Classification Techniques

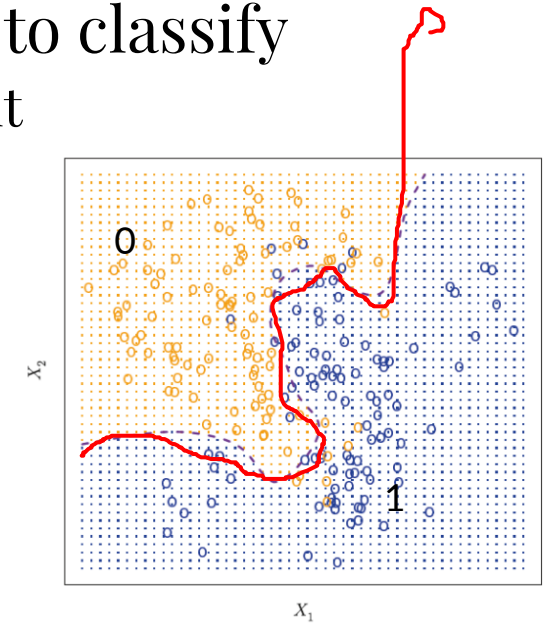
Nirav Bhatt,
Department of Biotechnology,
Robert Bosch Centre for Data Science and AI,
Indian Institute of Technology Madras,
India
Email: niravbhatt@iitm.ac.in

What is Classification?

- Linear regression
  - Response variable Y quantitative
- Scenarios
  - Fraudulent transactions of credit cards
  - Benign vs Cancerous Tumors
  - Reject or Accept quality of a product
- Qualitative variables are referred as categorical
- Classification: Y is categorical

What is Classification

- Why is it sometimes confused with Clustering?
- What does complexity in classification mean?
- What is the problem of using regression to classify
  - Coding approach for 2 classes (works OK, but assumptions?)
  - More than two classes:
    - Simple coding
    - The 0 and 1 approach for each class.



