TELKOM
DIGITAL TALENT
INCUBATOR 2020

Modul 5: Classification

Organized by:
CAE

Module Objectives

- Understand what is classification and its application
- Understand how classification algorithm work
- Understand how to build classification model

Module Overview

Topics

- Classification Model
- Decision Tree
- Random Forest
- Naive Bayes
- KNN
- Build Classification Model using Python

Activities

- Group Discussion
- Coding Practice

Classification

- In machine learning and statistics, classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, based on a training set of data containing observations (or instances) whose category membership is known.
- Examples are assigning a given email to the "spam" or "non-spam" class, and assigning a diagnosis to a given patient based on observed characteristics of the patient (sex, blood pressure, presence or absence of certain symptoms, etc.)

Use of Classification

- **Handwriting recognition:** used to interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices
- **Web search engine:** used to classify information on World Wide Web
- **Speech recognition:** used for recognition and translation of spoken language into text by computers.

Use of Classification (Cont.)

- **Biological classification:** used for classifying biological organism based on shared characteristics (taxonomy)
- **Credit scores:** used to determine who qualifies for a loan, at what interest rate, and what credit limits.

Classification Algorithm

- Decision Tree
- Random Forest
- Bayesian
- Lazy Learner (kNN)

Decision Tree

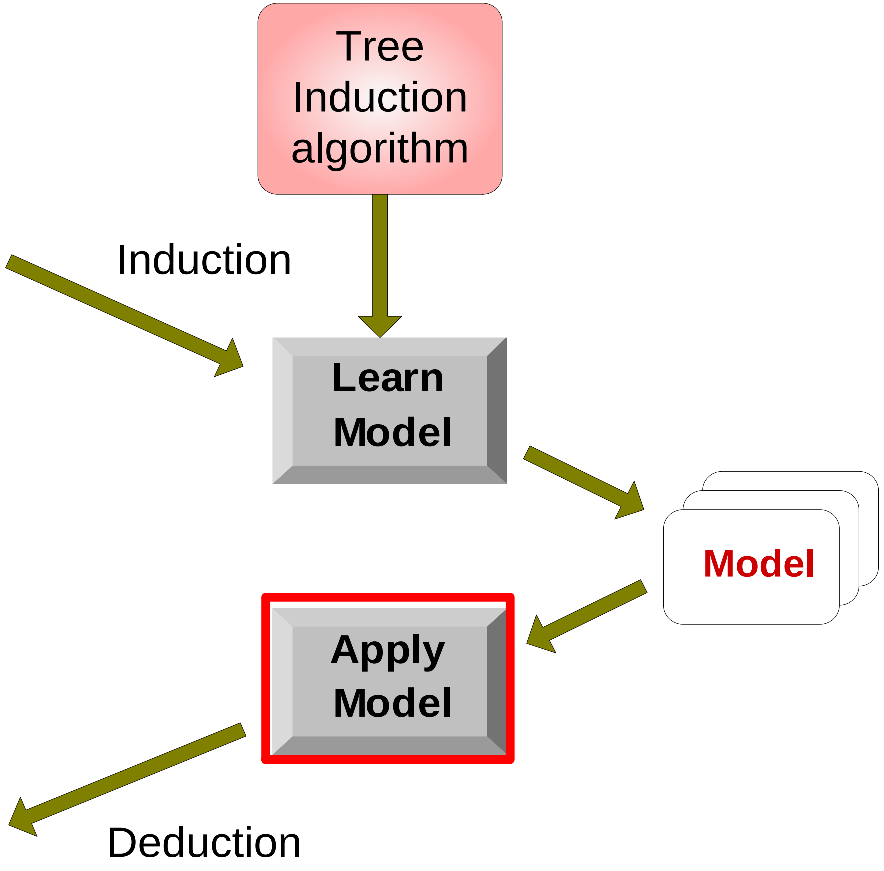
- Decision tree is a tree shaped diagram used to determine a course of action. Each branch of the tree represents a possible decision, occurrence or reaction

- Important Terms
  - Entropy : measure of randomness in the dataset
  - Information gain : measure of decrease in entropy after dataset is split
  - Leaf node : carries the classification or decision
  - Root node: top most decision node

Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

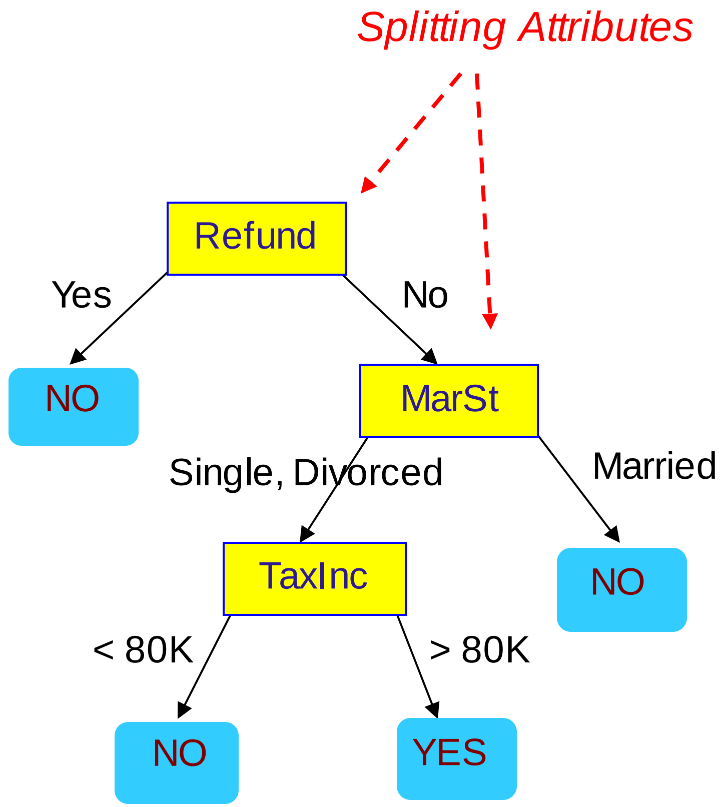


Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

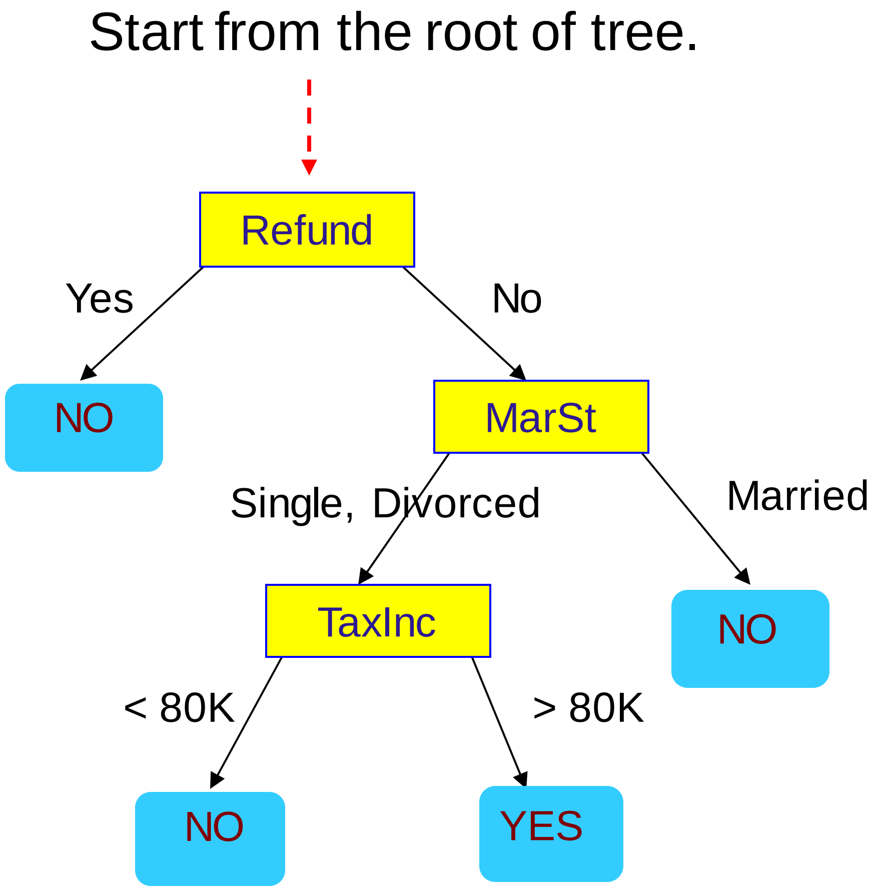
Test Set

Decision Tree Classification Task

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Apply Model on Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

