

# Machine Learning

Samatrix Consulting Pvt Ltd

# Classification

# Why Classification?

- In many situations the response variable  $Y$  is qualitative.
- Example is eye color, which can be blue, brown, or green.
- We also refer the qualitative variables as categorical too.
- The process or approach for predicting qualitative response is known as **classification**.
- As we have studied in the case of regression, we have a set of training observation  $(x_1, y_1), \dots, (x_n, y_n)$ .
- We can use the training set to build a classifier that can perform well not only on training data but also on the test data.

# Classification Techniques

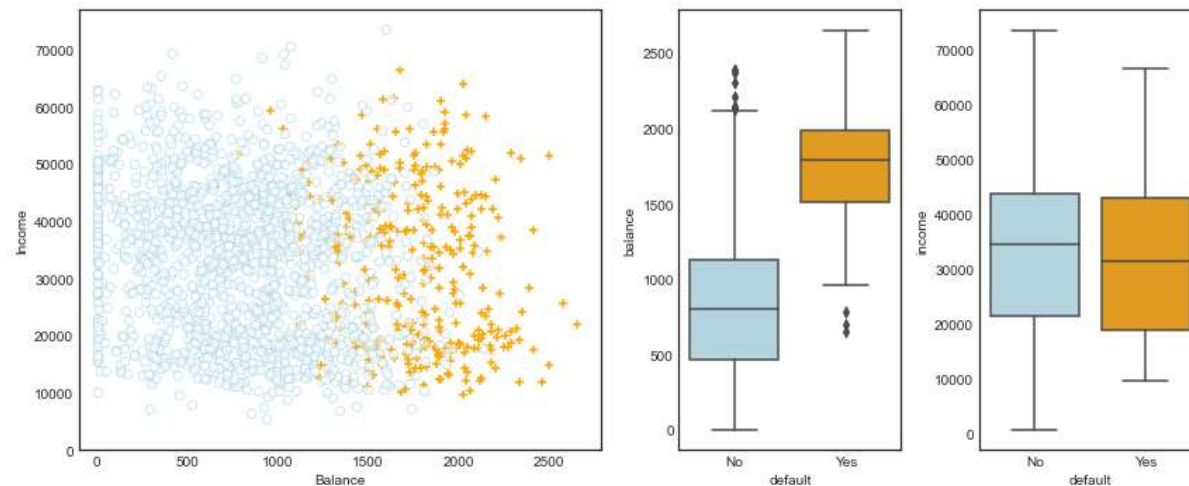
- Many possible classification techniques or classifiers to predict a qualitative response are available.
- The three of the most widely used classifiers are logistic regression, linear discriminant analysis, and K-nearest neighbors.

# Example of Classification Problems

- Classification problems occur more often than regression problems. Some of the examples are
  - A patient reaches the emergency ward of a hospital. He has a set of symptoms. These symptoms could be attributed to one of three medical conditions. Which of the three medical conditions the patient may have?
  - A payment gateway service wants to determine whether a particular transaction that has been performed on the website is fraudulent on the basis of user's IP address, past transaction history, so forth.

# Case Study – Credit Card Default

- We will try to understand the concepts of classification using a case study.
- The objective of the case study is to predict whether an individual will default on his or her credit card balance, on the basis of annual income and monthly credit card balance.



# Case Study - Credit Card Default

- The diagram above shows plot of the annual income and monthly credit card balance for a subset of 10,000 individual.
- In the left-hand panel, the data for individuals who defaulted in a given month has been displayed in orange and who did not in blue.
- From the plot we may infer that individuals who defaulted tended to have higher credit card balances than those who did not.
- In the right-hand side panel, we have shown two pairs of boxplots.
- The first box plot shows the distribution of balance that has been split by the binary default variable.
- The second plot shows the similar distribution for income.
- In this case study, we shall learn, how can we build a model to predict default ( $Y$ ) for a given value of balance ( $X_1$ ) and income ( $X_2$ ). Since the response variable, default, is not quantitative, the use of simple linear regression will not be appropriate.

# Why not linear regression?

- Why the linear regression is not appropriate model for qualitative response?
- Suppose the research goal of a study is to predict the medical condition of a patient in the emergency ward based on the symptoms.
- In this example three possible diagnoses would be, stroke, drug overdose, and epileptic seizure.
- If we encode these values as the quantitative response variable  $Y$  we get

$$Y = \begin{cases} 1 & \text{if stroke} \\ 2 & \text{if drug overdose} \\ 3 & \text{if epileptic seizure} \end{cases}$$



# Why not linear regression?

- This coding means that there is an ordering on the outcome whereas drug overdose is placed between stroke and epileptic seizure.
- If also means that the difference between stroke and drug overdose is same as the difference between drug overdose and epileptic seizure.
- In the practical scenario, this is not the case.
- On the other hand, we may also choose some other coding

$$Y = \begin{cases} 1 & \text{if epileptic seizure} \\ 2 & \text{if stroke} \\ 3 & \text{if drug overdose} \end{cases}$$

# Why not linear regression?

- This coding suggests a totally different relationship among the three conditions and would produce a different linear model.
- This will lead to different set of predictions on test observations.
- In general, it is not possible to convert a qualitative response variable with more than two levels into a quantitative response for linear regression.
- Suppose we consider binary (two level) qualitative response.
- For the example given above, suppose there are only two possibilities for the patient's medical condition: stroke and drug overdose

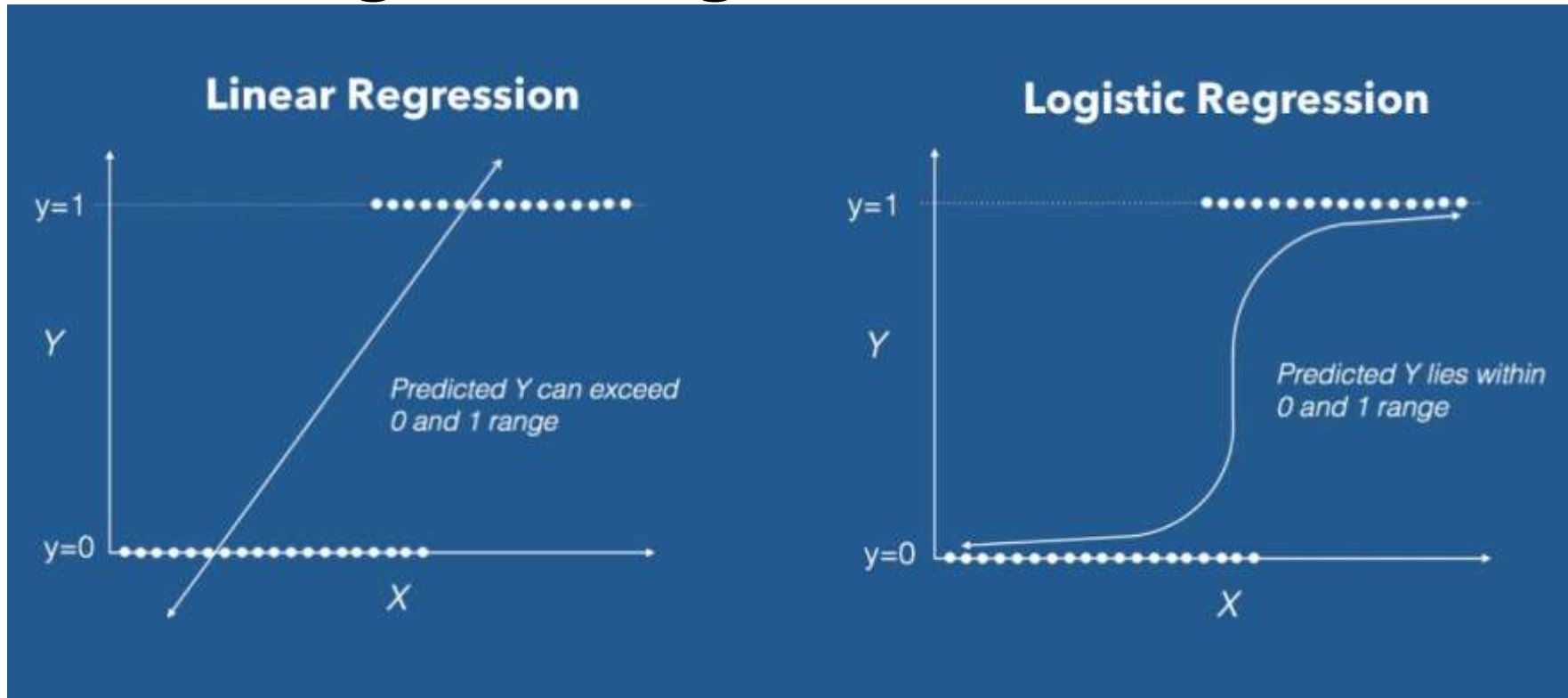
# Why not linear regression?

- In such situation, we can use the dummy variable approach to code the response as follows:

$$Y = \begin{cases} 0 & \text{if stroke} \\ 1 & \text{if drug overdose} \end{cases}$$

- For this binary response, we can fit a linear regression and predict drug overdose if  $\hat{Y} > 0.5$  and stroke otherwise.
- Even if we flip the above coding, the linear regression will produce the same results.

# Linear Vs Logistic Regression



Linear regression

$$Y = b_0 + b_1 \times X_1 + b_2 \times X_2 + \dots + b_K \times X_K$$

Sigmoid Function

$$P = \frac{1}{1 + e^{-Y}}$$

$$\ln \left( \frac{P}{1 - P} \right) = b_0 + b_1 \times X_1 + b_2 \times X_2 + \dots + b_K \times X_K$$

## Calculate Odds from Logit

$$\log \frac{p(x)}{1 - p(x)} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_n X_n + \varepsilon$$

Logit(p)

$$\frac{p(x)}{1 - p(x)} = \exp^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_n X_n}$$

odds of p.