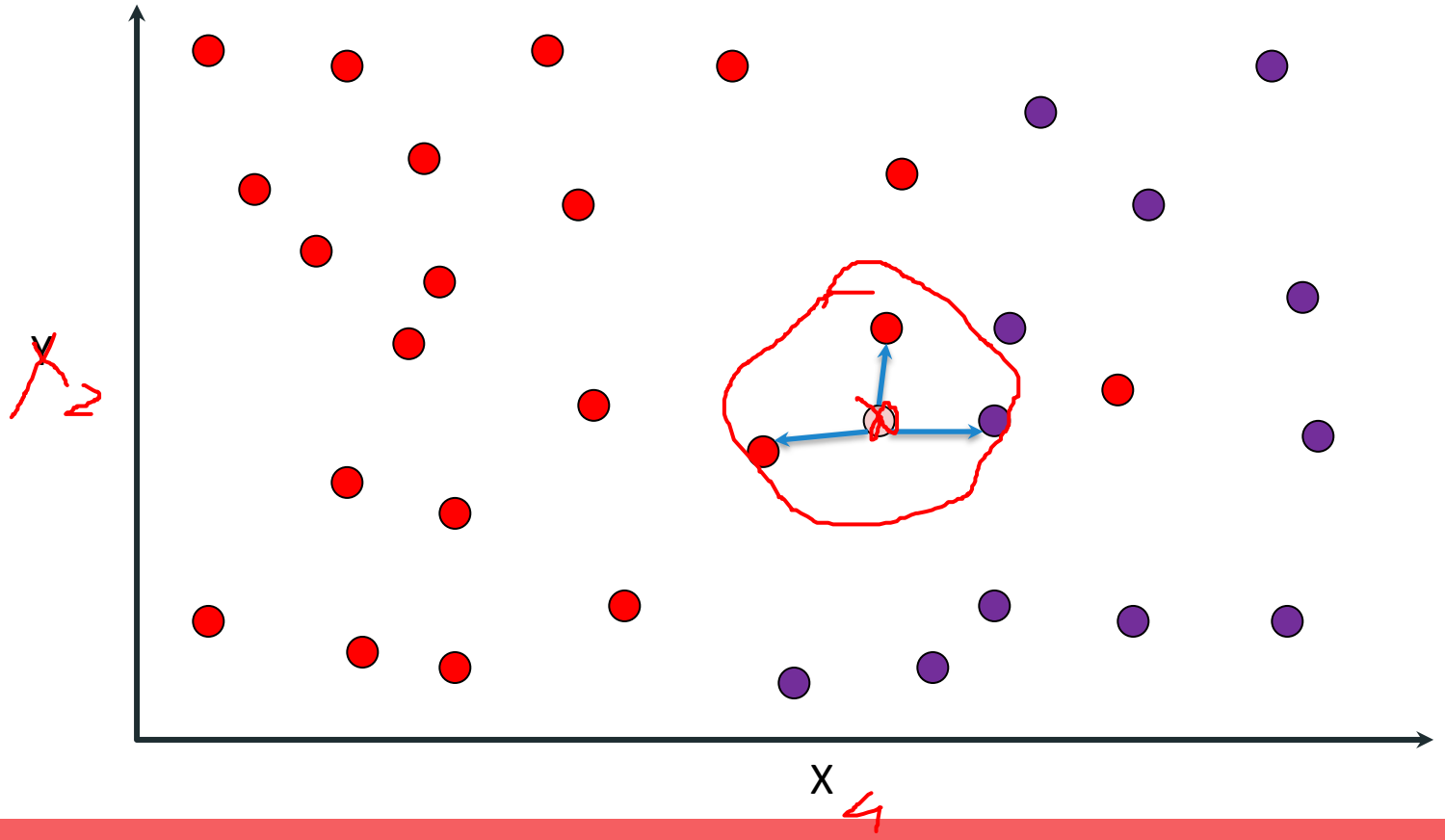
Different types of Classifiers

- k-Nearest neighbor classifier
- Decision Trees
- Naive Bayes classifier
- Logistic classifier
- Linear or Quadratic Discriminant classifier
- Logistic Regression
- Perceptrons

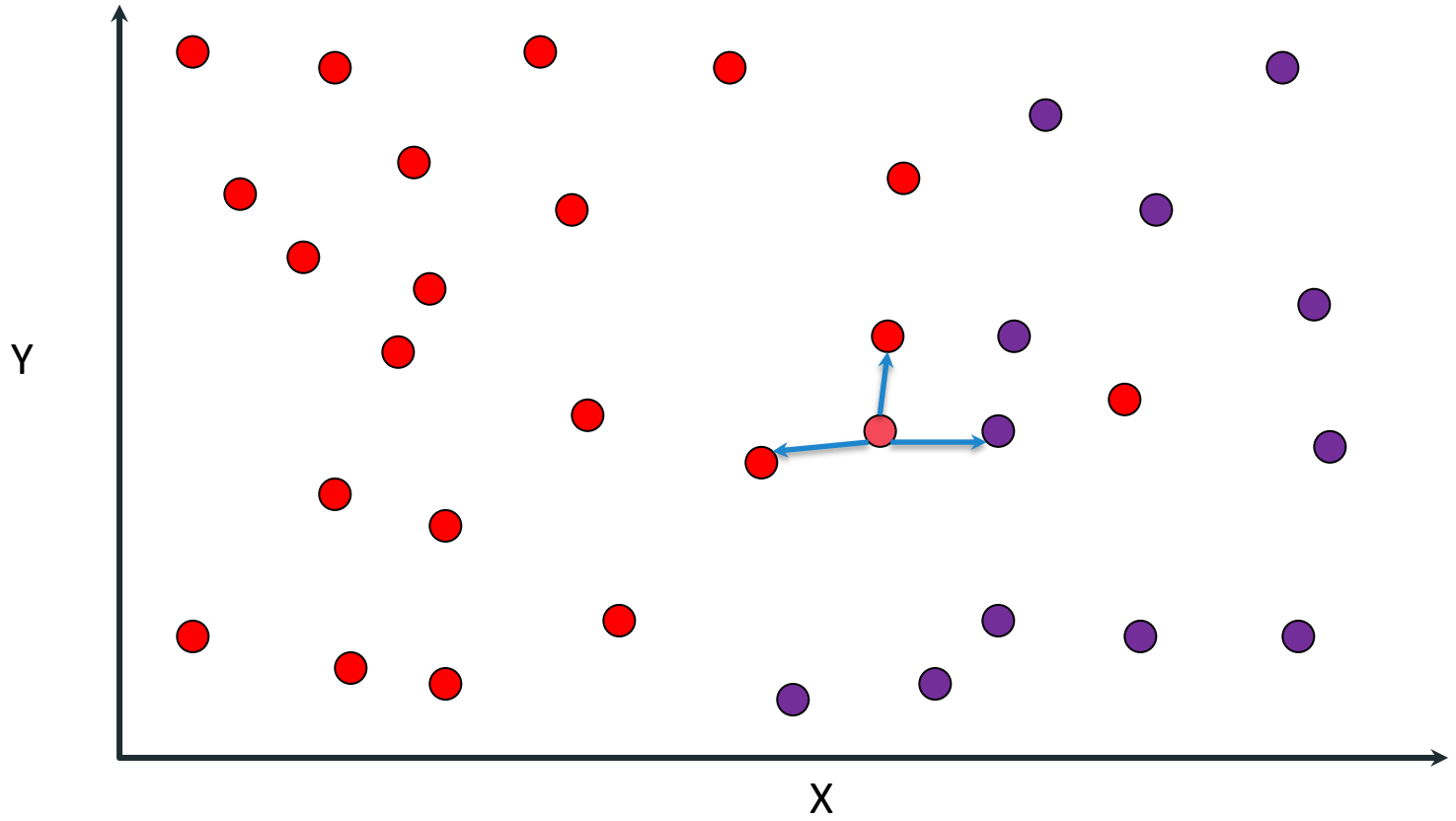
K Nearest Neighbors Classifier

- Assumption: *Small regions* have the same label
- Defined by the *k* nearest neighbours
- Label given by majority vote
- *k* nearest neighbours (kNN)