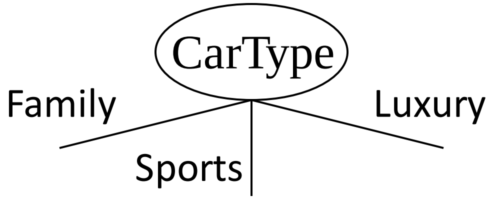
Specify Attribute Test Condition

- Depends on attribute types
  - Discrete
    - Nominal
    - Ordinal
  - Continuous

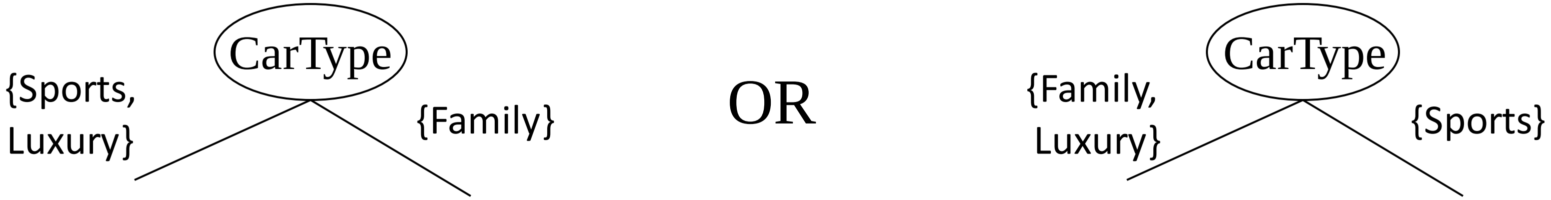
- Depends on number of ways to split
  - Multi-way split
  - Binary split

Splitting Based on Nominal Attributes

- Multi-way split: Use as many partitions as distinct values.

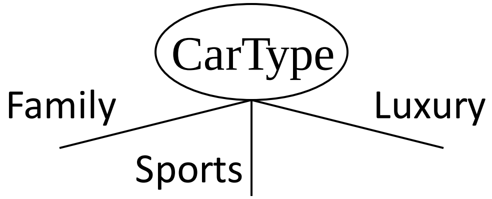


- Binary split: Divides values into two subsets. Need to find optimal partitioning.

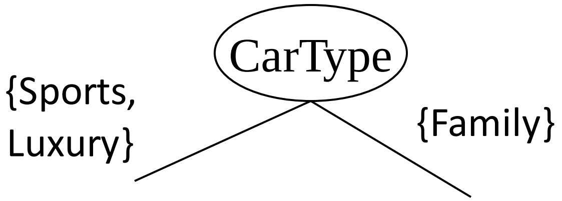


Splitting Based on Ordinal Attributes

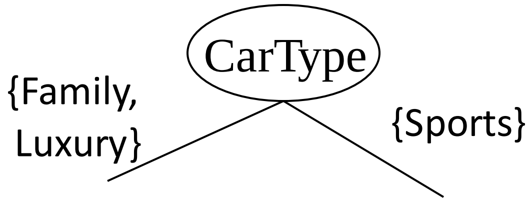
- Multi-way split: Use as many partitions as distinct values.



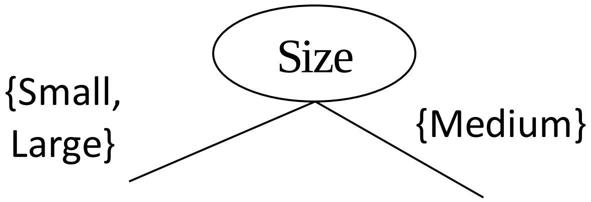
- Binary split: Divides values into two subsets.
  Need to find optimal partitioning.



OR

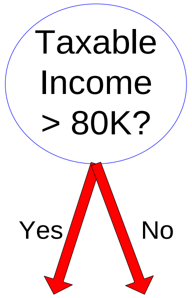


- What about this split?

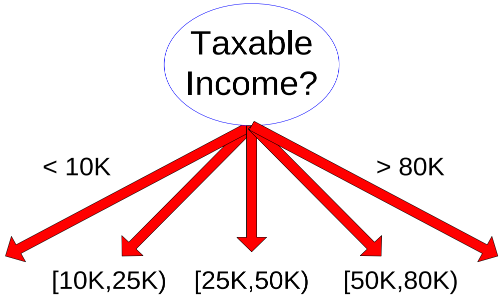


Splitting Based on Continuous Attributes

- Different ways of handling
  - Discretization to form an ordinal categorical attribute
    - Static – discretize once at the beginning
    - Dynamic – ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.
- Binary Decision: (A < v) or (A v)
  - consider all possible splits and finds the best cut
  - can be more compute intensive



