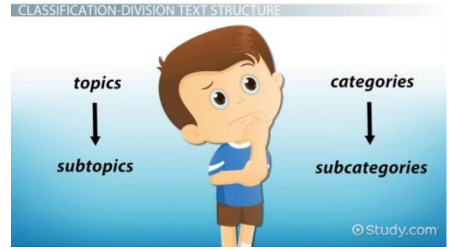
Tour Version

Data Mining

Tour 4
Classification

Classification

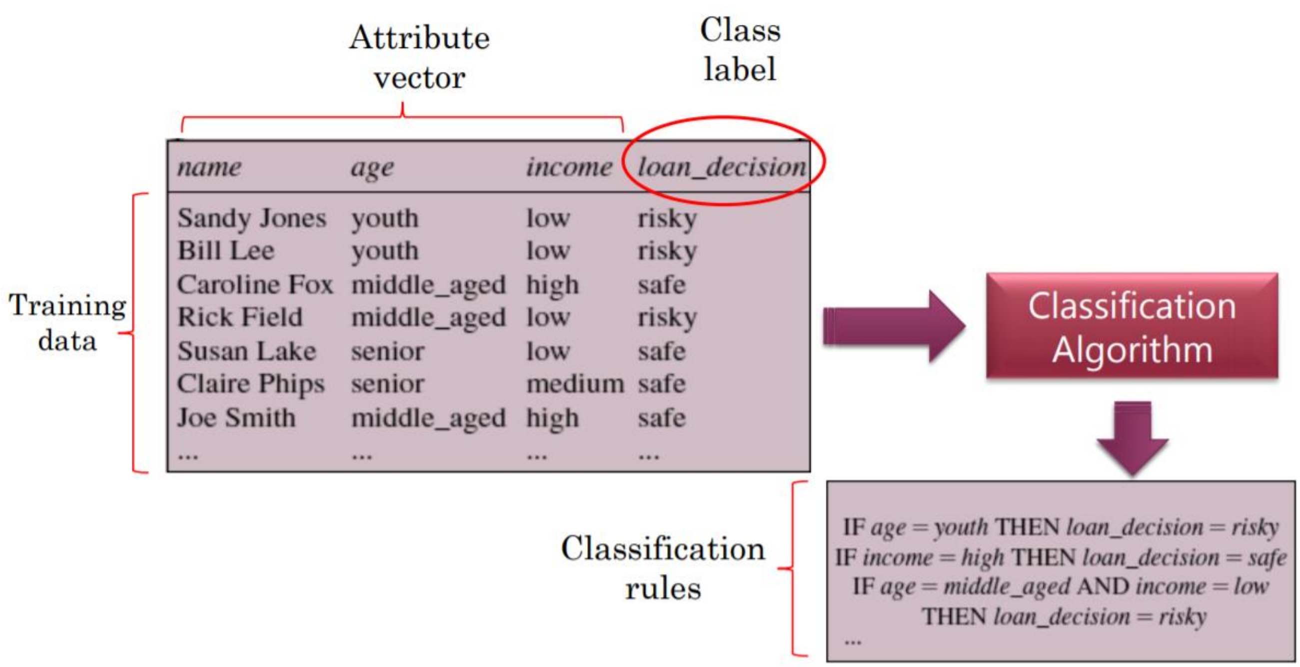


What is The Classification

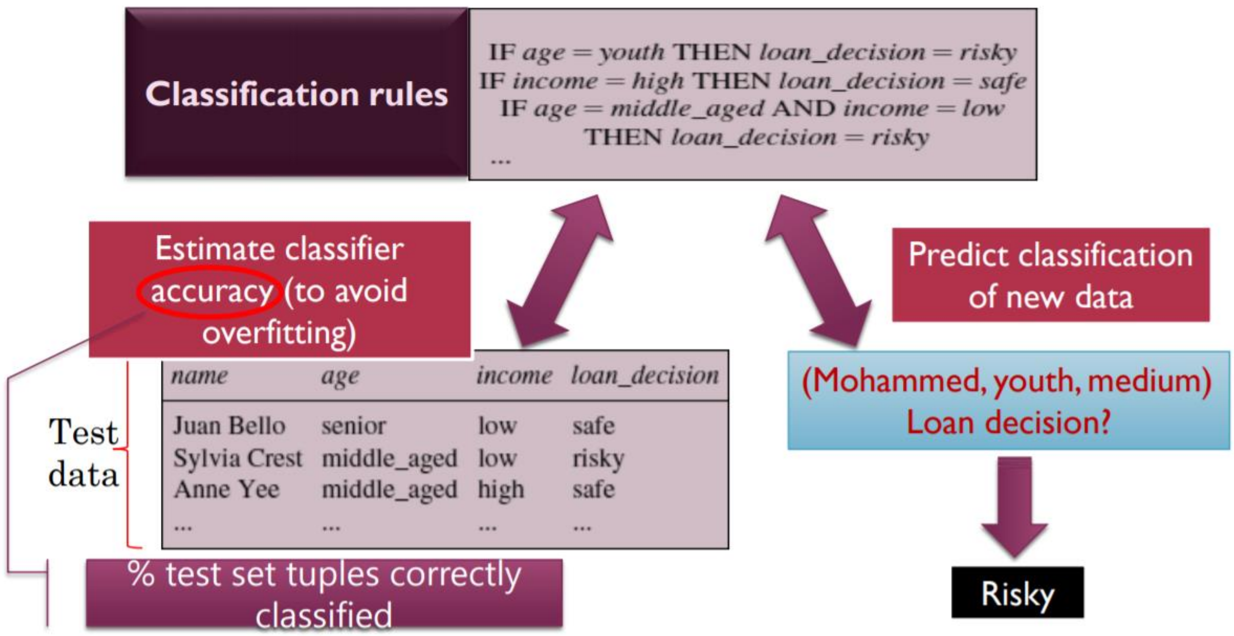
- Classification is a data analysis task where a model is constructed to predict class labels (categories)
- Motivation: Prediction
- المحرك الأساسى للتصنيف هو التنبؤ رغم الأغراض التانية زى التوصيف للبيانات
- **Descriptive vs Predictive Tasks** Chapter 1
- Is a bank loan applicant “safe” or “risky”?
- بيكون متعلم من البيانات اللى فاتت هل مثلا الشخص بالامكانيات المتاحة هل هيكون اقتراضه أمن او خطر
- Which treatment is better for patient, “treatmentX” or “treatmentY”?