# Why not linear regression?

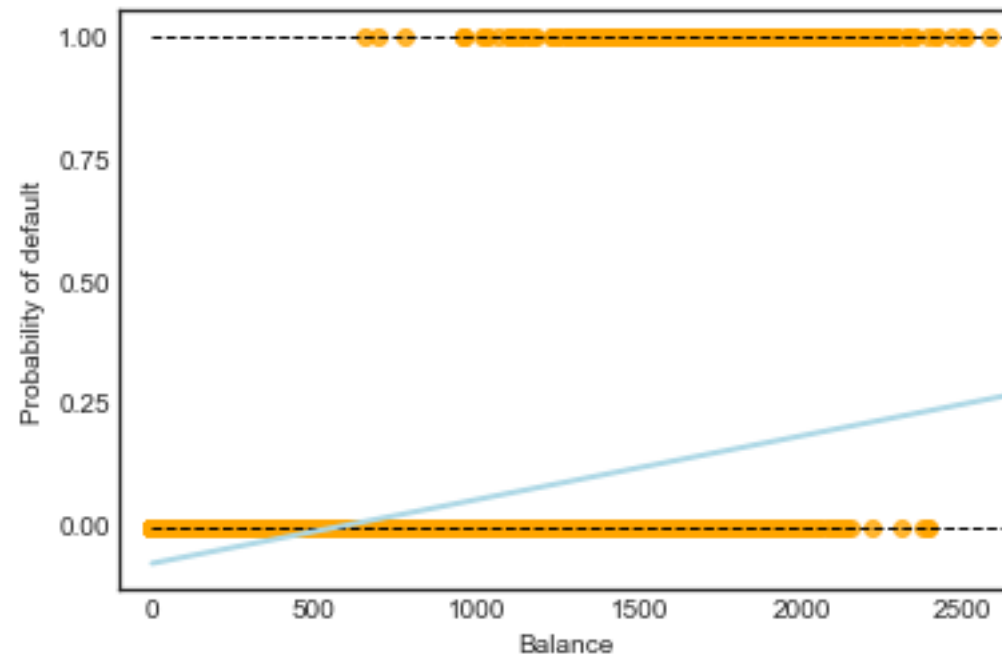
- Linear Regression may give

$$p(X) = \beta_0 + \beta_1 X$$

- The above equation can provide estimate of  $\Pr(\text{drug overdose}|X)$
- However, if we use the linear regression, some of the estimates might be outside the  $[0, 1]$  interval.
- For a predicted value that is close to zero we may predict a negative probability for default.
- If we predict a very large value, the probability could be bigger than 1.
- This is not sensible as probability should fall between 0 and 1.
- So, we should model  $p(X)$  using function that gives output between 0 and 1

# Why not linear regression?

- The figure 2 shows estimated probability of default when we use the linear regression.
- You may notice that some estimated probabilities are negative.
- The orange ticks represent the 0/1 value for default (No or Yes).



# Logistic Regression

- We can use logistic regression to model the probability that  $Y$  belongs to a particular category.
- For our case study, logistic regression models the probability of default.
- The probability of default given balance is
$$\Pr(\text{default} = \text{Yes} | \text{balance})$$
- The value of  $\Pr(\text{default} = \text{Yes} | \text{balance})$  can also be written as  $p(\text{balance})$ , will range between 0 and 1.
- Hence, we can predict  $\text{default} = \text{Yes}$  for any individual for whom  $p(\text{balance}) > 0.5$ .



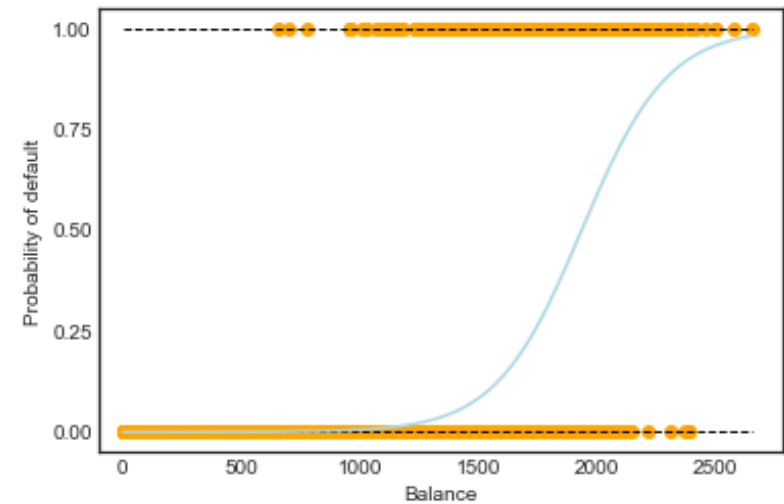
# Logistic Model

- In the previous section, we talked about how the linear regression may result in probabilities that may be lesser than 0 or bigger than 1.
- To avoid such problems, we need to model  $p(X)$  such that the outputs fall between 0 and 1 for all values of  $X$ .
- We can use the **logistic function**.

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

# Logistic Model

- The figure below illustrates the fit of the logistic regression model to the data used for credit card default case study.
- Here for low balance, we predict the probability of the default as close to zero but never below zero.
- Similarly for high balance, we predict the default probability close to, but never one.
- The logistic function will always produce S-Shaped curve.
- Hence, regardless of the value of  $X$ , we will always obtain a sensible prediction.
- We also notice that the logistic model can capture the range of probabilities more effectively than the linear regression model.



# Logistic Model

- We can also write the logistic equation as

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0 + \beta_1 X}$$

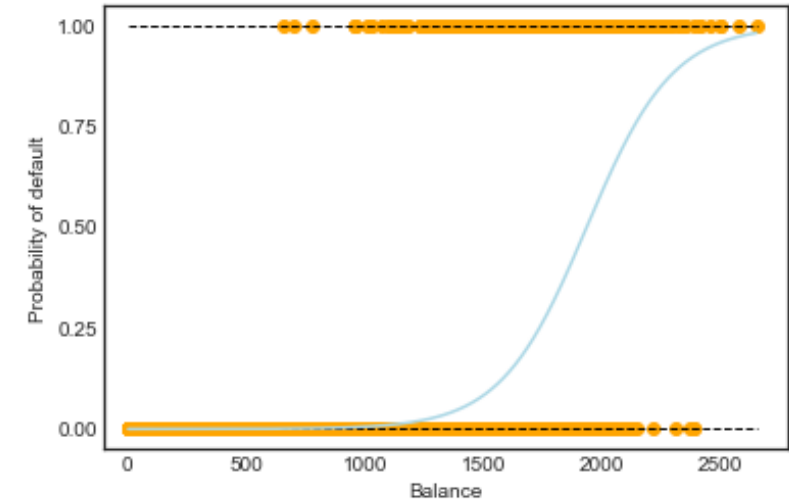
- We can call the quantity  $p(X)/[1 - p(X)]$  as **odds** which can take any value between 0 and  $\infty$ .
- Values that are close to 0 and  $\infty$  indicate very low and very high probabilities of default.
- For example, if 1 out of 5 people defaults,  $p(X) = 0.2$ , odds would be  $\frac{0.2}{1-0.2} = \frac{1}{4}$ .
- Similarly, on average nine out of every ten people with an odds of 9 will default because  $p(X) = 0.9$  means an odds of  $\frac{0.9}{1-0.9} = 9$ .

# Logistic Model

- We can take the logarithm of both sides to get

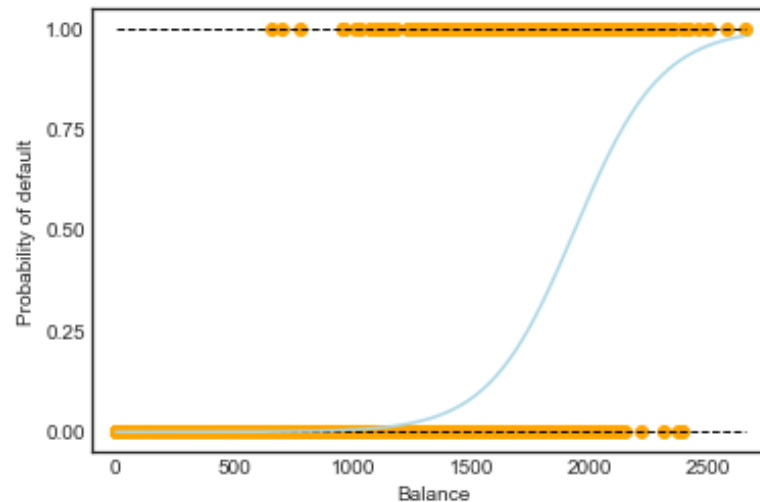
$$\log \left( \frac{p(X)}{1 - p(X)} \right) = \beta_0 + \beta_1 X$$


- The left-hand side is called the **log-odds** or **logit**. So we can see that logit is linear in  $X$ .
- In logistic regression model, the log odds change by  $\beta_1$ , if increase  $X$  by one unit.
- The relationship between  $p(X)$  and  $X$  is not a straight line, so  $\beta_1$  does not corresponds to change in  $p(X)$  with respect to one-unit increase in  $X$ .
- The amount that  $p(X)$  changes due to one-unit increase in  $X$  depends on the current value of  $X$



# Logistic Model

- However, we can say that if  $\beta_1$  is positive, an increase in  $X$  will be associated with increase in  $p(X)$ .
- Similarly, if  $\beta_1$  is negative, an increase in  $X$  will be associated with decrease in  $p(X)$ .



<u>Basis</u>	<u>Linear Regression</u>	<u>Logistic Regression</u>
Core Concept 	The data is modelled using a straight line	The probability of some obtained event is represented as a linear function of a combination of predictor variables.
Used with	Continuous Variable	Categorical Variable
Output/Prediction	Value of the variable	Probability of occurrence of event
Normality of Residual	Linear regression requires error terms should be normally distributed.	logistic regression does not require error terms should be normally distributed.
Linear Relationship	Linear regression needs a linear relationship between the dependent and independent variables.	logistic regression does not need a linear relationship between the dependent and independent variables.
Accuracy and Goodness of fit	measured by loss, R squared, Adjusted R squared etc.	Accuracy, Precision, Recall, F1 score, ROC curve, Confusion Matrix, etc

# Bayes Classifier

# Bayes Theorem

## Likelihood

Probability of collecting this data when our hypothesis is true

## Prior

The probability of the hypothesis being true before collecting data

$$P(H|D) = \frac{P(D|H) P(H)}{P(D)}$$

## Posterior

The probability of our hypothesis being true given the data collected

## Marginal

What is the probability of collecting this data under all possible hypotheses?



# Naive Bayes Classifiers

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is mainly used in text classification that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
- Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles

# Why is it called Naïve Bayes?

- The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:
- Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem.

# Types of Naive Bayes Model

- **Gaussian:** The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.
- **Multinomial:** The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems, it means a particular document belongs to which category such as Sports, Politics, education, etc.
- The classifier uses the frequency of words for the predictors.
- **Bernoulli:** The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

# Advantages of Naïve Bayes Classifier:

- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.
- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for text classification problems.