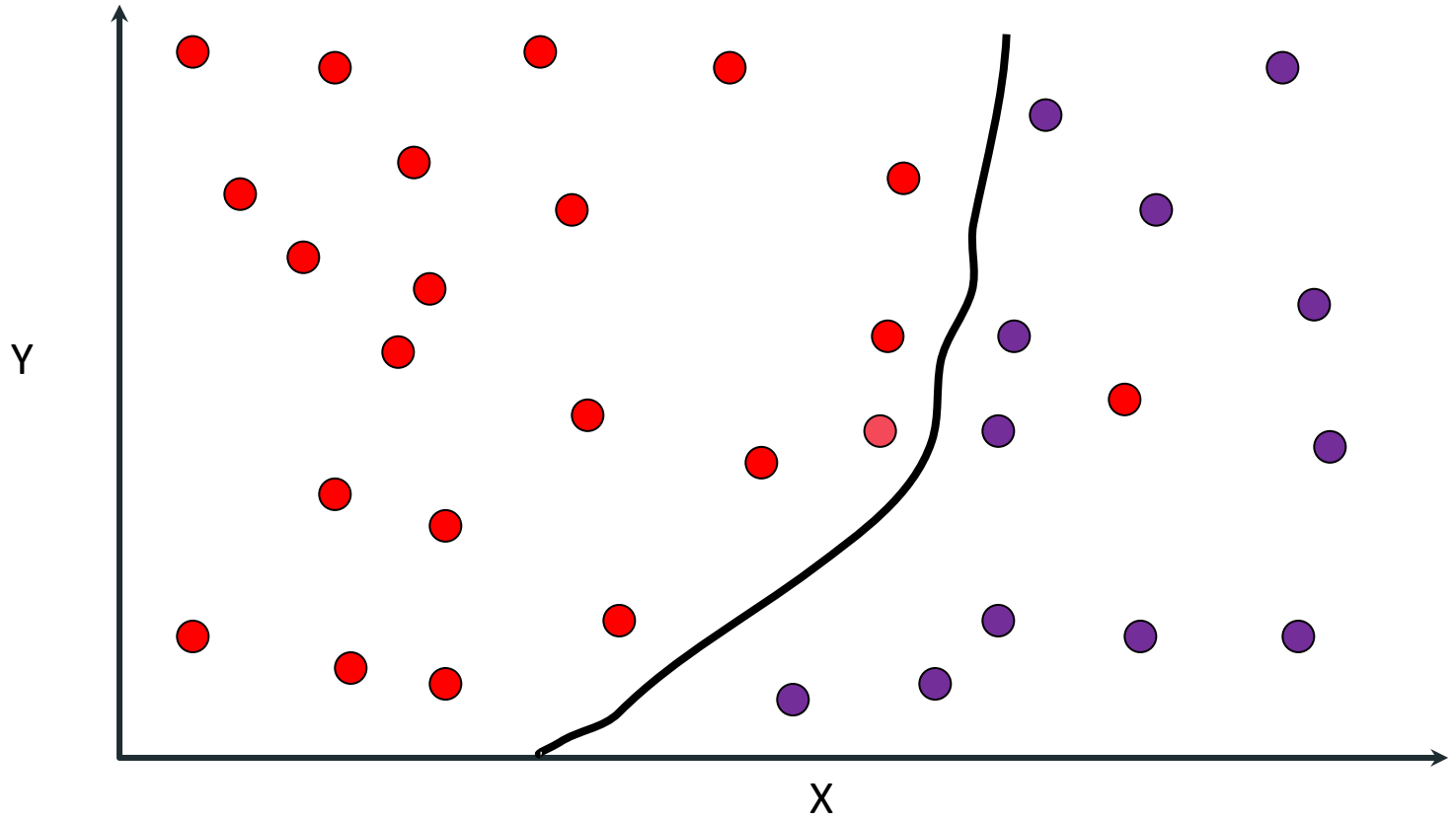
K Nearest Neighbors Classifier



kNN Classifier



K Nearest Neighbors Classifier



K Nearest Neighbors Classifier

- k Nearest Neighbors(kNN) is a non-parametric method used for classification
- It is a lazy learning algorithm where all computation is deferred until classification
- It is also an instance based learning algorithm where the function is approximated locally

K Nearest Neighbors Classifier

- Why kNN ?
  - Simplest of all classification algorithms and easy to implement
  - There is no explicit training phase and does not do any generalization of the training data
- When to use it ?
  - When there are nonlinear decision boundaries between classes
  - When the amount of data is large