two-step process

1. Learning (training) step → construct classification model
   - Build classifier for a predetermined set of classes
   - Learn from a training dataset (data tuples + their associated classes) → Supervised Learning
   - طبعا دى محتاجة داتا عشان يتدرب عليها و هنا بيدخل دور ال
   - Machine learning
2. Classification step → model is used to predict class labels for given data (test set)

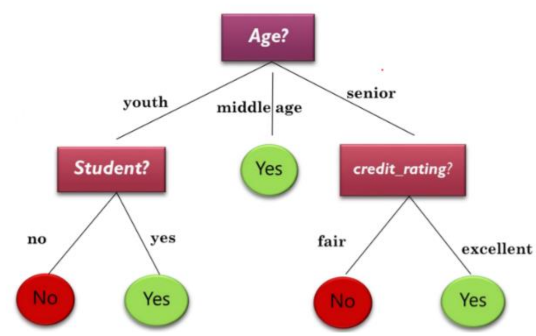


- بيحتاج داتا يتدرب عليها فيصنع مودل عبارة عن مجموعة من القواعد عن طريق الجوريزم معين
- Age and Income is the attributes that can judge about loan

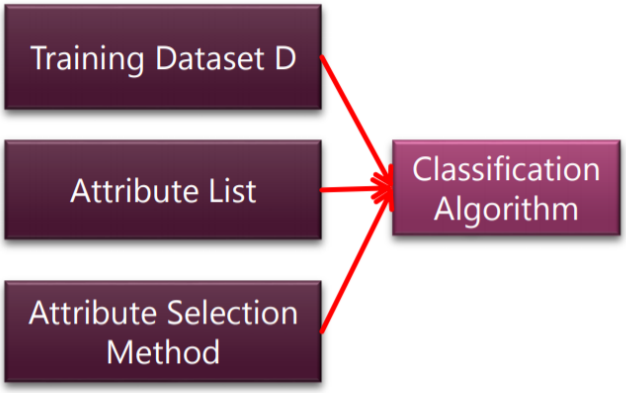


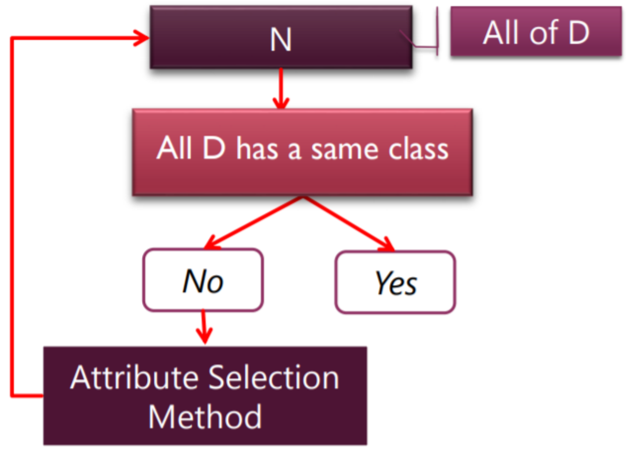
- بعد ما يطلع القواعد بنختبر مدى دقة المودل ده عن طريق انى بجيب داتا تانية بختبر عليها مدى صحة القواعد دى
- و ان كان مدى دقة القواعد سليمة كفاية ,, منكم اتنبأ لو فيه داتا من غير تصنيف اقدر اصنفها بالقواعد دى