# Disadvantages of Naïve Bayes Classifier:

- Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

# KNN Classifier

- KNN -“Birds of a feather flock together.” similar things are near to each other.
- K-NN is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

**Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.





# N Nearest Neighbour

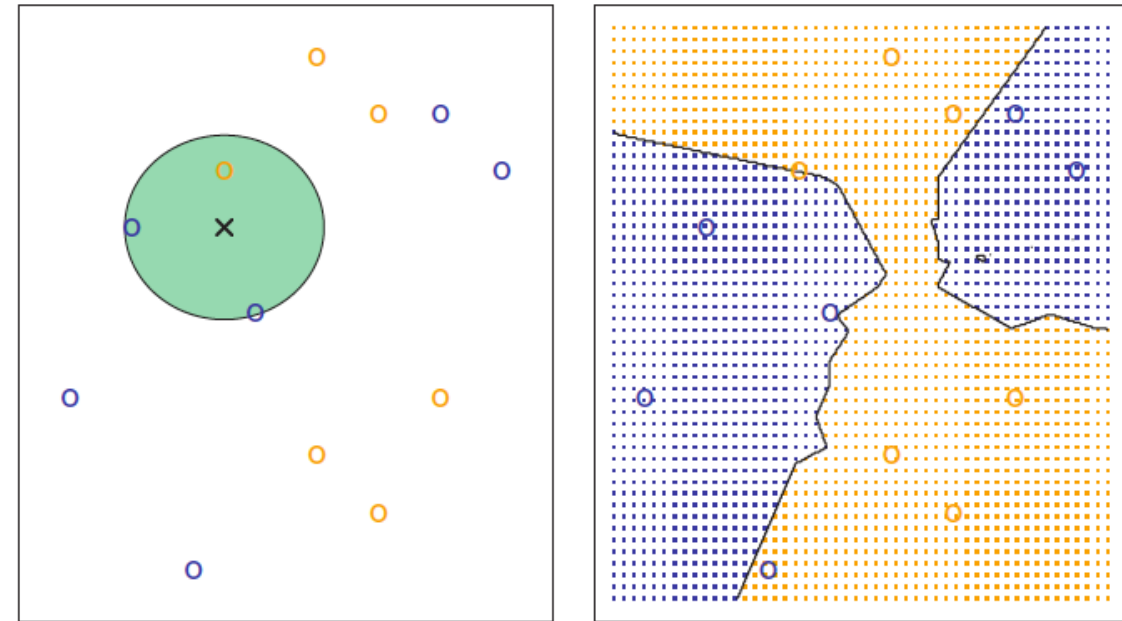
- K-nearest neighbors (KNN) classifier is one of the most popular classifiers.
- For a positive integer  $K$  and a test observation  $x_0$ , first of all, the KNN classifier identifies the  $K$  training observations that are nearest to  $x_0$ , represented by  $N_0$ .
- The second step is to estimate the conditional probability for class  $j$  as the fraction of points in  $N_0$ , whose response values equal  $j$ :

$$\Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$

- In the last step, KNN applies Bayes rule and classifies the test observation  $x_0$  to the class that has the highest probability.

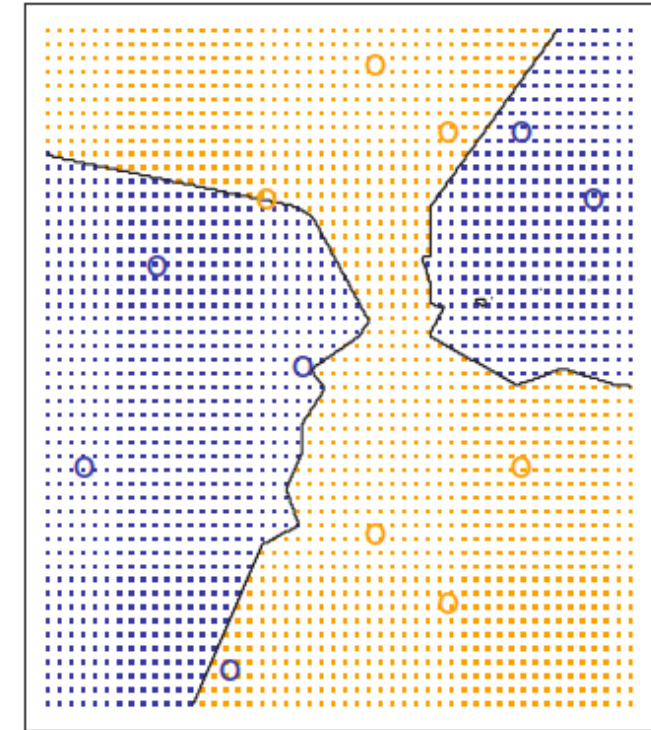
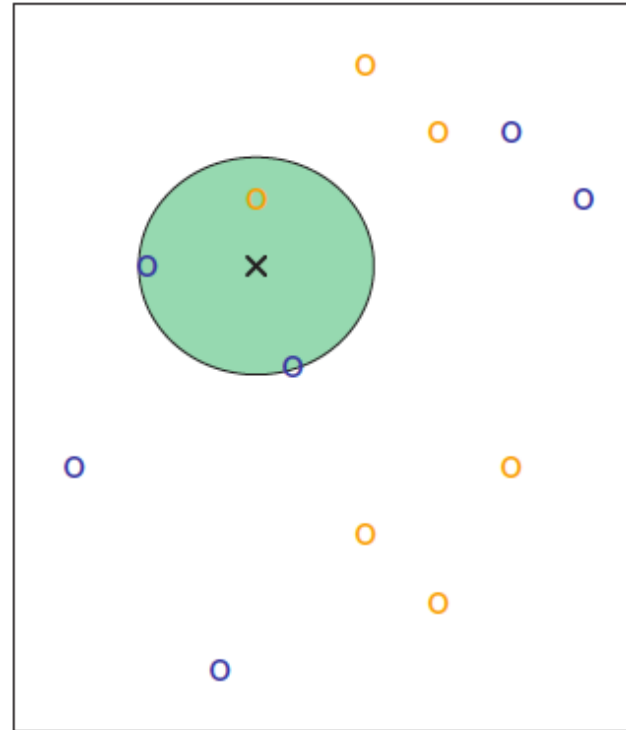
# N Nearest Neighbour

- An example of the KNN approach is given in figure 15.
- The left side panel demonstrates a small training data set that consists of six blue and six orange observations.
- The objective is to predict the label for the point marked by the black cross. Let's choose  $K = 3$ .
- The first step for KNN would be to identify the three observations that are nearest to the cross. We can show the neighborhood as a circle.
- It consists of two blue points and one orange point.



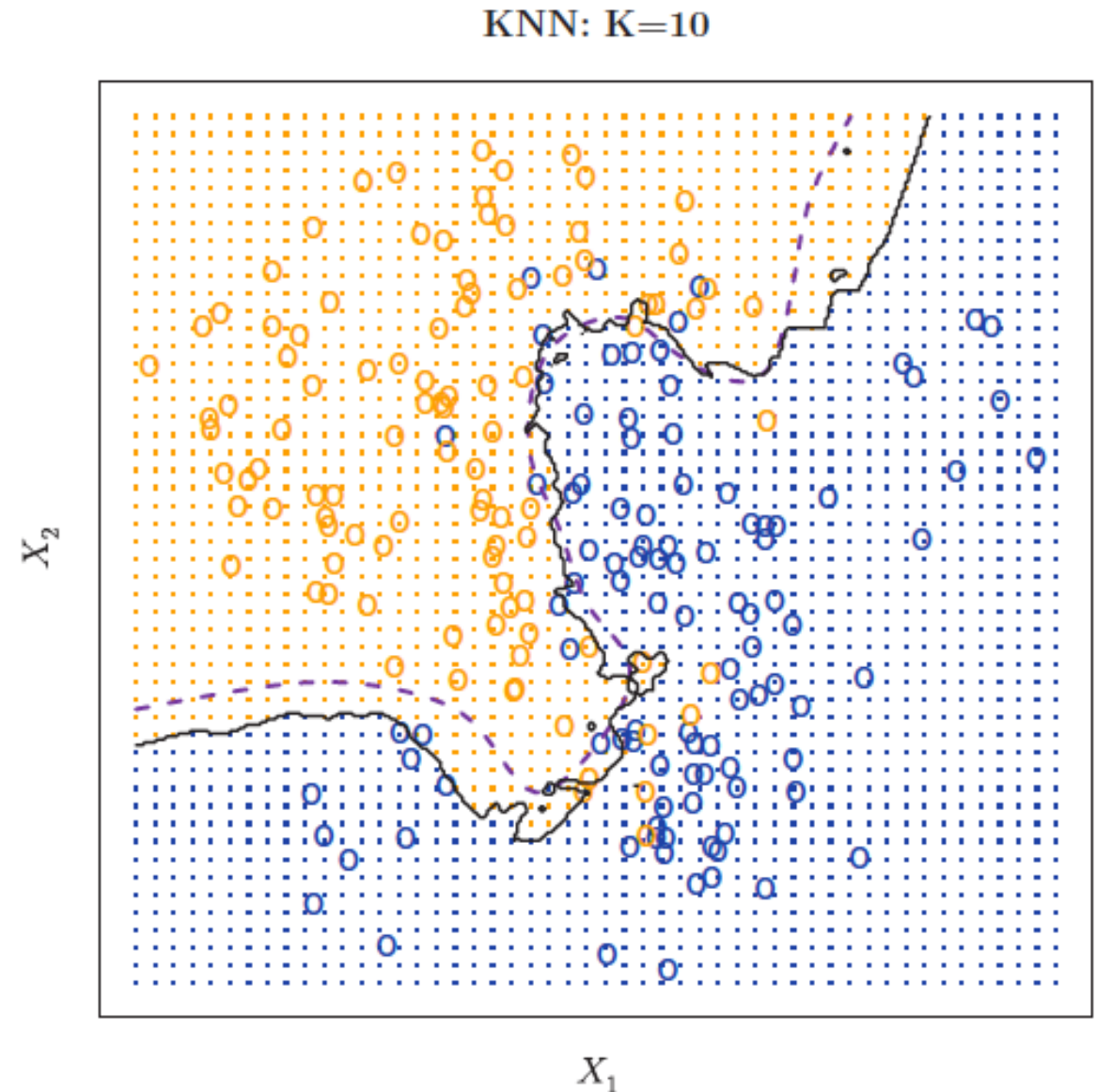
# N Nearest Neighbour

- That results in estimated probabilities of  $2/3$  for the blue class and  $1/3$  for the orange class.
- Thus, KNN prediction would be that the black cross belongs to the blue class.
- In the right-hand panel of Figure 15, KNN approach with  $K = 3$  has been applied to all of the possible values for  $X_1$  and  $X_2$ .
- Based on the classification approach as given above, the KNN decision boundary has been drawn.



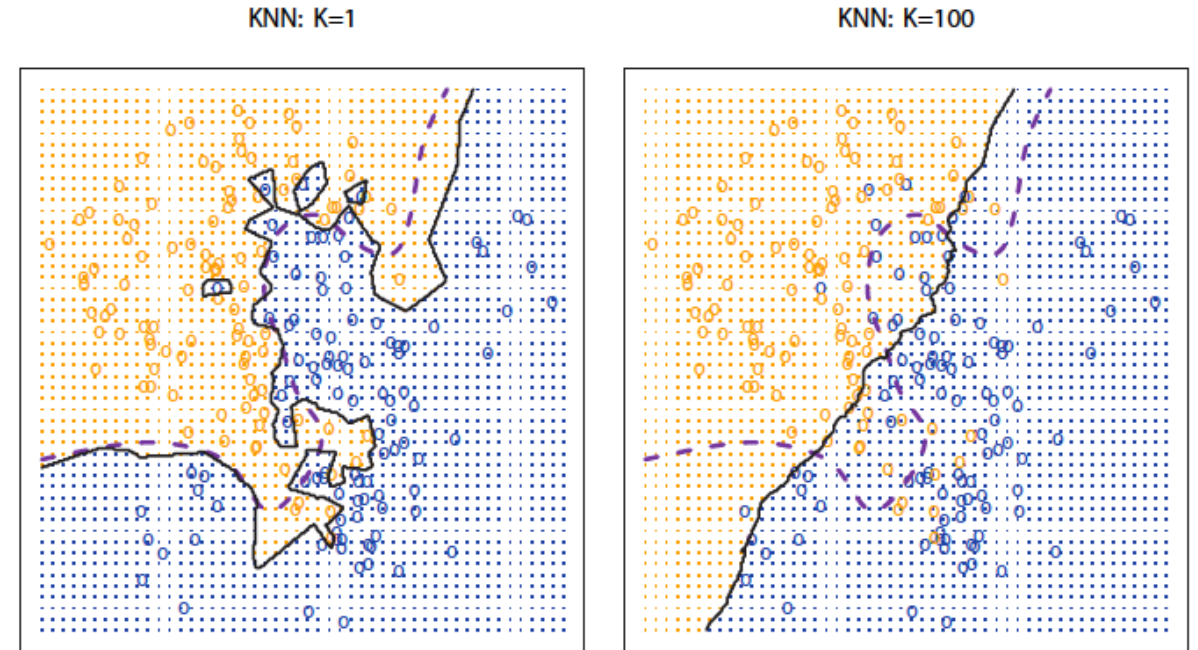
# N Nearest Neighbour

- Even though KNN is a very simple approach, it often provides the results that are close to the optimal Bayes classifier.
- In Figure 16, we have displayed the KNN decision boundary, using  $K = 10$ , that was applied to the larger simulated data set.
- We can see that even though the KNN classifies is not aware of the true distribution, the KNN decision boundary and Bayes classifier boundary is very close.
- In this case, the test error rate using KNN is 0.1363, where are the Bayes error rate is 0.1304 that are very close.