K Nearest Neighbors Classifier

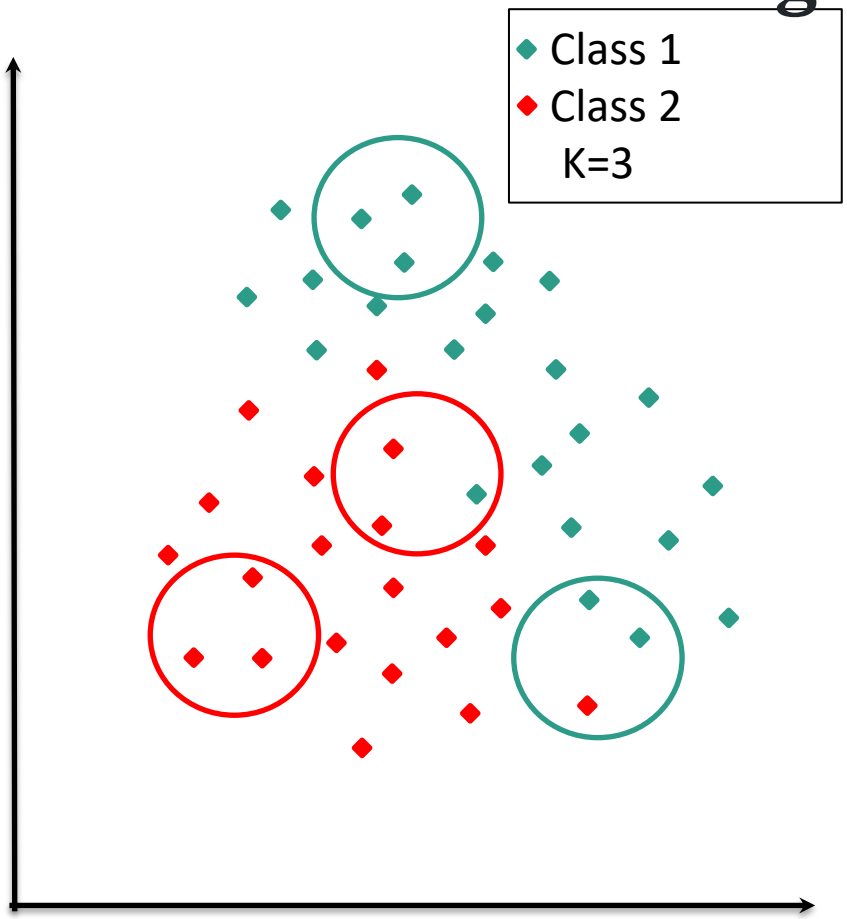
- Input features
  - Input features can be both quantitative and qualitative
- Outputs
  - Outputs are categorical values, which typically are the classes of the data
- kNN explains a categorical value using the majority votes of nearest neighbors

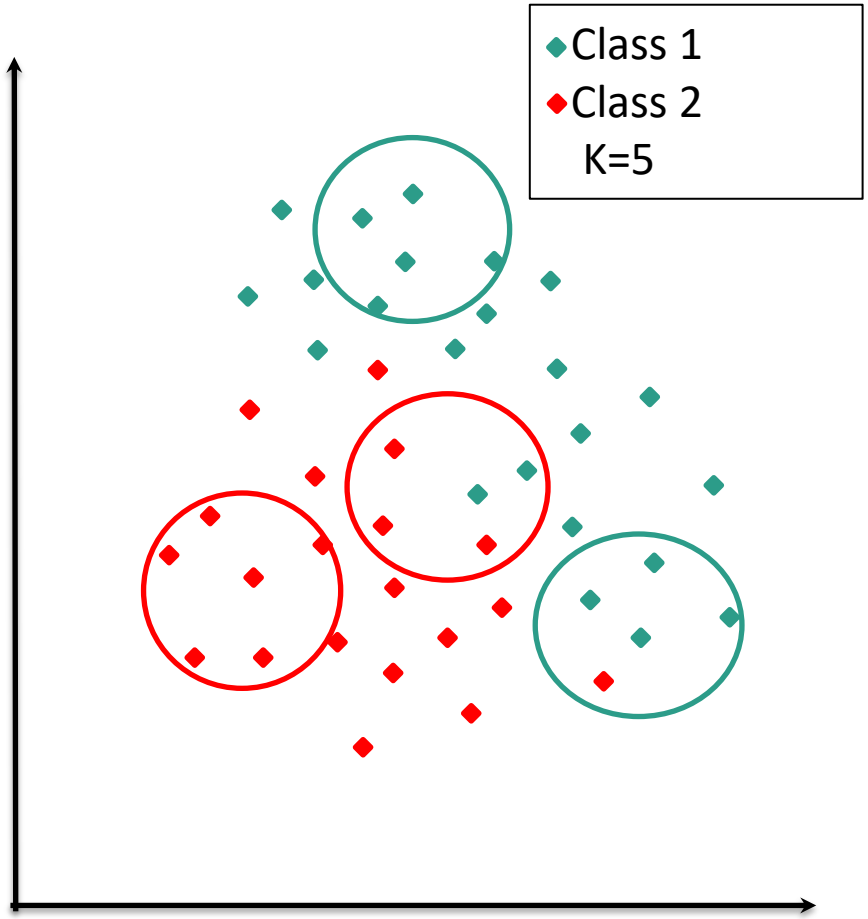
K Nearest Neighbors Classifier

Assumptions

- Being nonparametric, it does not make any assumptions about underlying data distribution
- Select the parameter k based on the data
- Requires a distance metric to define proximity between any two data points Example: Euclidean distance, Mahalanobis distance or Hamming distance

K Nearest Neighbors Classifier





K Nearest Neighbors Classifier

- Data: $\{(x_1, y_1), (x_2, y_2), \ldots\ldots, (x_n, y_n)\}$
  - Features: $(x^1, x^2, \ldots x^p)$: x_i
  - Label: y_i
- New test data x_o
  - What is the corresponding label?