Decision Tree Induction

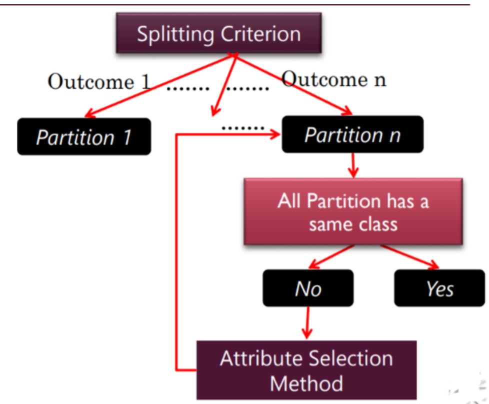


- Learning of decision trees from training dataset
- Decision tree → A flowchart-like tree structure
  - Internal node → a test on an attribute
  - Branch → a test outcome
  - Leaf node → a class label
- Constructed tree can be binary or otherwise
- Benefits
  - No domain knowledge required
  - No parameter setting
  - Can handle multidimensional data
  - Easy-to-understand representation
  - Simple and fast

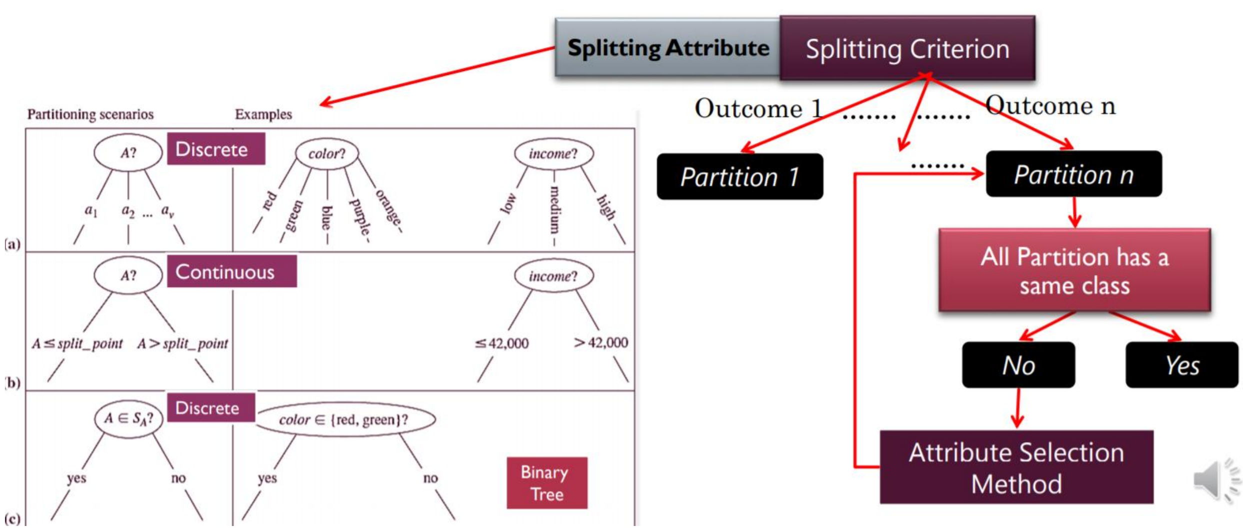




- عشان نعمل التصنيف محتاجين نعرف 3 حاجات هي ايه الداتا اللى هندرب بيها المودل و ايه هي مجموعة الأعمدة او الخصائص اللى هيتم عليهم الاختبار و ايه هي الطريقة او الميثود اللى هيتم اتخاذها في اختيار الخاصية



- أول حاجة بيدخل كل الداتا و بيسأل هل هم لهم نفس الكلاس ولا لا
- لو اه يبقى خلاص ,, اما لو لا فبيدور على عمود او خاصية تانية و بيتم اختيارها بمعايير معينة تبع
- Attribute Selection method
- و بعد كده بيشوف فيه كام مسار هيمشى فيه تبع الخاصية دى و على قدرهم هيقسمهم
- Partitions
- و كل جزء من دول هيعمل فيه نفس اللى حصل لغيت اما كل الداتا تكون ليها كلاس



- The splitting criterion can cause one of 3 partitioning Scenario
  - Discrete eg..{Low , High , Middle}
  - Continuous -> in this case we use the logical operation eg {> , < , >= }
  - Binary Discrete {Yes or no}

Splitting Criterion is a test

- Which attribute to test at node N →What is the “best” way to partition D into mutually exclusive classes
- ايه افضل عمود استخدمه انه يفصل البيانات لمجوعات منفصلة غير مترابطة
- which (and how many) branches to grow from node N to represent the test outcomes
- Resulting partitions at each branch should be as “**pure**” as possible
- A partition is “**pure**” if all its tuples belong to the same class
- When attribute is chosen to split training data set, it’s removed from attribute list

Terminating conditions

- 3 حالات بتظهر بعد عملية التصنيف
- All the tuples in D (represented at node N) belong to the same class
  - كل الداتا تاخد نفس الكلاس اللى تبع النقطة يبقى ما فيش تقسيم و انتهى
- There are no remaining attributes on which the tuples may be further partitioned
  - لو خلص مجموعة الخصائص اللى انت بتصنف من خلالهم بتصنفهم تبع الاكثر الأكثر شيوعا
  - majority voting is employed → convert node into a leaf and label it with the most common class in data partition
- There are no tuples for a given branch

  a leaf is created with the majority class in data partition

Attribute selection measure

a **heuristic** for selecting the splitting criterion that “**best**” splits a given data partition into smaller mutually exclusive classes