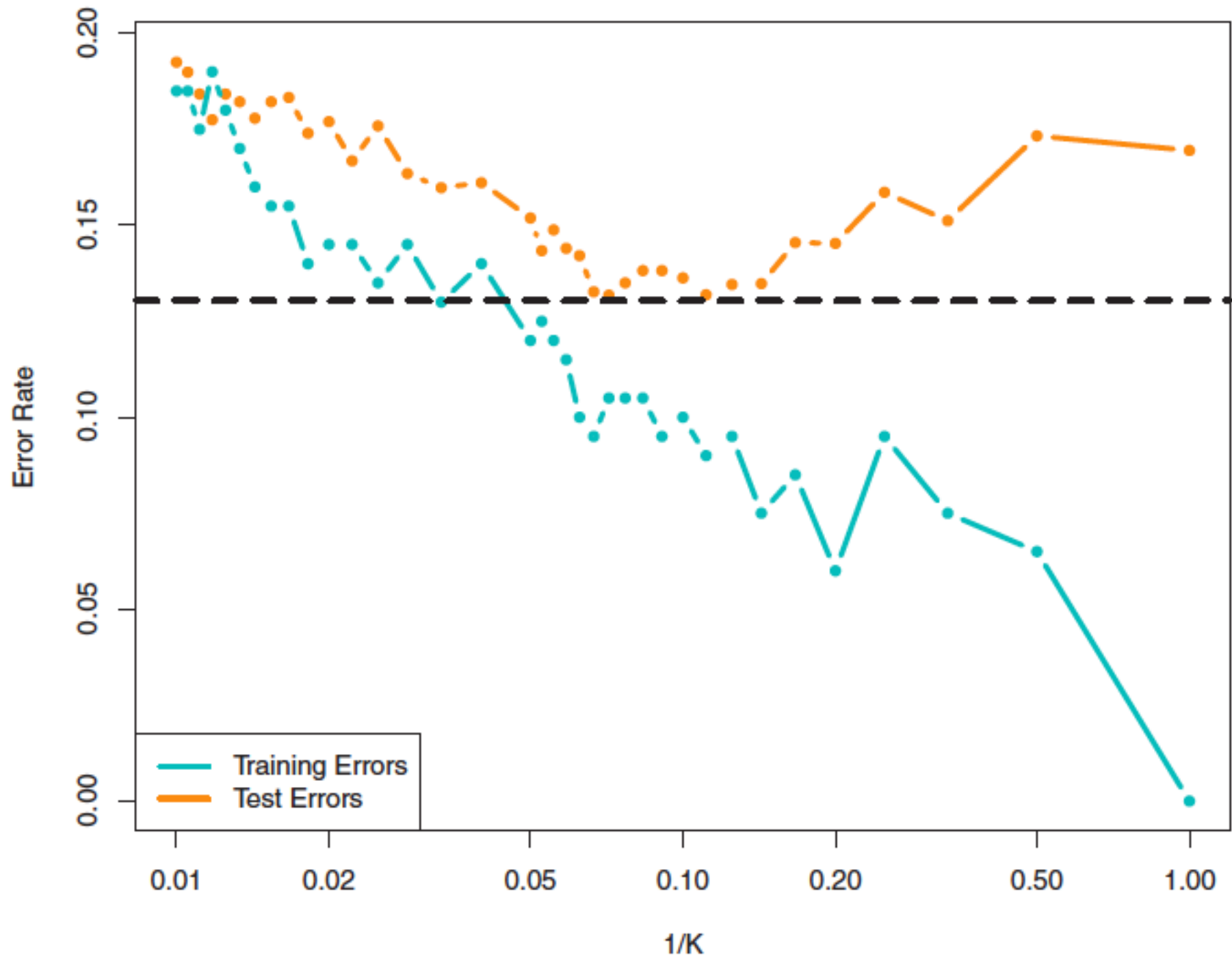
# N Nearest Neighbour

- The results from KNN classifier are drastically dependent on the choice of  $K$ . Figure 17 shows two KNN fits using  $K = 1$  and  $K = 100$  on the data shown in figure 14.
- With  $K = 1$ , the decision boundary in this case is very flexible and it finds patterns in the data that do not represent the Bayes decision boundary.
- Hence, we get a classifier, that has low bias but very high variance.
- As we increase the value of  $K$ , the method becomes less flexible and produces a decision boundary that is close to linear.
- Hence, we get a classifier, that has a low-variance but high-bias.
- On this data set, we neither get good predictions from  $K = 1$  nor  $K = 100$  good predictions.
- The test error rates are 0.1695 and 0.1925 respectively.



# N Nearest Neighbour

- Similar to the regression methods, we do not find a strong relationship between the training error rate and the test error rate.
- When we choose  $K = 1$ , we get KNN training error rate as 0, but we get high the test error rate. In general, by using more flexible classification models, we may achieve lower the training error rate but higher test error rate.
- Figure 18, shows the KNN test and training errors as a function of  $1/K$ . As we increase  $1/K$ , the method becomes more flexible.
- Similar to the regression methods, the training error rate consistently declines as the flexibility increases.
- But the test error demonstrates a characteristic U-shape.
- First it declines (with a minimum at approximately  $K = 10$ ) before it increases again when the method becomes excessively flexible and overfits.





# Advantages of KNN

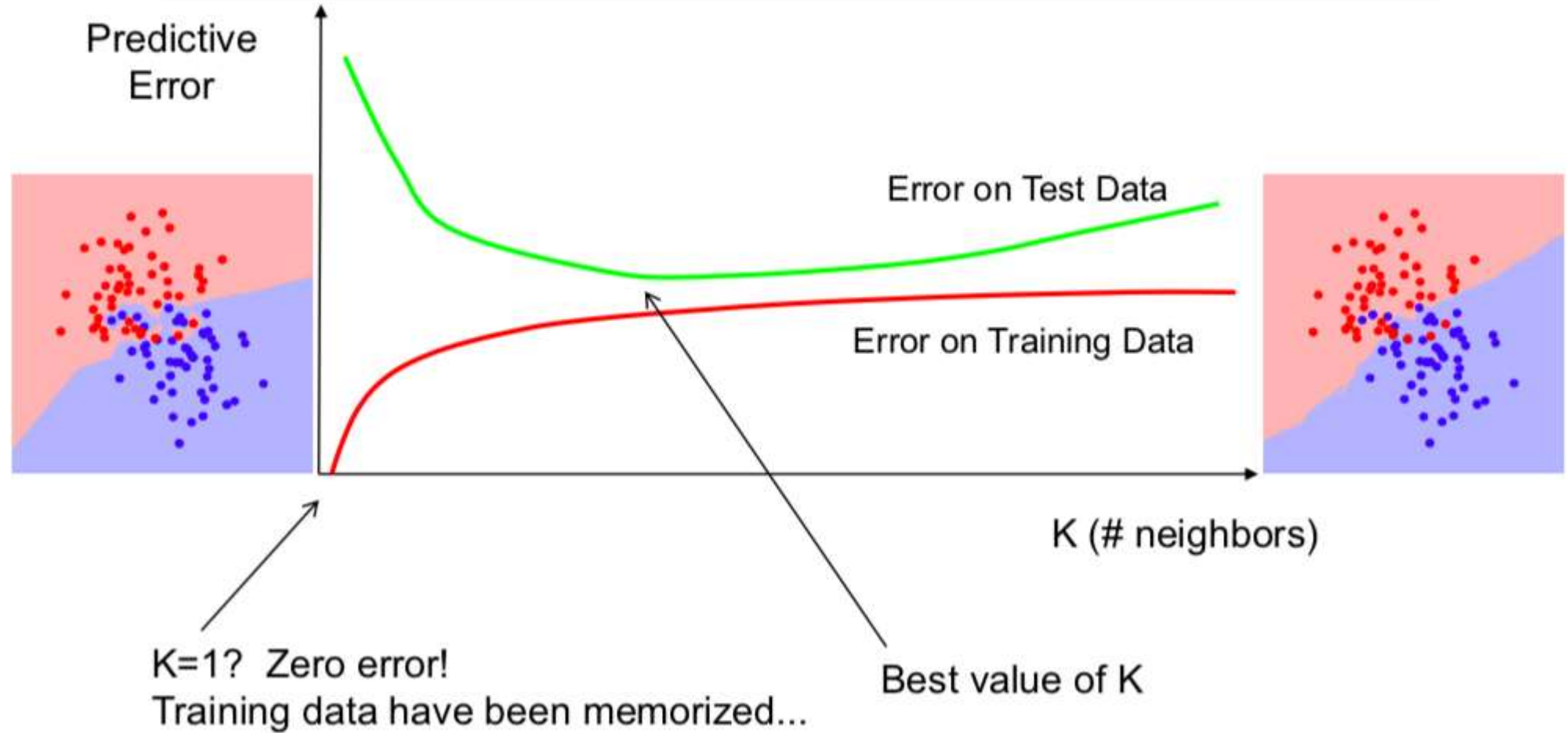
- 1. No Training Period:** KNN is called **Lazy Learner (Instance based learning)**. It does not learn anything in the training period. It does not derive any discriminative function from the training data. In other words, there is no training period for it. It stores the training dataset and learns from it only at the time of making real time predictions. This makes the KNN algorithm much faster than other algorithms that require training e.g. SVM, Linear Regression etc.
- 2.** Since the KNN algorithm requires no training before making predictions, **new data can be added seamlessly** which will not impact the accuracy of the algorithm.
- 3.** KNN is very **easy to implement**. There are only two parameters required to implement KNN i.e. the value of K and the distance function (e.g. Euclidean or Manhattan etc.)