(i) Binary split



(ii) Multi-way split

Decision Tree Summary

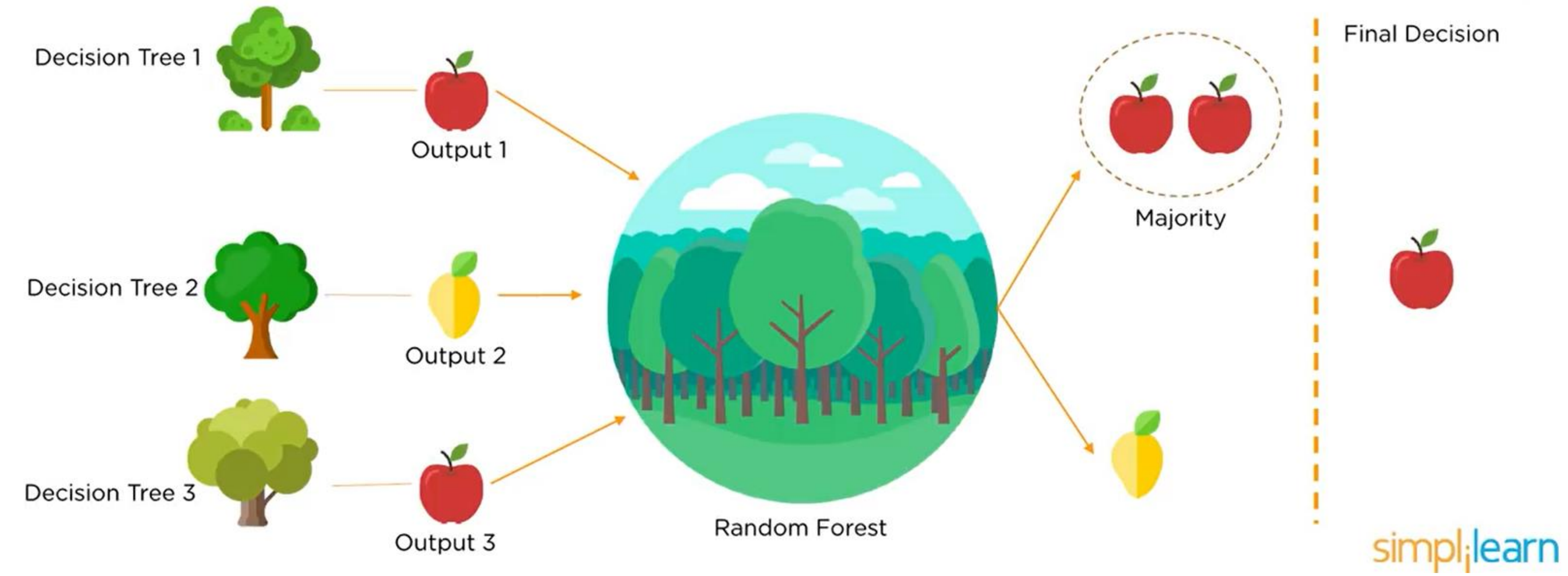
- Advantages:
  - Inexpensive to construct
  - Extremely fast at classifying unknown records
  - Easy to interpret for small-sized trees
  - Accuracy is comparable to other classification techniques for many simple data sets

- Disadvantages:
  - Overfitting when algorithm capture noise in the data
  - The model can get unstable due to small variation of data
  - Low biased tree: difficult for the model to work with new data

Random Forest

- Random forest or Random Decision Forest is a method that operates by constructing multiple decision trees during training phases
- The Decision of the majority of the trees is chosen as final decision.