- دى طرق عشان نعرف بيها اى خاصية او عمود هنختاره بالترتيب عشان نعمل التصنيف في اقل عدد من الكلاس الغير متداخلين
- Attributes are ranked according to a measure
  - attribute having the best score is chosen as the splitting attribute
  - split-point for continuous attributes
  - splitting subset for discrete attributes with binary trees
- Measures: **Information Gain, Gain Ratio, Gini Index**

Information Gain

Based on **Shannon’s information theory** ′Goal is to minimize the expected number of tests needed to classify a tuple

- guarantee that a simple tree is found ′Attribute with the highest information gain is chosen as the splitting attribute
- ال information gain بتحاول ان هي تلاقى ابسط tree ممكنة عن طريق تقليل كمية المعلومات اللى بنحتاجها عشان ن classify
- minimizes information needed to classify tuples in resulting partitions
- reflects least “**impurity**” in resulting partitions
- Given m class labels (Ci , i =1 to m)
- Expected Information needed to classify a tuple in D
- Info (D)= entropy = − σi=1 m pi log2(pi)
- pi → probability that an arbitrary tuple in D belong to class Ci

$$pi = \frac{|Ci\ D|}{|D|}$$

- Ci, D → set of tuples having class label Ci in partition D

Shanon Theory ?

شانون بدأ بقيس المعلومة من خلال ال Entropy والاحتمالات

$$Entropy = H = -\sum P_i \log P_i$$

أما ال Expected Info اللى محتاجها عشان تقدر تصنف ال tuple من الداتا وانت بتعمل Partitioning لل A بواسطة Attribute (A)

$$info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times info(D_j)$$

وبالتالى كل ما كان ال Expected info قليلة ~~يبقى~~ كل ما كان ال Classes بتكون أكتر independent وبالتالى هيديلك أفضل Purity

يبقى $Gain(A) = info(D) - info_A(D)$

وانت عايز تحافظ على كمية المعلومة اللى فى الداتا كلها Sele يبقى بتدور على أقل قيمة ممكنة لـ Expected $info_A$ ⇐ $info_A(D)$

department	age	salary	status	count
sales	Middle aged	medium	senior	30
sales	youth	low	junior	30
sales	Middle aged	low	junior	40
systems	youth	medium	junior	20
systems	Middle aged	high	senior	20
systems	senior	high	senior	10
marketing	senior	medium	senior	10
marketing	Middle aged	medium	junior	20
secretary	senior	medium	senior	10
secretary	youth	low	junior	10

ال Status هو ال~~Attribute~~ الفاصيح اللى عايزين نصنفها ع Class labels

أول حاجة محتاجين نجيب ال Expected Info و ال Entropy

C_1 (Senior) = 80 , C_2 (Junior) = 120

$$Info(D) = E = H = -\sum p_i \log p_i$$

$$= -\left(\frac{80}{200} \log \frac{80}{200} + \frac{120}{200} \cdot \log \frac{120}{200}\right)$$

$$= 0,97$$

كده محتاجين نجيب ال Expected info لكل Attribute من A_List

$\boxed{\text{DePartmenT}}$ :: Sales = 100 , system = 50
marketing = 30 , secretary = 20

$$Info_{dept} = \sum_{i=1} \frac{|D_i|}{|D|} \times Info(D_i)$$

$$= \frac{100}{200} \times -\left(\frac{\overset{\text{seniors}}{30}}{100} \log \frac{30}{100} + \frac{70}{100} \times \log \frac{70}{100} \right) +$$

$$\frac{50}{200} \times -\left(\frac{30}{50} \log \frac{30}{50} + \frac{20}{50} \times \log \frac{20}{50} \right) +$$

$$\frac{30}{200} \times -\left(\frac{10}{30} \log \frac{10}{30} + \frac{20}{30} \times \log \frac{20}{30} \right) +$$

$$\frac{20}{200} \times -\left(2 \times \frac{10}{20} \log \frac{10}{20} \right) = 0,92$$