# Disadvantages of KNN

- 1. Does not work well with large dataset:** In large datasets, the cost of calculating the distance between the new point and each existing points is huge which degrades the performance of the algorithm.
- 2. Does not work well with high dimensions:** The KNN algorithm doesn't work well with high dimensional data because with large number of dimensions, it becomes difficult for the algorithm to calculate the distance in each dimension.
- 3. Need feature scaling:** We need to do feature scaling (standardization and normalization) before applying KNN algorithm to any dataset. If we don't do so, KNN may generate wrong predictions.
- 4. Sensitive to noisy data, missing values and outliers:** KNN is sensitive to noise in the dataset. We need to manually impute missing values and remove outliers.

# Error rates and K



# KNeighborsClassifier from sklearn

- **n\_neighbors**: the value of k, the number of neighbors considered
- **weights**: if you want to use weighted attributes, here you can configure the weights. This takes values like uniform, distance (inverse distance to the new point) or callable which should be defined by the user. The default value is uniform.
- **algorithm**: if you want a different representation of the data, here you can use values like ball\_tree, kd\_tree or brute, default is auto which tries to automatically select the best representation for the current data set.
- **metric**: the distance metric (Euclidean, Manhattan, etc), default is Euclidean.

# Distance Functions

- Minkowski distance is a generalized metric to measure the distance between two points  $x$  and  $y$  in  $n$  dimensional space

$$d(x, y) = \sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p}$$

- *Manhattan distance* (city-block or L1 distance) is a special case of Minkowski, where we set  $p=1$ :

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

- *Euclidean Distance* is another special case of the Minkowski distance, where  $p=2$ :

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

# KNN is a curse for higher dimensions

- **It becomes computationally more expensive to compute distance and find the nearest neighbors in high-dimensional space**
- **Our assumption of similar points being situated closely breaks**
- As a result, the initial assumption of similar observations being close fails. We can't say similar observations are clustered closer together in a high-dimensional dataset as the distance between all data points becomes smaller and smaller.

# Maximum likelihood estimation

**Example:**  $X_1, \dots, X_n$  - i.i.d. random variables with probability  $p_X(x|\theta) = P(X=x)$  where  $\theta$  is a parameter

- likelihood function  $L(\theta|x)$  where  $x=(x_1, \dots, x_n)$  is set of observations

$$L(\theta|x) = \prod_{i=1}^n p_X(x_i|\theta)$$

- maximum likelihood estimate  $\hat{\theta}(x)$   
maximizer of  $L(\theta|x)$

# Estimating the Regression Coefficients

- For logistic regression, the coefficients  $\beta_0$  and  $\beta_1$  are unknown.
- We need to estimate them based on the available training data.
- For non-linear models, we prefer the **maximum likelihood** approach.
- In the approach, we try to estimate  $\beta_0$  and  $\beta_1$  such that the predicted probability  $\hat{p}(x_i)$  of default of each individual is as close as possible to the individual's observed default status.
- In other words, we can say that we try to find  $\hat{\beta}_0$  and  $\hat{\beta}_1$  such that if we plug the estimates into the model for  $p(X)$ , we get a number that is close to 1 for all those individuals who defaulted and a number that is close to zero for all the individuals who did not.
- We can easily fit the logistic regression using statistical software packages such as R and Python, so in this book, we need not focus on the details of maximum likelihood fitting procedure.

# Case Study - Results

- The table 1 exhibits the coefficient estimates and related information that we got after fitting the logistics regression model on the Default data to predict the probability of *default* = *Yes* using balance.
- From the results we can see that  $\hat{\beta}_1 = 0.0055$ . This indicates that the increase in balance and increase in the probability of default is associated.
- A one-unit increase in balance results in an increase in the log odds of default by 0.0055 units.

	Coef.	Std.Err.	z	P> z
const	-10.651331	0.361169	-29.491287	3.723665e-191
balance	0.005499	0.000220	24.952404	2.010855e-137

Table - 1



# Case Study - Results

- Table 1 also exhibits the similarity between linear regression output and logistic regression output.
- The z-statistics in the Table 1 plays the same role as the t-statistics in linear regression output.
- The z-statistics associated with  $\beta_1$  in this case is equal to  $\hat{\beta}_1 / SE(\hat{\beta}_1)$ .
- A large value of z-statistics is an indication of evidence against the null hypothesis  $H_0: \beta_1 = 0$ .
- Moreover, due to a very small p-value, we can reject  $H_0$  and we can conclude that there is an association between balance and probability of default.
- At this point, we are not interested in the estimated intercept value as shown in Table – 1.

# Making Predictions

- Once we have estimated the coefficients, we can easily compute the probability of default for any given credit card balance.
- For example, we can predict the default probability for an individual with balance \$1000 as

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X}} = \frac{e^{-10.6513 + 0.0055 \times 1000}}{1 + e^{-10.6513 + 0.0055 \times 1000}} = 0.00576$$

- The probability is less than 1%. Similarly, we can calculate the probability of default for an individual with a balance \$2000

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X}} = \frac{e^{-10.6513 + 0.0055 \times 2000}}{1 + e^{-10.6513 + 0.0055 \times 2000}} = 0.586$$

- The probability in this case is 58.6%

# Dummy Variable Approach

- We can also use the qualitative predictors with logistic regression model by using dummy variable approach.
- For our case study, we created a dummy variable that can take a value of 1 for students and 0 for non-students.
- The Table-2 exhibits the logistic regression model for predicting probability of default from student status.

	Coef.	Std.Err.	z	P> z
const	-3.504128	0.070713	-49.554094	0.000000
student2	0.404887	0.115019	3.520177	0.000431

Table – 2

# Dummy Variable Approach

- The coefficient associated with dummy variable is positive and associated p-value is statistically significant that indicates that the students tend to have higher default probabilities than non-students

$$\widehat{Pr}(\text{default} = \text{Yes} | \text{student} = \text{Yes}) = \frac{e^{-3.5041 + 0.4049 \times 1}}{1 + e^{-3.5041 + 0.4049 \times 1}} = 0.0431$$

$$\widehat{Pr}(\text{default} = \text{Yes} | \text{student} = \text{No}) = \frac{e^{-3.5041 + 0.4049 \times 0}}{1 + e^{-3.5041 + 0.4049 \times 0}} = 0.0292$$

# Multiple Logistic Regression

- We can predict a binary response using multiple predictors. Similar to multiple linear regression, we can generalize the multiple logistic regression as follows:

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X + \cdots + \beta_p X_p$$

- The  $X = (X_1, \dots, X_p)$  are  $p$  predictors. We can rewrite the above equation as

$$p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}$$

- Just in previous section, we can use maximum likelihood method to estimate  $\beta_0, \beta_1, \dots, \beta_p$ .

# Multiple Logistic Regression

- Table 3 exhibits the coefficient estimate for a logistic regression model using balance, income (in thousands of dollars), and student status to predict the probability of default.

	Coef.	Std.Err.	z	P> z
const	-10.869045	0.492273	-22.079320	4.995499e-108
balance	0.005737	0.000232	24.736506	4.331521e-135
income	0.000003	0.000008	0.369808	7.115254e-01
student2	-0.646776	0.236257	-2.737595	6.189022e-03

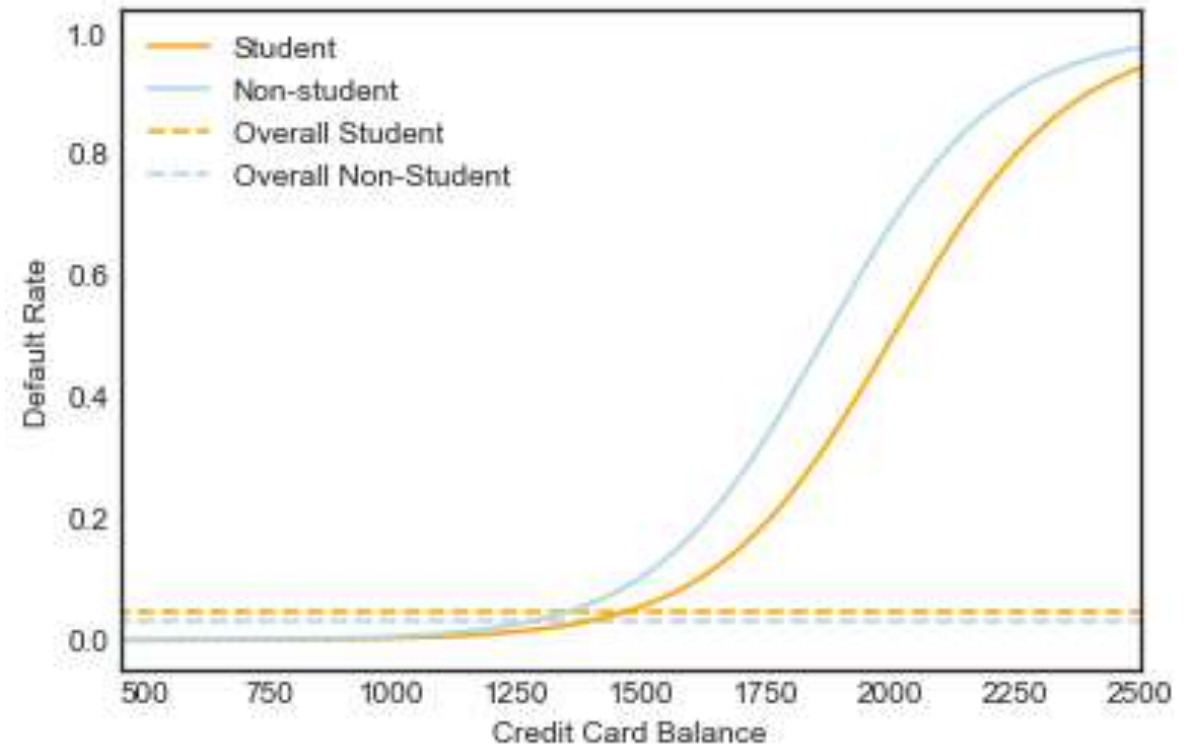
Table – 3

# Multiple Logistic Regression

- The results are surprising. The p-value associated with balance and the dummy variable for student status is very small.
- This indicates that these two variables are closely associated with the probability of default.
- But interestingly, the coefficient of dummy variable is negative.
- This indicates that the students are less likely to default than non-students.
- This contradicts with results we exhibited in Table - 2. We did further analysis to analyze the paradox.
- The results have been displayed in Figure – 4