Random Forest

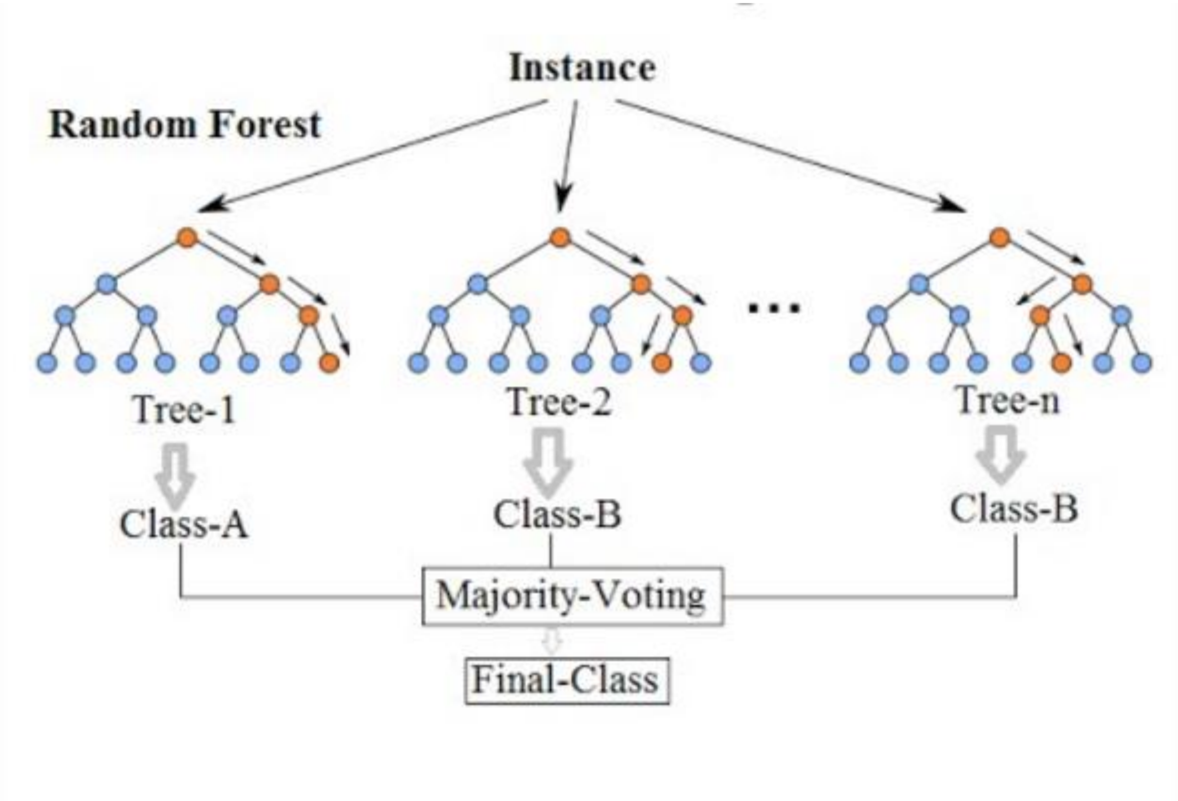


Random Forest Summary

- Random Forest:
  - Each classifier in the ensemble is a decision tree classifier and is generated using a random selection of attributes at each node to determine the split
  - During classification, each tree votes and the most popular class is returned
- Two Methods to construct Random Forest:
  1. Forest-RI (random input selection): Randomly select, at each node, F attributes as candidates for the split at the node. The CART methodology is used to grow the trees to maximum size
  2. Forest-RC (random linear combinations): Creates new attributes (or features) that are a linear combination of the existing attributes (reduces the correlation between individual classifiers)
- Insensitive to the number of attributes selected for consideration at each split, and faster than bagging (grouping based on frequency) or boosting

TELKOM
DIGITAL TALENT
INCUBATOR 2020

How Random Forest Work



Organized by:

CAE

Random Forest Summary

- Advantages:
  - It can be used for both regression and classification tasks and that it's easy to view the relative importance it assigns to the input features.
  - It is also considered as a very handy and easy to use algorithm, because it's default hyper-parameters often produce a good prediction result.
- Disadvantages:
  - Many trees can make the algorithm to slow and ineffective for real-time predictions. A more accurate prediction requires more trees, which results in a slower model.
  - It is a predictive modeling tool and not a descriptive tool.

Naive Bayes Classifier

- Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set.
- There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.

Naive Bayes Classifier

A probabilistic framework for solving classification problems

- Conditional Probability:

$$P(C \mid A) = \frac{P(A,C)}{P(A)}$$

$$P(A \mid C) = \frac{P(A,C)}{P(C)}$$

- Bayes theorem:

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

Example of Naive Bayes Classifier

Given:

- A doctor knows that meningitis causes stiff neck 50% of the time
- Prior probability of any patient having meningitis is 1/50,000
- Prior probability of any patient having stiff neck is 1/20

If a patient has stiff neck, what's the probability he/she has meningitis?

$$P(M \mid S) = \frac{P(S \mid M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Bayesian Classifiers

- Consider each attribute and class label as random variables
- Given a record with attributes (A1, A2,…,An)
  - Goal is to predict class C
  - Specifically, we want to find the value of C that maximizes P(C| A1, A2,…,An)
- Can we estimate P(C| A1, A2,…,An) directly from data?