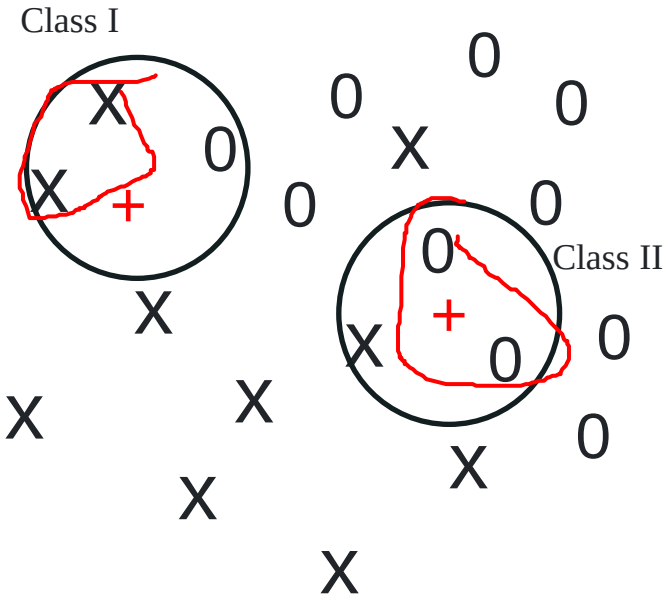
- Instant based Classifier
  - Use the data (or training data) for classification (no models)
  - Non-parametric method

K Nearest Neighbors Classifier

- How can we find the new Label?
- Old adage: Something walks and talks like peacock beware of statistics it may be hen
- kNN Idea: Something walks and talks like peacock it is high likely to be peacock not hen

K Nearest Neighbors Classifier

x: Class I and 0: Class II



- kNN classifier
  - Training Data:

$\{(x_1, y_1), (x_2, y_2), \ldots\ldots, (x_n, y_n)\}$

  - A distance Metric
  - Number of neighbors: K

K Nearest Neighbors Classifier

Algorithm

1. Data $\{(x_1, y_1), (x_2, y_2), \ldots\ldots, (x_n, y_n)\}$
2. For new data point, x_o
3. Find the nearest point(s)

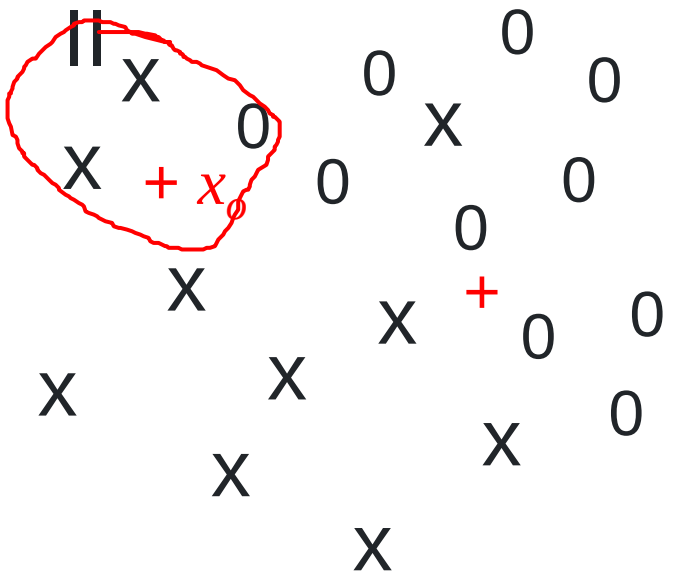
$$n^* = \underset{n=1,\ldots,n}{\operatorname{argmax}} ||x_0 - x_n||^2$$

4. Label $y_o = y_{n^*}$ based on majority votes

K Nearest Neighbors Classifier

Example:

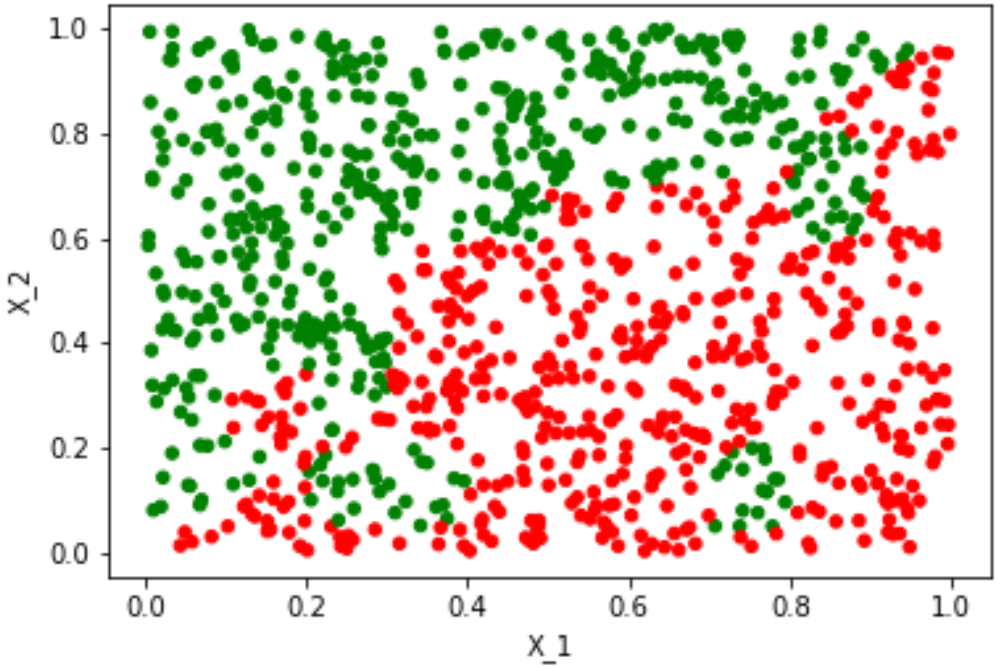
x: Class I and 0: Class II



- K=3
- Compute conditional probability
  - P(Y=Class I | x=x_o)= 0.67
  - P (Y= Class II| x=x_o)=0.33