Salary , age وهنجيب كمان بنفس الطريقة ال

$Info_{age} = 0,55$ $Info_{salary} = 0,95$

ونحسب ال Gain ال ③

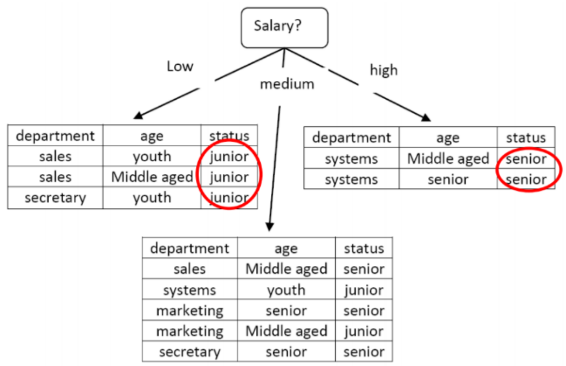
Gain (department) = 0,97 - 0,92 = 0,05

Gain (age) = 0,97 - 0,55 = 0,42

Gain (Salary) = 0,97 - 0,95 = 0,52

كده أخدنا ال Splitting Criterion عن طريق ال Salary

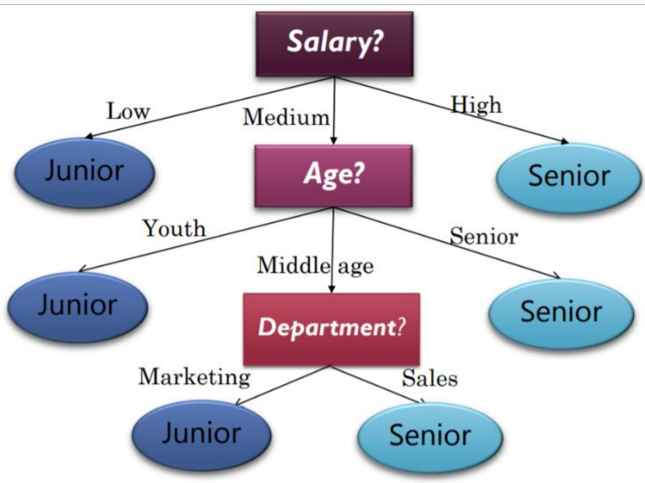
وبعد كده age ولو فاضل يبقى ال department



department	age	status
sales	youth	junior
sales	Middle aged	junior
secretary	youth	junior

department	age	status
systems	Middle aged	senior
systems	senior	senior

department	age	status
sales	Middle aged	senior
systems	youth	junior
marketing	senior	senior
marketing	Middle aged	junior
secretary	senior	senior

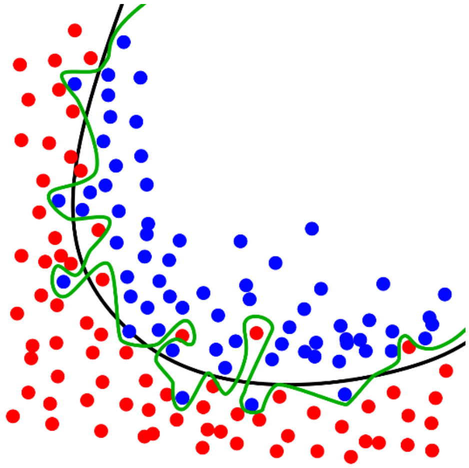


- أول حاجة اما نبدأ Classification نحدد أولا ال salary بتكون descrete و لها 3 Partitions
- نبدأ نقسم الداتا على ال Classes دى هنلاقى ان مازال ال Medium partition مالهوش pure class
- نبدأ ناخد تانى criterion عشان ن classify من خلاله
- و هكذا لغيت أما الاقى ال Termination point لكل الداتا
- لو انت عندك continuous Attribute بنحاول نجيب ال midpoint عشان نحوله شبه ال descrete و بنسمى النقطة دى ال Split Point

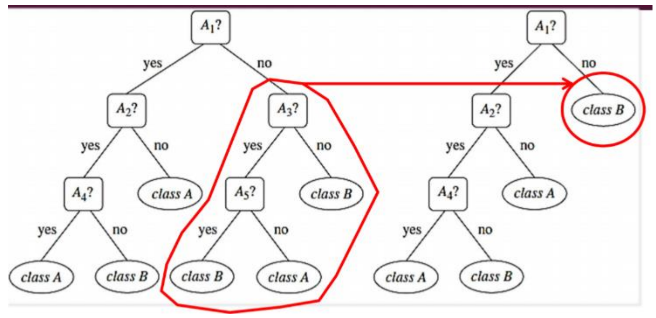
Information gain for **continuous attributes**

1. Sort values in <u>increasing</u> order
2. Each *midpoint* between two adjacent values can serve as *split-point*
3. Split-point between two values v_i *and* $v_{i+1} = \frac{v_i+v_{i+1}}{2}$
4. For each split-point, evaluate $info_A(D)$ with the number of partitions = 2 ($A \leq split\text{-}point$ & $A > split\text{-}point$)