# Multiple Logistic Regression

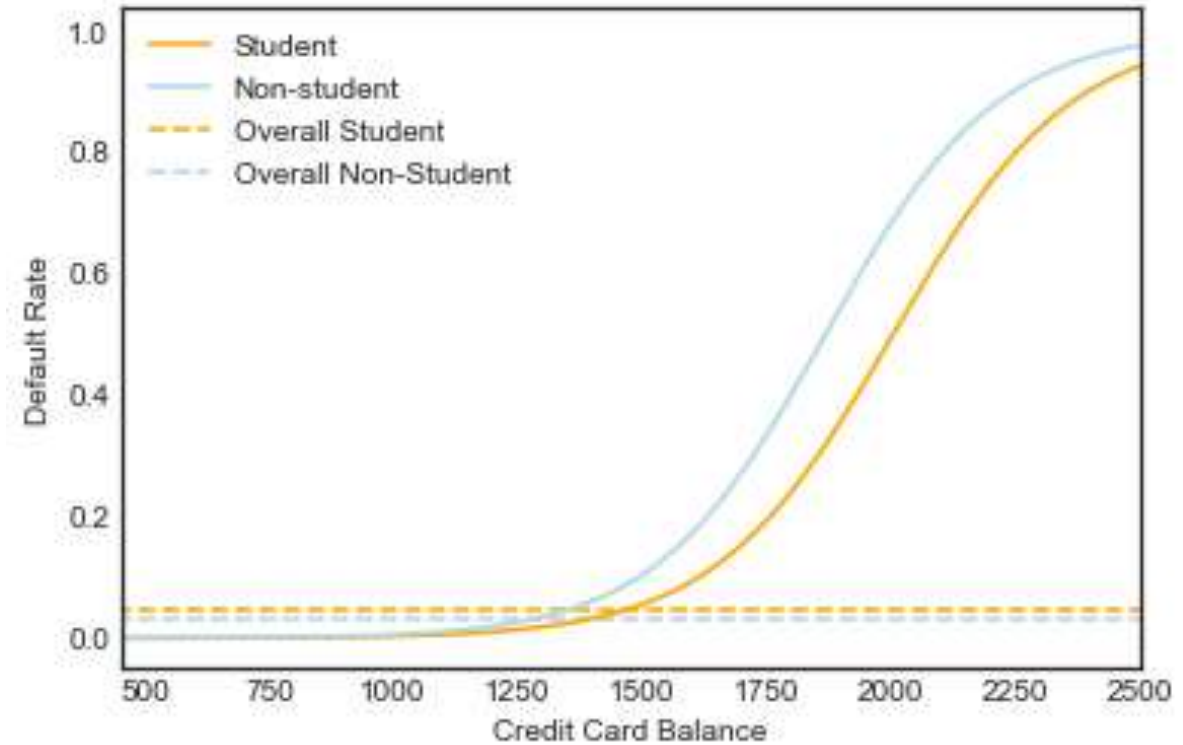
- The Figure 4 provides insights into the paradox.
- The default rate of students and non-students has been shown using the orange and blue lines respectively as a function of credit card balance.
- For a fixed value of balance and income, the students are less likely to default than a non-student.
- For every value of balance, the student default rate is at or below that of non-student default rate.





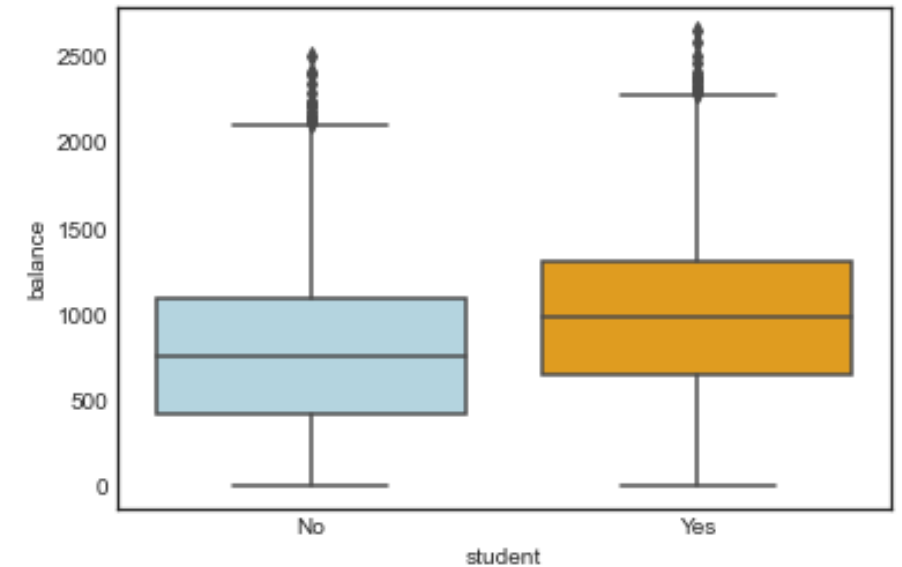
# Multiple Logistic Regression

- But the horizontal broken lines near the base of the plot, show that the default rate for students and non-students that has been averaged over all values of balance and income, has opposite effect.
- The overall student default rate is higher than non-student default rate. That has been shown by the positive coefficient for students in Table – 2.



# Multiple Logistic Regression

- The Figure – 5 provides an explanation for this discrepancy. We can see that the variables students and balance are correlated.
- The students tend to hold higher credit card balance which is associated with higher probability of default
- Even though an individual student with a given credit card balance will have a lower probability of default than a non-student with the same credit card balance, the student tends to default at a higher rate than the non-student.
- This is because overall students tend to default at a higher rate than non-students.



# Multiple Logistic Regression

- This insight may be important for a credit card company that tried to determine whom they should offer the credit.
- Overall a student is riskier than the non-student if the information about credit card balance is not available.
- But a student is less risky than a non-student with the same credit card balance.
- This case study highlights the dangers associated with applying regression methods to single predictors when another predictor may also be relevant.
- We have seen the same behavior in the simple linear regression as well. In general, we call this phenomenon as **confounding**.

# Multiple Logistic Regression

- From Table – 3 we can substitute the estimates for the regression coefficients into the equation for multiple logistics regression.
- For a student with a credit card balance of \$1,500 and an income of \$40,000, the estimated probability of default is

$$\hat{p}(X) = \frac{e^{-10.869+0.00574 \times 1500+0.000003 \times 40000-0.6468 \times 1}}{1 + e^{-10.869+0.00574 \times 1500+0.000003 \times 40000-0.6468 \times 1}} = 0.058$$

- A non-student with the same balance and income has an estimated probability of default of

$$\hat{p}(X) = \frac{e^{-10.869+0.00574 \times 1500+0.000003 \times 40000-0.6468 \times 0}}{1 + e^{-10.869+0.00574 \times 1500+0.000003 \times 40000-0.6468 \times 0}} = 0.105$$

# Logistic Regression > 2 Response Classes

- Sometimes we need to classify a response variable with more than two classes.
- For example, in the previous section, we discussed about the three possible categories of medical conditions in the emergency ward: stroke, drug overdose, epileptic seizure.
- In this case, we would like to model  $\Pr(Y = \textit{stroke}|X)$  and  $\Pr(Y = \textit{drug overdose}|X)$  and the remaining

$$\Pr(Y = \textit{epileptic seizure} |X) = 1 - \Pr(Y = \textit{s troke}|X) - \Pr(Y = \textit{drug overdose}|X)$$

- We can also use logistic regression for modeling this problem also but in practice they are not very popular.
- For multiple class classification, Linear Discriminant Analysis is more popular.

# Why LDA?

- Logistic Regression is one of the most popular linear classification models that perform well for binary classification but falls short in the case of multiple classification problems with well-separated classes. While LDA handles these quite efficiently.
- LDA can also be used in data preprocessing to reduce the number of features just as PCA which reduces the computing cost significantly.
- LDA is also used in face detection algorithms. In Fisherfaces LDA is used to extract useful data from different faces. Coupled with eigenfaces it produces effective results.

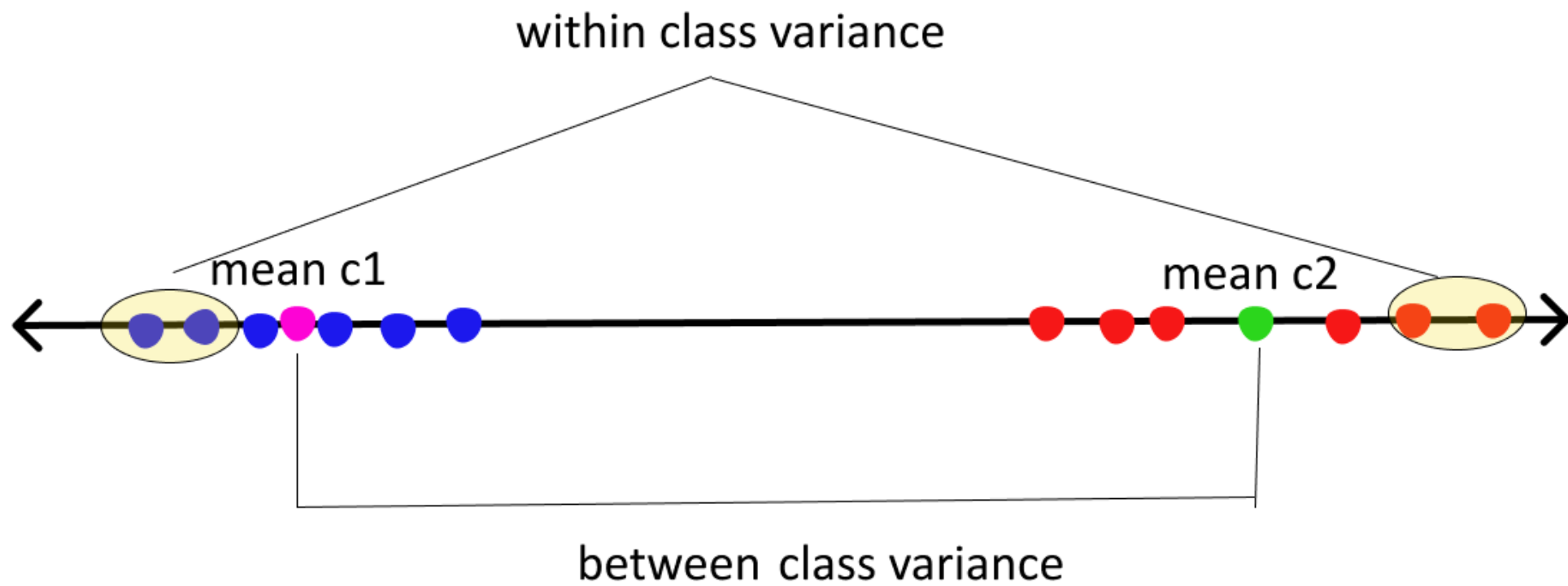
# Shortcomings

- Linear decision boundaries may not effectively separate non-linearly separable classes. More flexible boundaries are desired.
- In cases where the number of observations exceeds the number of features, LDA might not perform as desired. This is called *Small Sample Size* (SSS) problem. Regularization is required.