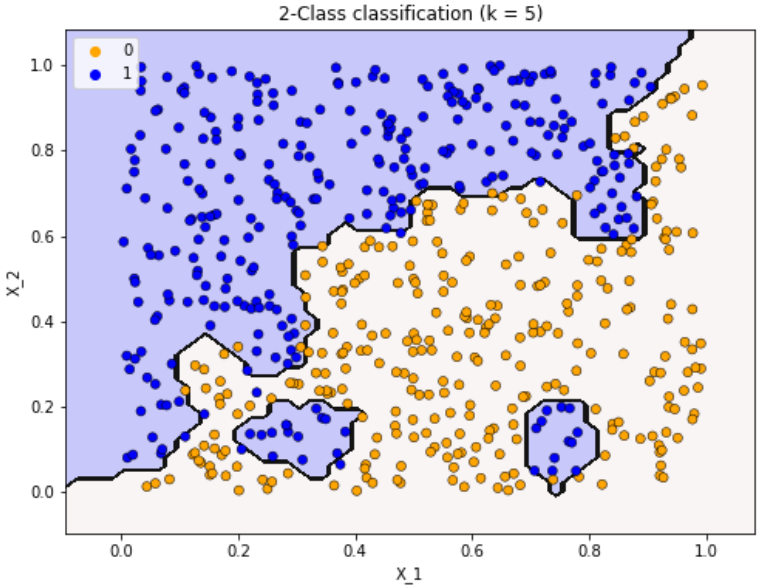
K Nearest Neighbors Classifier

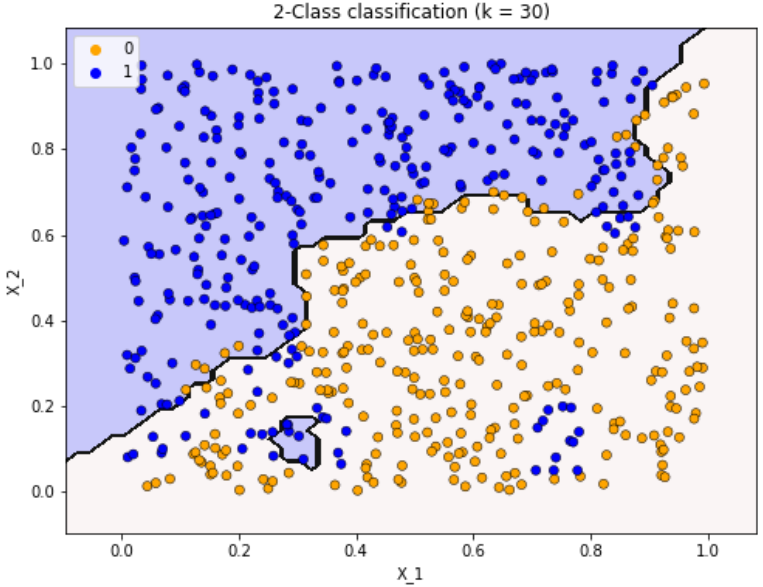
2-class classification problem with 2 features



K Nearest Neighbors Classifier

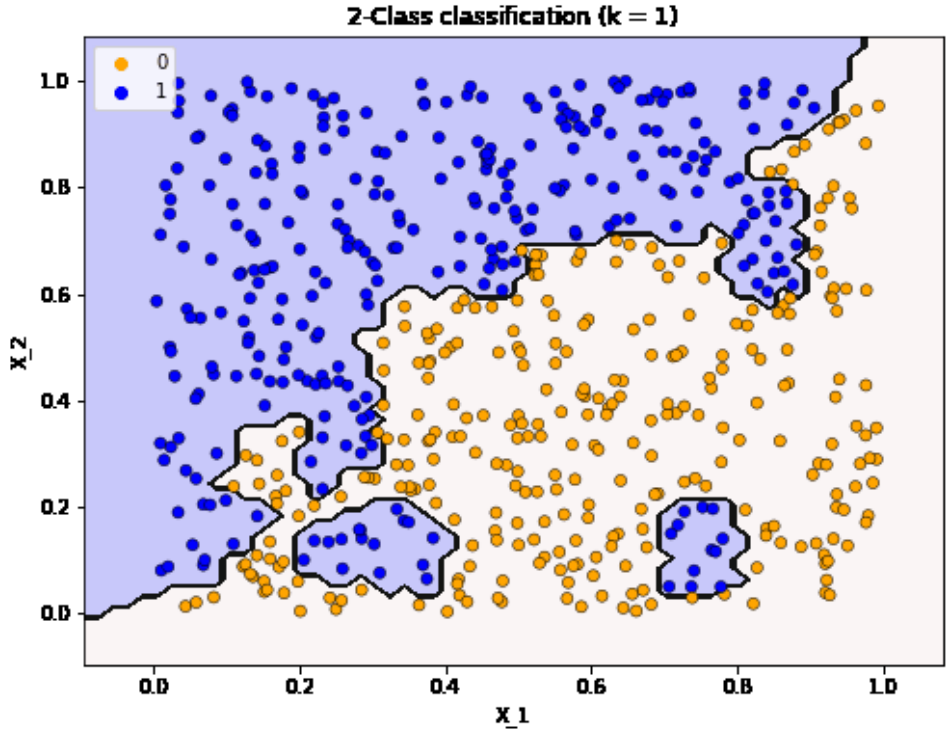
2-class classification problem with 2 features