Bayesian Classifiers

- Approach:
  - compute the posterior probability P(C | A1, A2, …, An) for all values of C using the Bayes theorem

$$\boxed{P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}}$$

  - Choose value of C that maximizes
    P(C | A1, A2, …, An)
  - Equivalent to choosing value of C that maximizes
    P(A1, A2, …, An|C) P(C)
- How to estimate P(A1, A2, …, An | C)?

Naïve Bayes Classifier

- Assume independence among attributes Ai when class is given:
  - P(A1, A2, …, An |C) = P(A1| Cj) P(A2| Cj)… P(An| Cj)
  - Can estimate P(Ai| Cj) for all Ai and Cj.
  - New point is classified to Cj if P(Cj) P(Ai| Cj) is maximal.

How to Estimate Probabilities from Data?

- Class: P(C) = Nc/N
  - e.g., P(No) = 7/10,
    P(Yes) = 3/10

- For discrete attributes:

  P(Ai | Ck) = |Aik|/ Nc

  - where |Aik| is number of instances having attribute Ai and belongs to class Ck
  - Examples:

P(Status=Married|No) = 4/7
P(Refund=Yes|Yes)=0

- For continuous attributes:
- Discretize the range into bins
- one ordinal attribute per bin
- violates independence assumption
- Two-way split: (A < v) or (A > v)
- choose only one of the two splits as new attribute
- Probability density estimation:
- Assume attribute follows a normal distribution
- Use data to estimate parameters of distribution
  (e.g., mean and standard deviation)
- Once probability distribution is known, can use it to estimate the conditional probability P(Ai|c)

Naive Bayes Summary

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

A: attributes

M: mammals

N: non-mammals

$$P(A|M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A|N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A|M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A|N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

P(A|M)P(M) > P(A|N)P(N)

=> Mammals

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Normal distribution:

$$P(A_i \mid c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(A_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

  – One for each (A_i,c_i) pair

- For (Income, Class=No):
  – If Class=No
    - sample mean = 110
    - sample variance = 2975

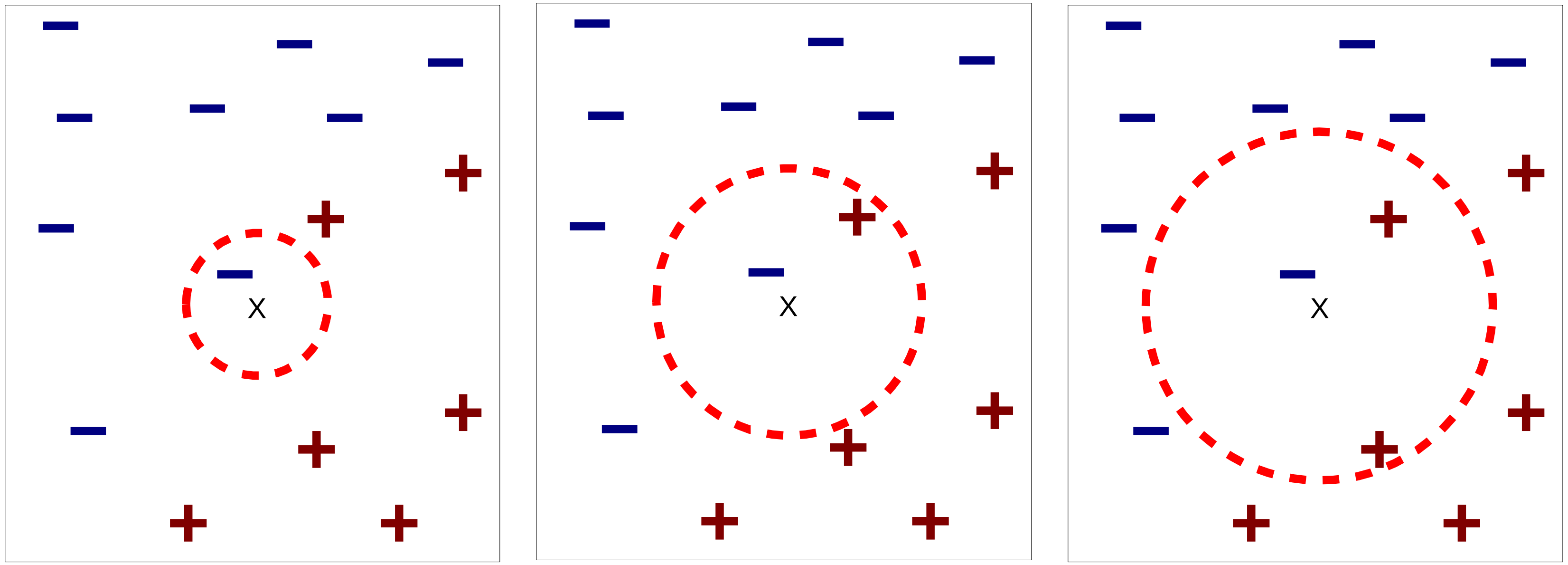
$$P(Income = 120 \mid No) = \frac{1}{\sqrt{2\pi}(54.54)} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