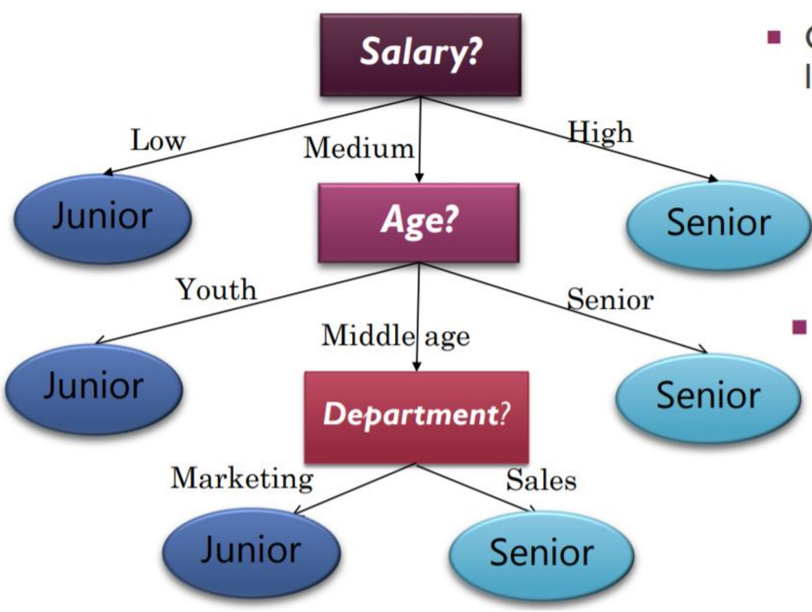
Tree Pruning



- تخيل ان عندك 2 class labels {Red & Blue}
- و لقيت ان وانت بت classify data ان معظما ال partition هيطلع أغلب الداتا ب Class {Blue} و احتمالية بسيطة جدا ان هو يبقى الكلاس التانى و المشكلة دى اسمها ال overfitting ,, فبدل ما أرهق الوقت و المجهود هقول ان ال partition ده الكلاس بتاعه Blue و هعمل pruning لل branches الزيادة دى
- زى الشكل اللى في الجنب ال overfitting هو انى اخلى ال tree بتاعتى تكون بدقة الخط الأخضر بس هيستغرق وقت و مجهود و منكم يخسر ,, فلو عملنا pruning هيبقى منطقى زى الخط الاسود



- Data may be overfitted to dataset anomalies and outliers
- Pruning removes the least reliable branches
- DT becomes less complex '
- Prepruning → statistically assess the goodness of a split before it takes place
  - hard to choose thresholds for statistical significance
- Postpruning → remove sub-trees from already constructed trees
  - remove sub-tree branches and replace with leaf node
  - leaf is labeled with most frequent class in sub-tree



- Create one rule for each path from root to leaf in the decision tree
  1. Each splitting criterion is ANDed to form rule antecedent (IF)
  2. Leaf node holds class prediction (THEN)
- R1: IF salary =medium AND age = youth THEN Status = Junior

Can the rules resulting from decision trees have conflicts?

RULE EXTRACTION FROM A DECISION TREE

department	age	salary	status	count
sales	Middle aged	medium	senior	30
sales	youth	low	junior	30
sales	Middle aged	low	junior	40
systems	youth	medium	junior	20
systems	Middle aged	high	senior	20
systems	senior	high	senior	10
marketing	senior	medium	senior	10
marketing	Middle aged	medium	junior	20
secretary	senior	medium	senior	10
secretary	youth	low	junior	10

Data item Belong To medium Partition

R1 = If Salary = medium AND age = youth
Then Status = Junior

and youth

Class label

Rule

لحاجة ال rule بتغطي أد إيه

$$\text{Coverage}(R) = \frac{n_{\text{Covers}}}{|D|} = \frac{20}{200} = 10\%$$

$$\text{accuracy}(R) = \frac{n_{\text{Correct}}}{n_{\text{Covers}}} = \frac{20}{20} = 100\%$$

Prediction Rule

لو جالي emp جديد بالشكل ده

X : (Department = System , age = youth , Salary = medium)

Prediction of A Tree for X = Junior

أحيانا بيحصل مشكلة وإحنا بنبني الـ Tree وهي Rule Conflict

إن يكون فيه Tuple بيحقق أكثر من قاعدة داخل الـ Tree

الحل

2 Resolution Strategies

Size ordering

معتمد على حجم الـ Tuples اللي موجودة

تبع الـ Rules كل ما كان أكبر حجم

يختار الـ Rule ~~هو~~

Rule ordering → Priority for Apriori

→ Class based ordering → decreasing importance

~~أول طريقة~~ أول طريقة ترتيب عن طريق الـ Classes كل

ما كان الـ Class ظهوره أكبر وإحتماله بيبقى أختاره الأول

→ Rule - based ordering

وهنا بيحسب الـ Quality لكل Rule اللي أعلى accuracy بيختار

Fall back Rule → بيحط الـ default زي الـ Switch

لو مالقاش Rule بيستخدمه

Naïve Bayesian Classifier

هو method لـ Classify عن طريق Statistics عن طريق

تنبؤ احتمالية إن الـ Tuple ينتمي إلى الكلاس المعين

High accuracy => دقيقة جداً
and Speed وسريعة

مبني على Bayes Theory مبني على مبدأ { Class_Conditional Independence }

وإن الـ Attributes بتأثر إزاي في تصنيف الـ Tuple

n = # Attributes m = # Classes

Naïve Bayes بيتنبأ إن الـ X Tuple بينتمي لـ Class اللي هو أعلى احتمالية

$$P(C_i|X) > P(C_j|X) \text{ for } 1 \leq j \leq m, j \neq i$$

Conditional Probability

C_i => Maximum Posteriori Hypothesis

$$P(C_i|X) = \frac{P(X|C_i)\,P(C_i)}{P(X)}$$

الـ $P(X)$ ثابت يبقى احنا محتاجين نـ maximize البسط (numerator)