K Nearest Neighbors Classifier

2-class classification problem with 2 features



K Nearest Neighbors Classifier

Things to consider

- Following are some things one should consider before applying kNN algorithm
  - Parameter selection
  - Presence of noise
  - Feature selection and scaling
  - Curse of dimensionality

K Nearest Neighbors Classifier

- Choice of K
- Large K value
  - Small K value
  - But sensitive to noisy data point

K Nearest Neighbors Classifier

Parameter selection

- The best choice of k depends on the data
- Larger values of k reduce the effect of noise on classification but make the decision boundaries between classes less distinct

Less flexible model

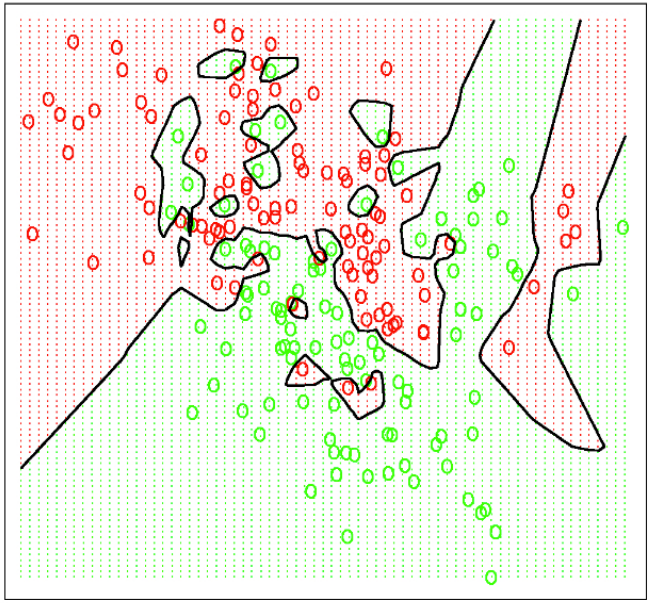
- Smaller values of k tend to be affected by the noise with a clear separation between classes