Naive Bayes Summary

- Robust to isolated noise points
- Handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
- Independence assumption may not hold for some attributes
- Use other techniques such as Bayesian Belief Networks (BBN)

K Nearest Neighbourhood

- K-nearest neighbors of a record x are data points that have the k smallest distance to x.

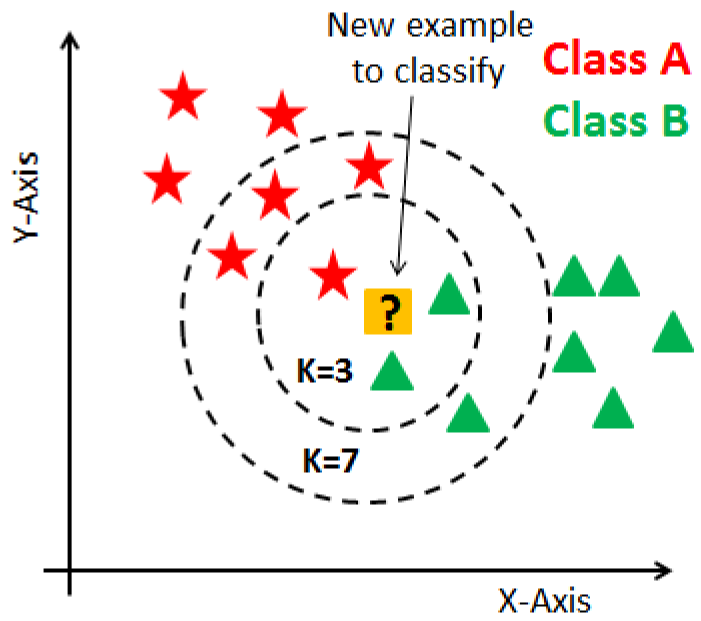


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

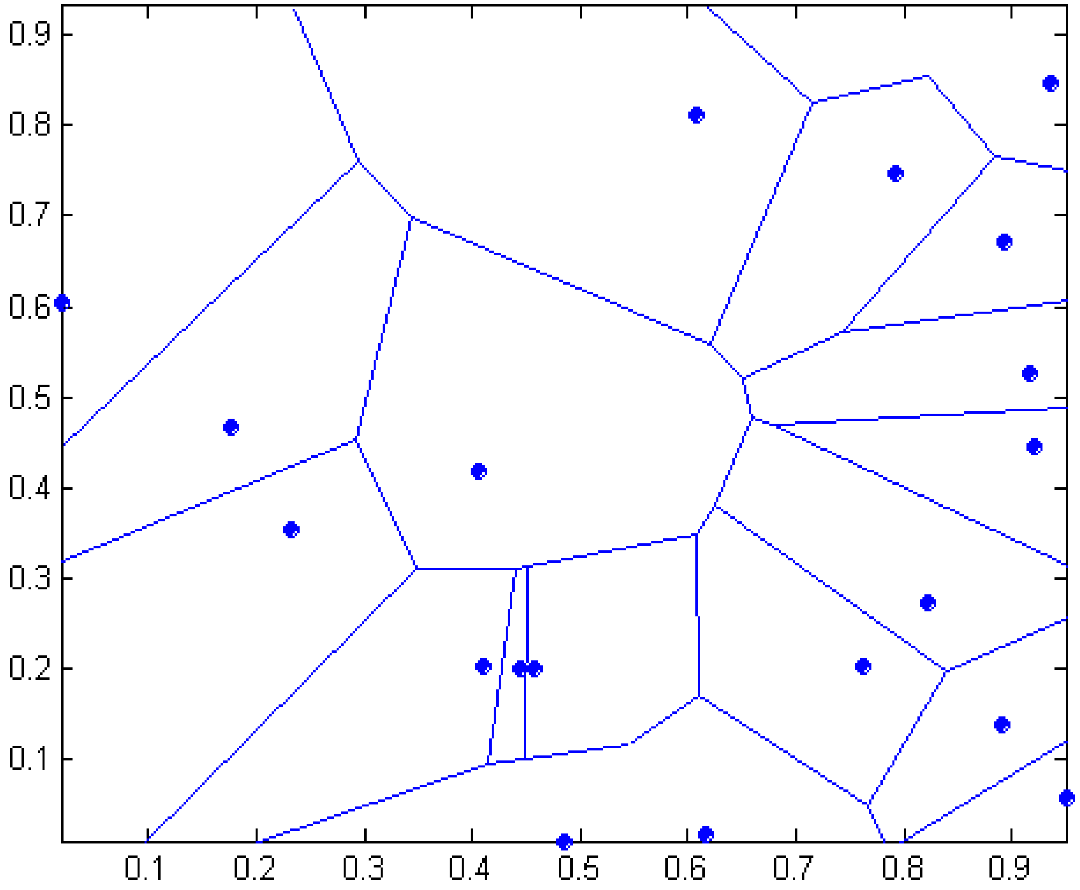
Nearest-Neighbor Classifiers

- Requirement
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of k, the number of nearest neighbors to retrieve

- To classify an unknown record:
  - Compute distance to other training records
  - Identify k nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record



Voronoi Diagram



KNN Summary

- Advantages:
  - Simple technique that is easily implemented
  - Building model is cheap
  - Extremely flexible classification scheme
- Disadvantages:
  - Classifying unknown records are relatively expensive
  - Requires distance computation of k-nearest neighbors
  - Computationally intensive, especially when the size of the training set grows
  - Accuracy can be severely degraded by the presence of noisy or irrelevant features

Metrics for Performance Evaluation