# Assumptions

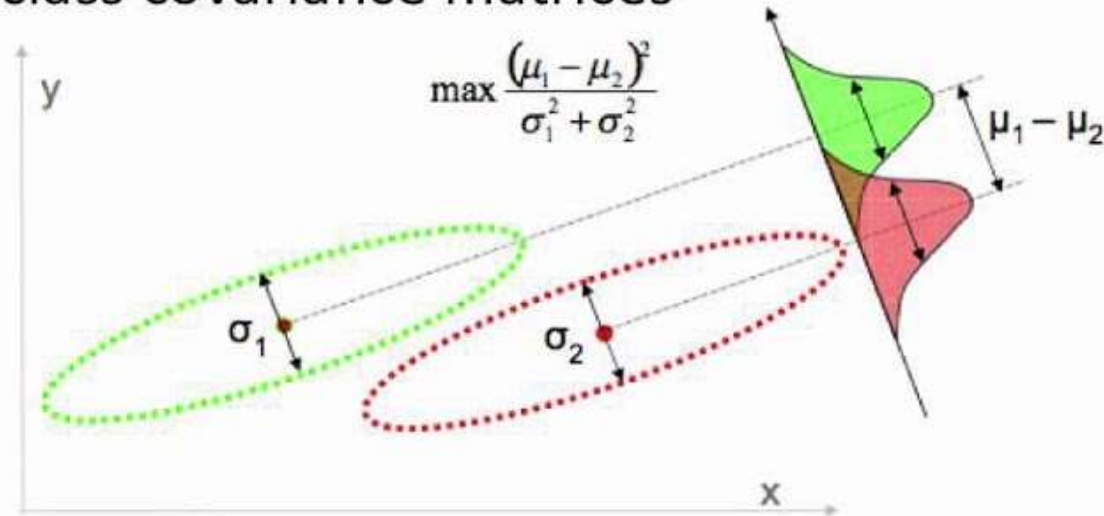
- Assumes the data to be distributed normally or Gaussian distribution of data points i.e. each feature must make a bell-shaped curve when plotted.
- Each of the classes has identical covariance matrices.





# Linear Discriminant Analysis

- LDA: pick a new dimension that gives:
  - maximum separation between means of projected classes
  - minimum variance within each projected class
- Solution: eigenvectors based on between-class and within-class covariance matrices



Copyright © 2013 Victor Lavrenko

# Linear Discriminant Analysis

- In the case of linear discriminant analysis, we model the distribution of the predictors  $X$  separately in each of the response classes (i.e given  $Y$ ).
- We use Bayes' theorem to flip these to find the estimates for  $\Pr(Y = k|X = x)$ .
- When we assume the distributions to be normal, the models are very similar to logistic regression.

# Comparison between Logistic Regression / LDA

- In the situations, where the classes are well-separated, the parameter estimates for the logistic model are unstable whereas in the case of linear discriminant analysis, we do not face such problems
- For a small sample size ( $n$  is small), the distribution of the predictors  $X$  is approximately normal in each of the classes.
- In such cases linear discriminant model is more stable than the logistic regression model.
- Linear discriminant analysis is more popular than logistic regression model when we have two response classes.

# LDA on credit card default data

- We performed LDA on the data for credit card default case study.
- The objective was to predict whether an individual will default on the basis of credit card balance and student status.
- The confusion matrix is used to compare the LDA prediction to the true default status for the 10000 training observations in the data set.
- The elements on the diagonal represents the individuals whose default statuses were correctly predicted.
- The elements on the off-diagonal represents the individuals that were misclassified.

# LDA on credit card default data

True Default/ Predicted Default	No	Yes	Total
No	9645	254	9899
Yes	22	79	101
Total	9667	333	10000

- From Table- 4, we can infer that that LDA predicted that 101 people will default.
- Out of 101 people, only 79 actually defaulted whereas 22 did not defaulted.
- Hence only 22 out of 9667 individuals who did not defaulted were incorrectly labelled.
- This shows a low error rate. However, out of 333 individuals who defaulted, 254 (or 76.3%) were missed by LDA model.
- Hence even though the error rate is low, the error rate among the individuals who defaulted is very high.
- A credit card company always try to identify the high-risk individuals.
- An error rate of  $254/333 = 76.3\%$  for individuals who defaulted is very high and it would be unacceptable to the company.

# LDA on credit card default data

- We can characterize the performance of a classifier using the terms sensitivity and specificity.
- The **sensitivity** is the percentage of true defaulters that were identified. That is  $79/333 = 23.7\%$ .
- The specificity is the percentage of non-defaulters that we correctly identified. That is  $9645/9667 = 99.8\%$
- Why do we get such poor results while classifying the customers, who defaulted or why it has such a low sensitivity?
- We have seen that LDA approximate the Bayes classifier which has the lowest error rate out of all classifiers.
- Still, it will incorrectly assign a customer who does not default to the default class and assign a customer who defaults to non-default class.

# LDA on credit card default data

- For credit card company the customers, who were wrongly classified as non-defaults, are more problematic than the customers, who were wrongly classified as defaulters.
- Hence, we need a classifier that can meet the credit card company's need.
- The Bayes' classifier assigns an observation to the default class if

$$\Pr(\text{default} = \text{Yes} | X = x) > 0.5$$

- Hence, the Bayes' classifier uses a threshold of 50% to assign an observation to the default class.



# LDA on credit card default data

True Default/ Predicted Default	No	Yes	Total
No	9435	140	9575
Yes	232	193	425
Total	9667	333	10000

- If we want to fix the issue of incorrectly predicting the default status for individuals who default, we should lower the threshold.
- It means, we should assign the customer with posterior probability of default above 20% to default class

$$\Pr(\text{default} = \text{Yes} | X = x) > 0.2$$

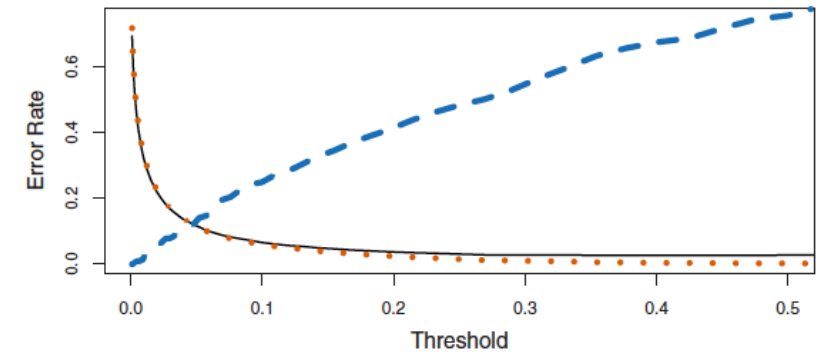
- Table – 5 exhibits the results from this approach.

# LDA on credit card

True Default/ Predicted Default	No	Yes	Total
No	9435	140	9575
Yes	232	193	425
Total	9667	333	10000

- LDA predicts ( $232+193 = 425$ ) individuals will default.
- Of 333 individuals who default, LDA could not predict 140 individuals correctly that is 42.04%.
- Hence, we get a vast improvement over the error rate of 76.3% that we got by using the threshold of 50%.
- But in the meantime: the number of individuals who do not default but incorrectly classified, increased to 232 from 22.
- The overall error rate has increased slightly.
- But the credit card company may be ready to accept the slightly increase in total error rate if a greater number of individuals, who actually default, can be identified.

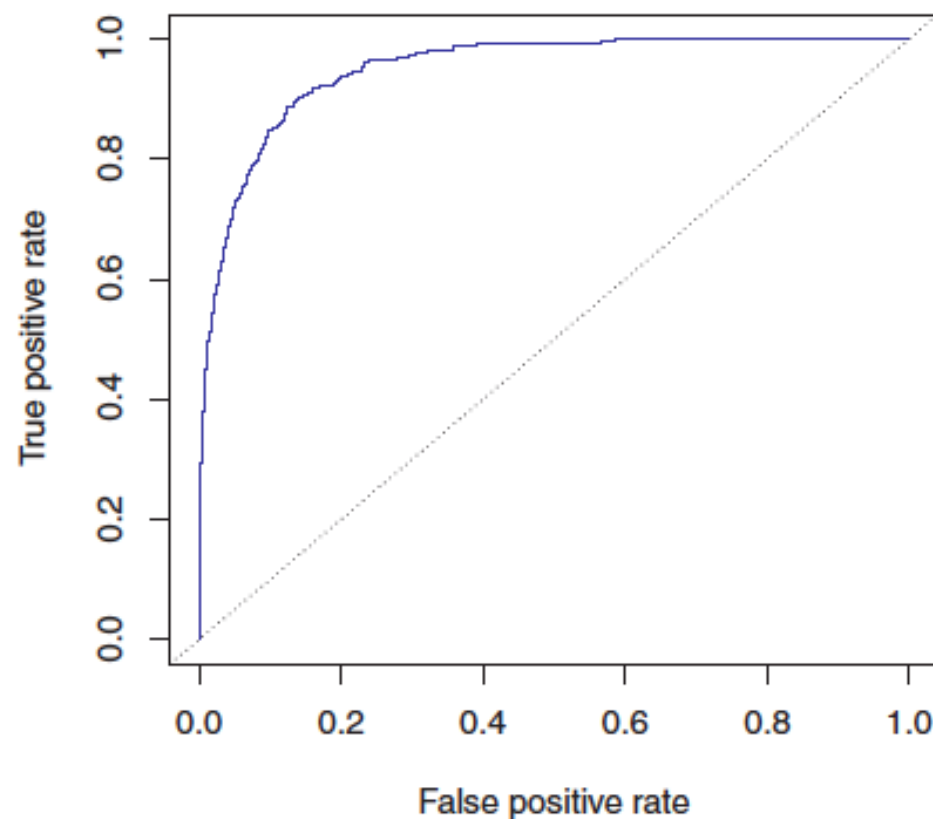
# LDA on credit card default data



- We have demonstrated this tradeoff that we got by modifying the threshold value of the posterior probability of default in Figure – 8
- In the Figure – 8, the error rates have been shown as a function of the threshold value.
- If we use the threshold value of 0.8, the overall error rate as shown in black solid line minimizes.
- But for the threshold value of 0.5, the error rate among the individuals who default is quite high (blue dashed line).
- As we reduce the threshold, the error rate among the individuals who default decreases, but the error rate among the individuals who do not default increases.
- We can define this threshold value based on domain knowledge and business objectives.

# ROC Curve

- We use the ROC curve to display the two types of errors for all possible thresholds simultaneously.
- ROC is an acronym for receiver operating characteristics.
- In the Figure – 9, we have demonstrated the ROC curve for the LDA classifier on the training data.
- The overall performance of a classifier, summarized over all the possible thresholds, is given by the area under the (ROC) curve (AUC).
- An ideal ROC curve will touch the top left corner.
- Hence the larger AUC, the better classifier.
- For this data, we got AUC as 0.95



# ROC Curve

- In the ROC curve in Figure – 9, the true positive rate is the sensitivity: the fraction of defaulters who were correctly identified for a threshold value.
- The false positive rate is specificity: the fraction of non-defaulters whom we incorrectly classified as defaulters.
- The different ROC curve for 50% threshold have been given in Figure 10 – Figure 13.

