + \left(1 - y^{(i)}\right)\log\left(1 - P_\beta\left(X^{(i)}\right)\right)\right]$$

Logistic regression; Cost function

- By minimizing this cost function through optimization algorithms like gradient descent, we can learn the optimal parameters for the logistic regression model.

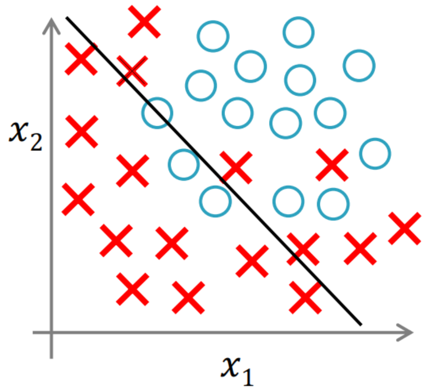
✓ Why didn't we use this cost function?

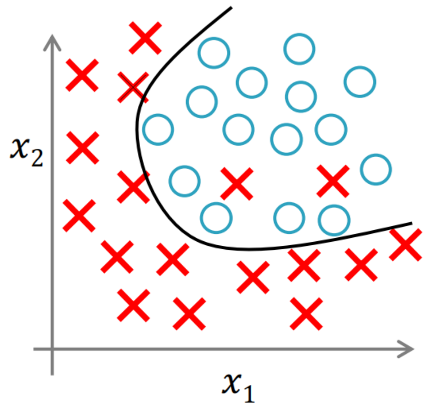
$$J(\beta_0, \beta_1) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

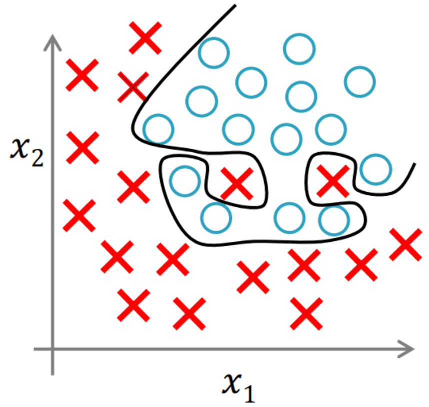


Overfitting in Classification

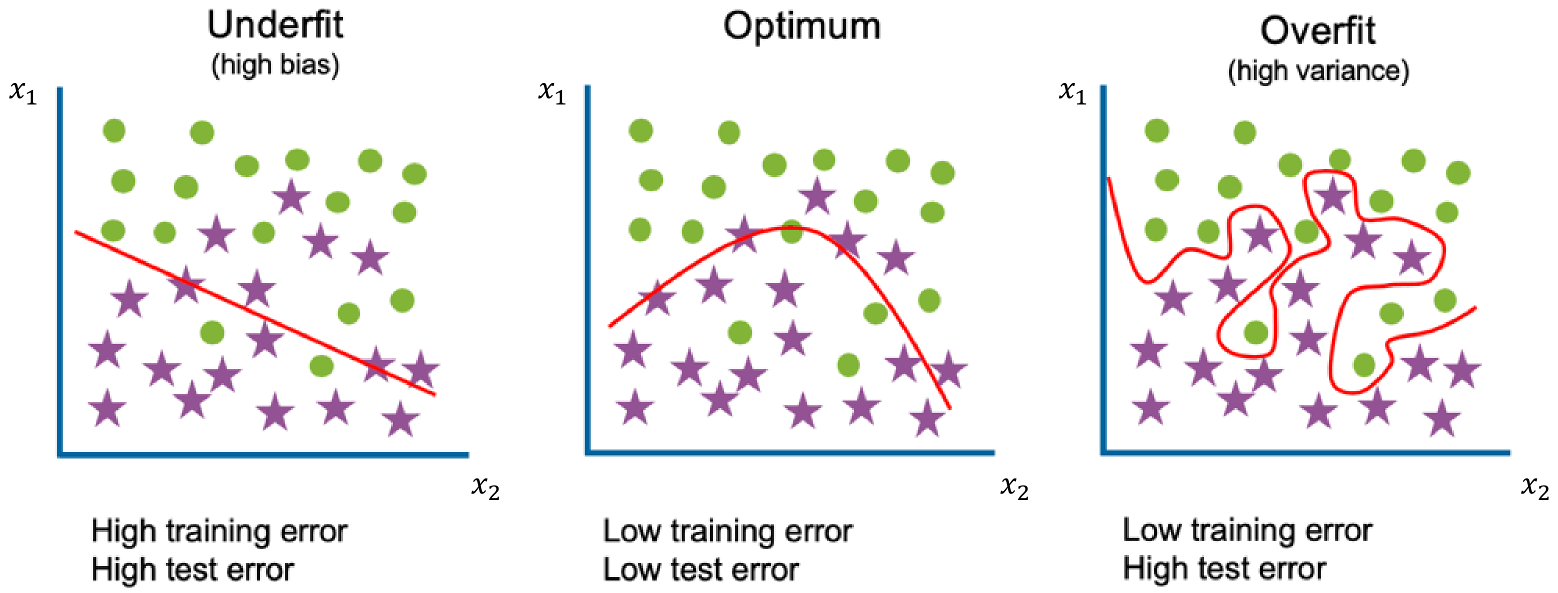
- ✓ Which one is appropriate?
- ✓ Bias and variance?







Overfitting in Classification



$z = \beta_0 + \beta_1 . x_1 + \beta_2 . x_2$

$z = \beta_0 + \beta_1 . x_1 + \beta_2 . x_2 + \beta_3 . x_1^2 + \beta_4 . x_2^2$

$z = \beta_0 + \beta_1 . x_1 + \beta_2 . x_2 + \beta_3 . x_1^2 + \beta_4 . x_2^2 + \beta_5 . x_2^3 + \beta_6 . x_2^2 . x_1 + \beta_7 . x_1^2 . x_2 + \beta_8 . x_2^4$

Overfitting in Classification

- ✓ Which one is overfitted?