Flexible model

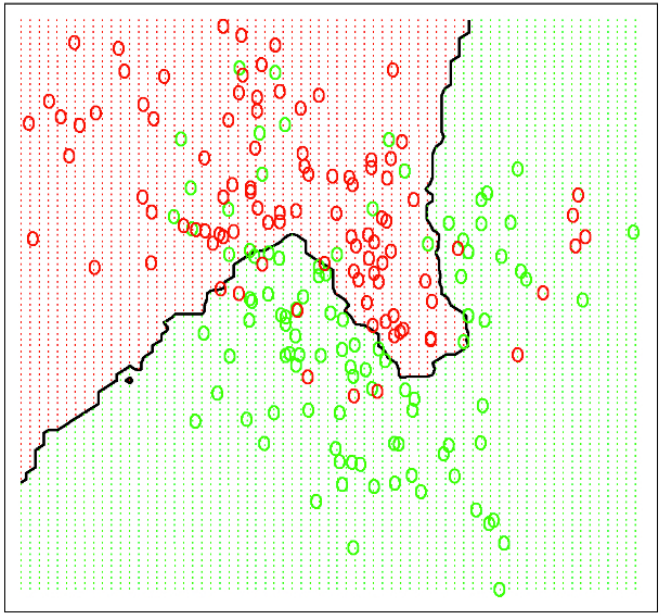
K Nearest Neighbors Classifier

Effect of *k*

1-Nearest Neighbor Classifier



15-Nearest Neighbor Classifier

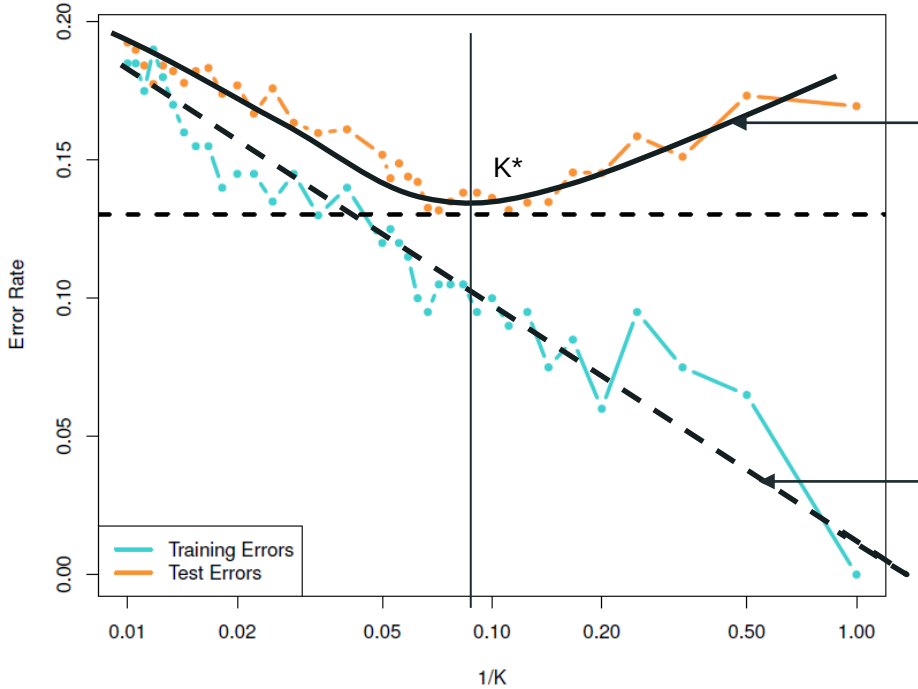


Feature selection and scaling

- It is important to remove irrelevant features
- When the number of features is too large, and suspected to be highly redundant, feature extraction is required
- If the features are carefully chosen then it is expected that the classification will be better
- PCA is a good feature selection and scaling technique

K Nearest Neighbors Classifier

How do we decide the "K"?



Test error rate

Training error rate

[1] James, G., Witten, D., Hastie, T., and Tibshirani, R. An Introduction to statistical learning, 2021

Irreducible and Reducible Errors

Mean Square Error between the actual and predicted y *using* the fit $\hat{f}(x, \hat{p})$

$$E[(y - \hat{y})^2] = [f(x, p) - \hat{f}(x, \hat{p})]^2 + Var(\epsilon)$$

Irreducible Error $Var(\epsilon)$

Reducible Error $[f(x, p) - \hat{f}(x, \hat{p})]^2$

Bias-Variance Trade-off and Prediction error

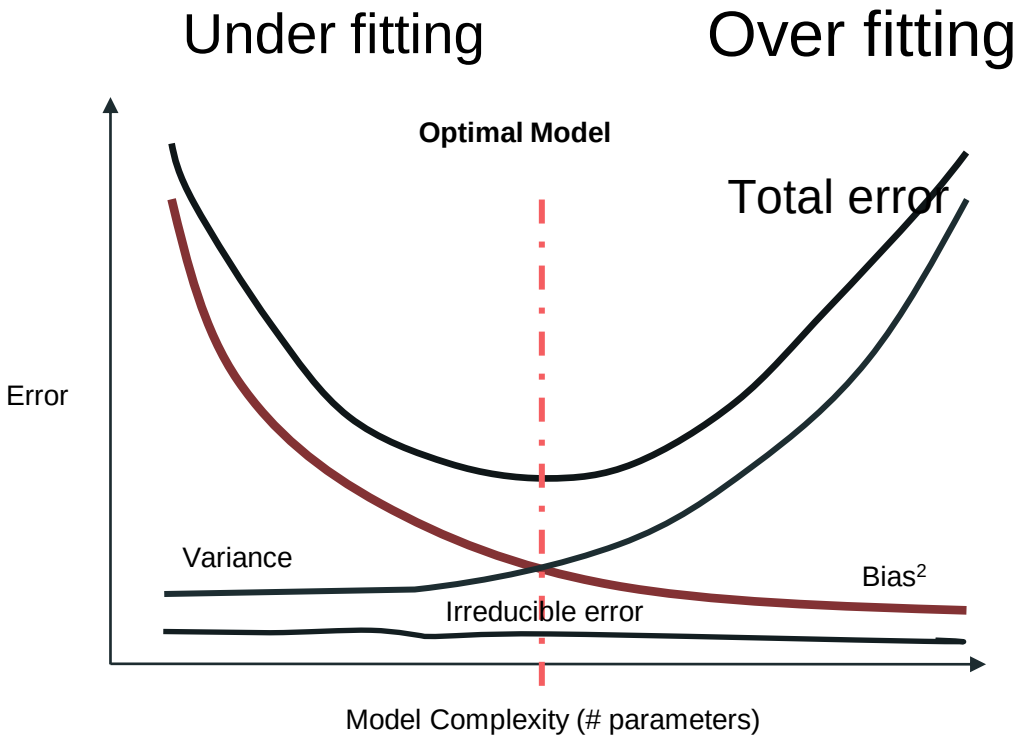
kNN MSE

$$E[(\hat{y}_{x_o} - y)^2] = Var(\epsilon) + \frac{1}{K}\sigma^2 + (f(x_o) - \frac{1}{K}\sum_{i \in \mathcal{A}} f(x_i))^2$$

Linear Regression MSE

$$E[(\hat{y} - y)^2] = \sigma^2 + (\mathbf{x}_p^T Var[\hat{\boldsymbol{\beta}}_p]\mathbf{x}_p) + (\mathbf{x}_p^T \mathbf{A}\boldsymbol{\beta}_r - \mathbf{x}_r\beta_r)^2$$

Bias-Variance Trade-off



K Nearest Neighbors Classifier

- Assumption: *Small regions* have the same label
- Defined by the *k* nearest neighbours
- Label given by majority vote
- Performs badly when
  - data is sparse
  - large dimensional input space

- Challenge: Efficiently finding nearest neighbours
  - Near Neighbours