# LDA Vs QDA

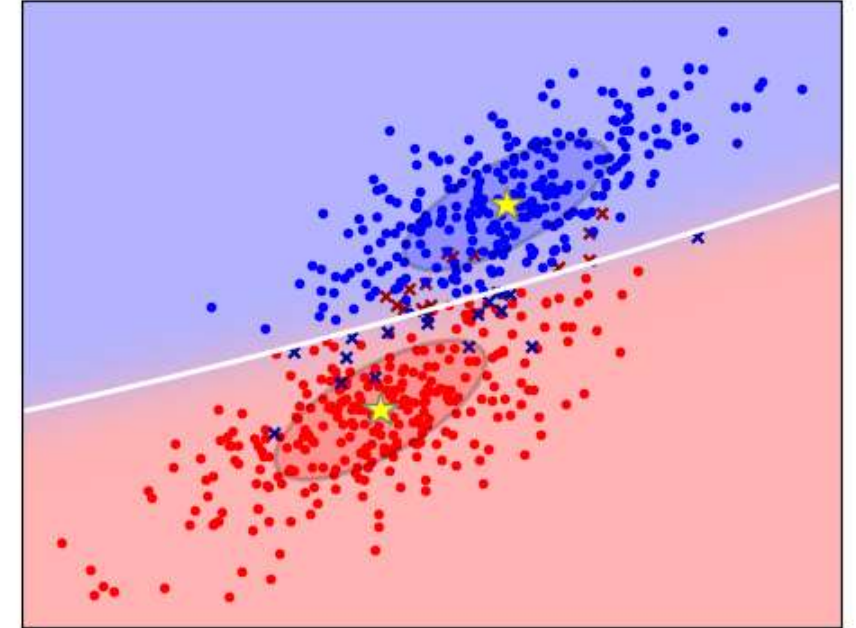
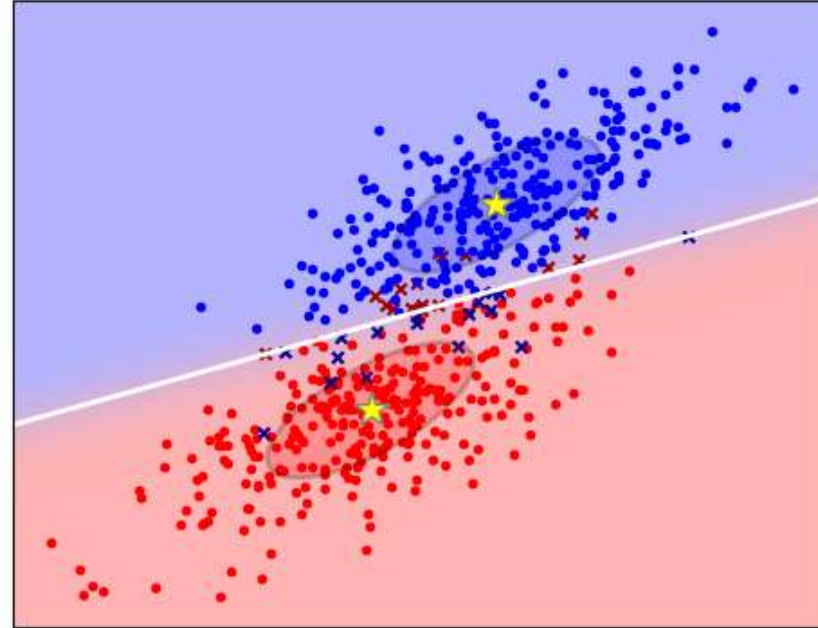
- A major difference between the two is that LDA assumes the feature covariance matrices of both classes are the same, which results in a linear decision boundary.
- In contrast, QDA is less strict and allows different feature covariance matrices for different classes, which leads to a quadratic decision boundary.
- QDA, because **it allows for more flexibility for the covariance matrix**, tends to fit the data better than LDA, but then it has more parameters to estimate. The number of parameters increases significantly with QDA. Because, with QDA, you will have a separate covariance matrix for every class.

# Linear Discriminant Analysis vs Quadratic Discriminant Analysis

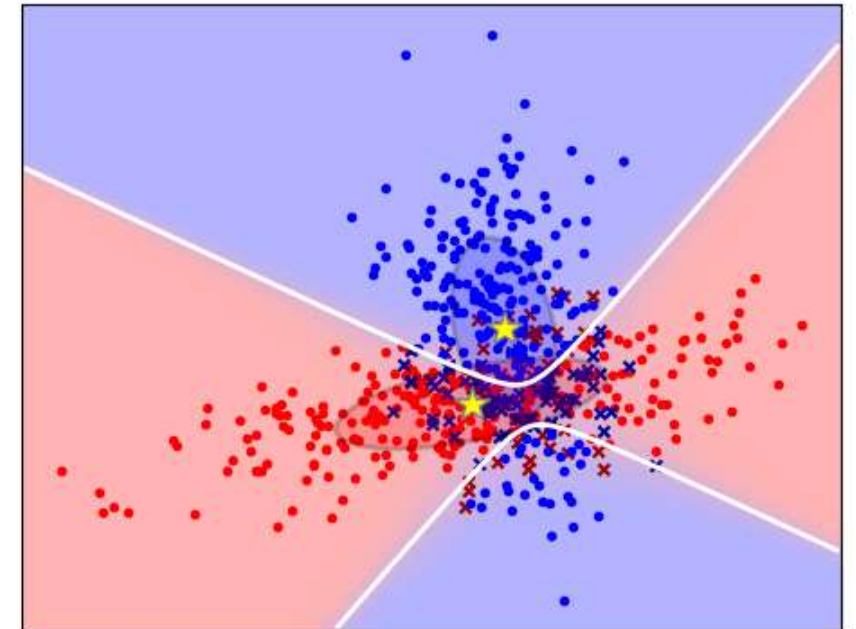
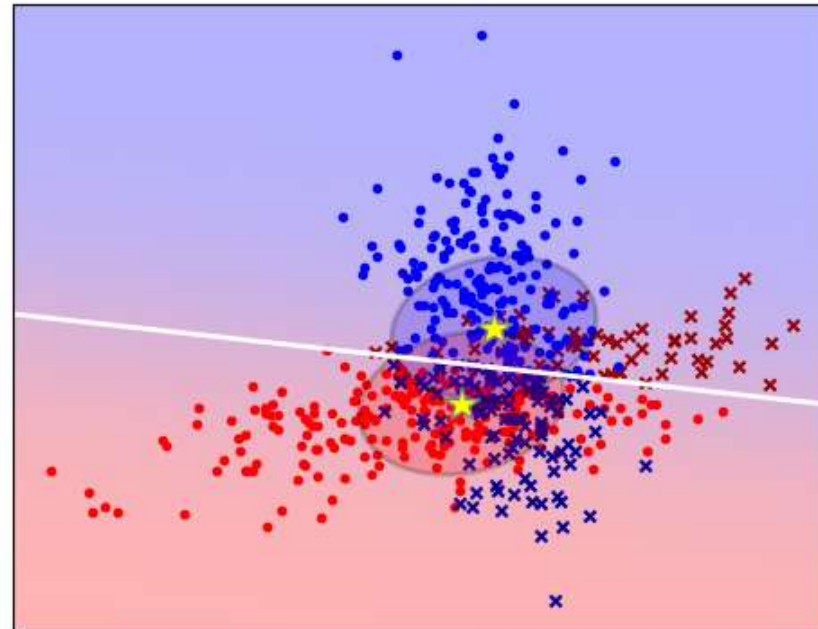
Linear Discriminant Analysis

Quadratic Discriminant Analysis

Data with  
fixed covariance



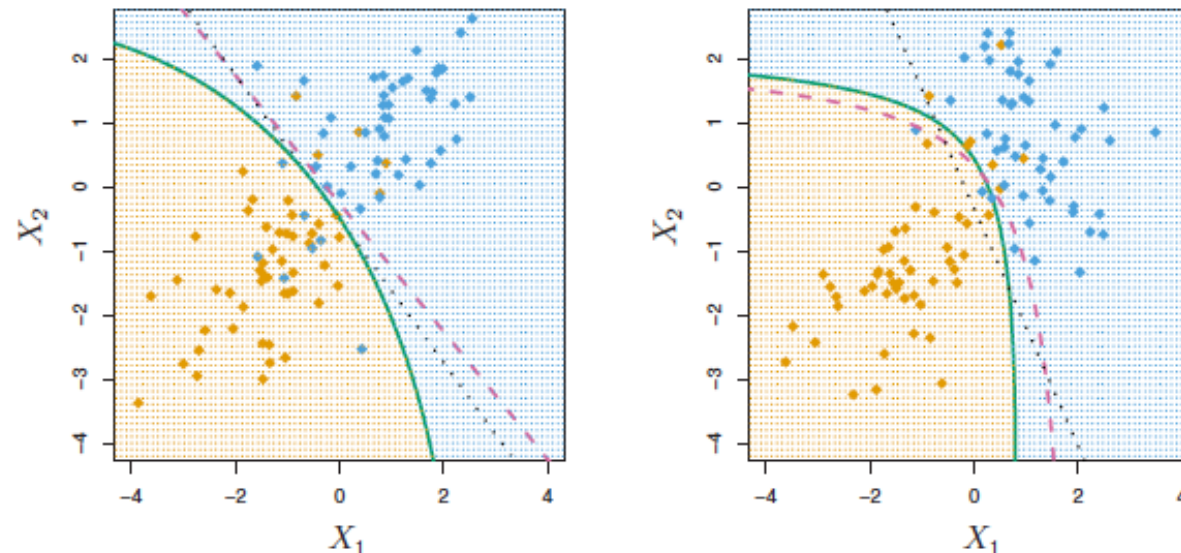
Data with  
varying covariances





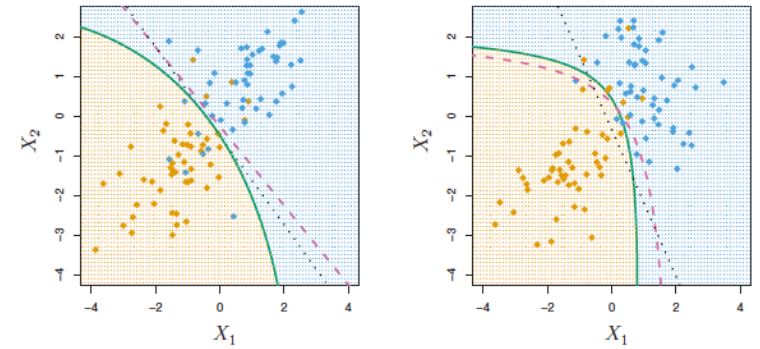
# Quadratic Discriminant Analysis

- Quadratic Discriminant Analysis (QDA) provides an alternative approach to the LDA.
- Like LDA, QDA also assumes that the observation from each class is drawn from a Gaussian distribution and we can plug the estimates for the parameter into the Bayes' Theorem to perform prediction.
- However, unlike LDA, QDA assumes that each class has its own covariance matrix.





# Quadratic Discriminant Analysis



- The figure -14 demonstrates the difference between the LDA and QDA.
- The purple dashed line represents the Bayes' decision boundary, black dotted line represents LDA and green solid represents QDA.
- The shading represents the QDA decision rule. The Left panel shows the linear Bayes' decision boundary for two class problem with  $\Sigma_1 = \Sigma_2$ .
- So, it is more accurately approximated by LDA than by QDA.
- Right hand side panel show the case where  $\Sigma_1 \neq \Sigma_2$ .
- In this case the Bayes' decision boundary is non-linear.
- Hence it is better approximated by QDA than by LDA.

Method	Pros	Cons
KNN	notable classification results no (re)training phase distance metrics error probability bounded	time consuming classification time memory utilization finding optimal k
LDA	linear decision boundary fast classification easy to implement	Gaussian assumptions training time complex matrix ops
QDA	quadratic decision boundary fast classification classification more accurate outperforms KNN and LDA	Gaussian assumptions training time complex matrix ops

# Thanks

Samatrix Consulting Pvt Ltd