وتخيل إن احنا متخيلين على Uniform Probability يبقى كمان

$P(C_i)$ هتبقى موحدة لكل قيم i

كده انت مش محتاج الا انك ت maximize قيمة $P(X|C_i)$

لو مش uniform ⇐ $P(C_i) = \frac{|C_{i,D}|}{|D|}$

وعشان ت Reduce Compute ل $P(X|C_i)$ هنفرض ان

ال Attributes مع Independent

$$P(X|C_i) = \overset{\boxed{\text{for All}}}{\prod_{K=1}^{n}} P(x_K|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \text{-----} \times P(x_n|C_i)$$

لو ال Attribute نوعه nominal ⇐ Categorical

$$P(x_K|C_i) = \frac{|C_{i,D}|}{|C_{i,D}|}$$

اما لو كان ⇐ numerical ⇐ Gaussian Distribution

$$P(x_K|C_i) = \frac{1}{\sqrt{2\pi}\,\sigma_{C_i}} e^{-\frac{(x_K - \mu_{C_i})^2}{2\sigma_{C_i}}}$$

department	age	salary	status	count
sales	Middle aged	medium	senior	30
sales	youth	low	junior	30
sales	Middle aged	low	junior	40
systems	youth	medium	junior	20
systems	Middle aged	high	senior	20
systems	senior	high	senior	10
marketing	senior	medium	senior	10
marketing	Middle aged	medium	junior	20
secretary	senior	medium	senior	10
secretary	youth	low	junior	10

ناخد مثلاً object ⇐ X : { department = "marketing" ; age = youth ; salary = "low" }

وفيه 2 class label ⇐

يبقى محتاجين نجيب

$$① \quad P(X \mid \overset{\text{Senior}}{C_1}) = \frac{10}{80} \times \frac{0}{80} \times \frac{0}{80} = 0 \rightarrow$$

$$\therefore \quad P(X \mid C_1)\, P(C_1) = 0$$

$$② \quad P(X \mid \overset{\text{Junior}}{C_2}) = \frac{20}{120} \times \frac{60}{120} \times \frac{80}{120} = 0{,}055$$

$$\therefore \quad P(X \mid C_2)\, P(C_2) = 0{,}055 \times \frac{120}{200}$$

$$= 0{,}033$$

بيبقى ال X object بينتمي لكلاس Junior

بس فيه مشكلة إن لو فيه Attribute ماحققتش يعني احتمال

ظهور ال X مابيعتمدش عليه كده هيخلي كل المعادلة (صفر)

وعشان كده بنستخدم Laplacian estimator ⇐ Correction

إن بنعوض ~~بقيمة~~ بقيمة (1) مكان الأصفار

$$P(X \mid C_1) = \frac{10}{80} * \frac{1}{80} * \frac{1}{80} = 0{,}0002$$

$$\therefore P(X \mid C_1)\, P(C_1) = 0{,}00008$$

ومن كل الحالتين $X class = Junior$

Lazy Learners

هم مجموعة Algorithms بتشتغل أو بتعمل Classify للـ Test data

بس مابتبنيش models أو Pattern على الـ Learning Data الا لما تيجي

داتا جديدة عايزة يحصل لها Classify

من الالجوريزمات دي K-NEArest Neighbor Classifiers

بتستخدم الـ (Similarity measure) عشان تحسب التباعد مابين

الـ Test Data والداتا الـ (Training)

- ملحوظة إذا كانت الـ Attributes مختلفة في الـ ranges

اعمل Normalization ⇐ Preprocessing

K هم عدد الـ neighbours objects الأقرب للـ Test object من خلال

الـ distance

و بنعمل عملية الـ (majority Voting) عشان نعرف

الـ Class للـ Test Data

أولاً حاجة هنحسب ال distance

عن طريق Euclidean ⇐

$$D = \sqrt{(x_1-y_1)^2 + (x_2-y_2)^2}$$

تمام هاخد مثال

for X => Test Data
and Y => RID(1)

$$D_{X,Y} = \sqrt{(48-25)^2 + (142000-40000)^2}$$

$$= \boxed{102,000}$$

وهكذا

بس هنلاقي مشكلة إن ال Loans هيأثر بحجمها ده

ولا علاجها ال Distance هنعملها Normalization عن طريق

RID	age	Loan ($)	Default	Distance
1	25	40000	No	102000
2	35	60000	No	82000
3	45	80000	No	62000
4	20	20000	No	122000
5	35	120000	No	22000
6	52	18000	No	124000
7	23	95000	Yes	47000
8	40	62000	Yes	80000
9	60	100000	Yes	42000
10	48	220000	Yes	78000
11	33	150000	Yes	8000
	48	142000	?	fair

Preprocessing (Transform) ⇐ [(Min_max) or Z-Score]

Loan min = 18000 Loan max = 220000

$$Loan_{RID1} = \frac{40000 - 18000}{220000 - 18000} \times (60 - 20) + 20$$

(Scaler)

وهكذا

وكذلك ال Distance

نحسبها تاني بعد ال Norm نفسها

new_min = 20 new_max = 60

عشان نخليها في نفس ال Ranges ما بين

ال age , ~~Distance~~ Loans

Loan ($)	Distance
24.4	30.6
28.3	20.8
32.3	12.7
20.4	37.0
40.2	13.7
20.0	24.9
35.2	26.7
28.7	17.8
36.2	14.6
60.0	15.4
46.1	15.1
44.6	