- Focus on the predictive capability of a model rather than how fast it takes to classify or build models, scalability, etc.
- Confusion Matrix:

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Metrics for Performance Evaluation

- Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	A (TP)	B (FN)
	Class=No	C (FP)	D (TN)

Limitation of Accuracy

- Consider a 2-class problem
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10

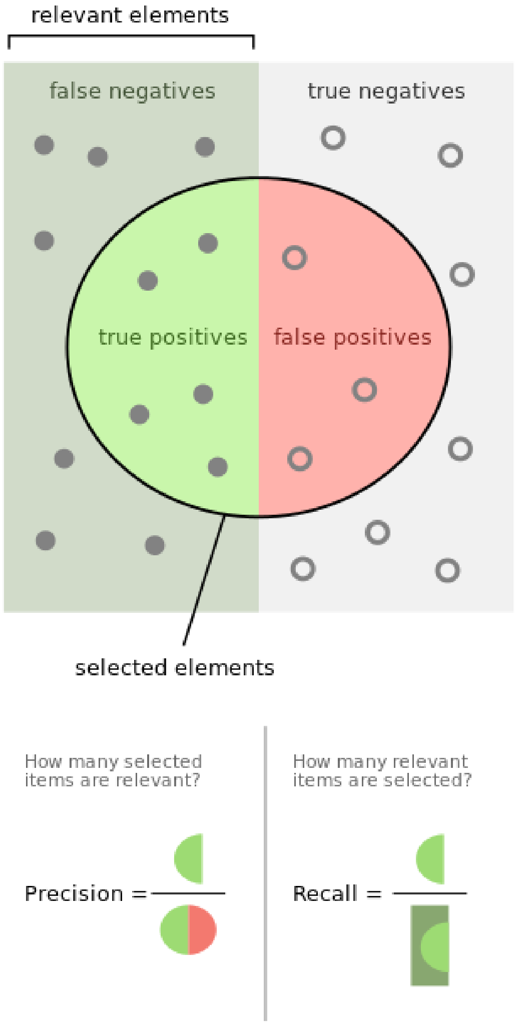
- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %
  - Accuracy is misleading because model does not detect any class 1 example

Cost-Sensitive Measures

$$\text{Precision (p)} = \frac{a}{a+c}$$

$$\text{Recall (r)} = \frac{a}{a+b}$$

$$\text{F - measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$



Practice with Python

- Practice Link:
- https://github.com/rc-dbe/dti

TELKOM
DIGITAL TALENT
INCUBATOR 2020

Assignment

- Create a classification model using “bank-marketing.csv” dataset from Kaggle https://www.kaggle.com/janiobachmann/bank-marketing-dataset
- Choose minimal 2 algorithm
- Compare performance of algorithm by using appropriate metric
- Use Google Collab (or Jupyter Notebook if you want)
- Put the code in your GitHub
- Make it informative as possible

Organized by:
CAE