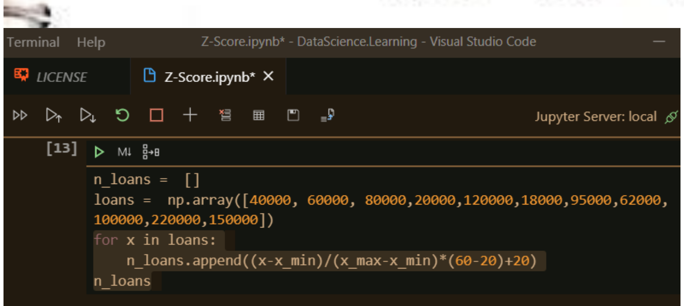
نحسب مين أقرب ال object Test
جيرانه

(No) ⇐ K = 1 NN is RID 3 أقرب لها

K = 3 NN is RID (3 = No , 5 = No , 9 = Yes)

majority?

Default = No



```
[13]
n_loans = []
loans = np.array([40000, 60000, 80000,20000,120000,18000,95000,62000,
100000,220000,150000])
for x in loans:
    n_loans.append((x-x_min)/(x_max-x_min)*(60-20)+20)
n_loans
```

Linear Regression

من أحد الطرق اللي نعمل بيها Prediction

و بحاول نعمل علاقة مابين ال 2 Variable

والنتيجة بتاعت ال Regression هي (Linear regression equation)

إني من خلالك Variable أقدر أعمل Prediction ل Variable التاني

$$Y = a + bX$$

بعمل إني أكون (Scatter Plot) وبشوف أو بختبر إذا

كان الداتا roughly fits a line لأن لو مش متحيزة

للخط يبقى ال regression ملهوش لزمة ←

2 Variable are Independent

$\cancel{Y = a\,X + b}$

$$Y = a + bX$$

$Y \Rightarrow$ dependent variable

$X \Rightarrow$ independent variable

$b \Rightarrow$ Slope $a \Rightarrow$ y-intercept

هسا أحسب (xy, x^2, y^2)

$$a = \frac{(\Sigma y)(\Sigma x^2) - (\Sigma x)(\Sigma xy)}{n(\Sigma x^2) - (\Sigma(x))^2}$$

$$b = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2}$$

SUBJECT	AGE X	GLUCOSE LEVEL Y	XY	X2	Y2
1	43	99	4257	1849	9801
2	21	65	1365	441	4225
3	25	79	1975	625	6241
4	42	75	3150	1764	5625
5	57	87	4959	3249	7569
6	59	81	4779	3481	6561
Σ	247	486	20485	11409	40022

بعد الحساب

$a = 65.1416$

$b = 0,385225$

$$Y = 65.1416 + 0,385225\,X$$

وحط أي قيمة لـ X بطلع قيمة Y

Metrics for Evaluating Classifier Performance

إزاي أقيس تقييمات (Performance) الـ Classifier [Performance]

Parameters

TP : [True Positives]

ودول عدد الـ objects اللي حصلهم تصنيف صح
وكانت لهم أهمية [Class of interest]

TN : [True Negatives]

دي مالهمش أهمية أو غرض في البحث صح

Positives [P] => لهم أهمية في البحث

Negatives [N] مالهمش أهمية

FP [False Positives]

هم مالهمش أهمية في البحث والـ Classifier أخطأ
في التصنيف

FN [False Negatives]

لهم أهمية في البحث بس الـ Classifier أخطأ في
التصنيف

Measures

accuracy = recognition rate

$$= \frac{TP + TN}{P + N} = \frac{\text{كل اللي إتصنف صح}}{\text{كل الداتا}}$$

error rate = misclassification rate

$$= \frac{FP + FN}{P + N} = \frac{\text{كل اللي إتصنف غلط}}{\text{كل الداتا}}$$

Sensitivity = True Positive rate

$$= \frac{TP}{P}$$

~~معدل التصنيف البيانات~~

كام عدد اللي كانوا صح من اللي هم إيجابي

specificity = true negative rate

$$= \frac{TN}{N}$$

Precision

$$= \frac{TP}{TP + FP}$$

- ال confusion matrix من أهم الاشكال الى بتخلينا نقيس ال performance لل classifirer
- الصفوف بتوضح ايه القيم الحقيقية اللى المفروض ال classifier لو هو مية في المية صحيح يطلعها
- الأعمدة بتوضح ايه القيم الفعلية اللى تنبأ
- بيها ال classifier

Actual \ Predicted	Yes	No	Total
Yes	TP	FN	P
No	FP	TN	N
Total	$\hat{P}$	$\hat{N}$	P + N

Confusion Matrix

Balanced Classes

Actual \ Predicted	Yes	No	Total	Recognition(%)
Yes	6954	46	7000	TP/P = 99.34
No	412	2588	3000	TN/N = 86.27
Total	7366	2634	10000	$\frac{TP+TN}{P+N}$ = 95.42

Example Buys_Computer Confusion Matrix

Use *sensitivity* (TPs or *recall*) and *specificity*

Imbalanced Classes

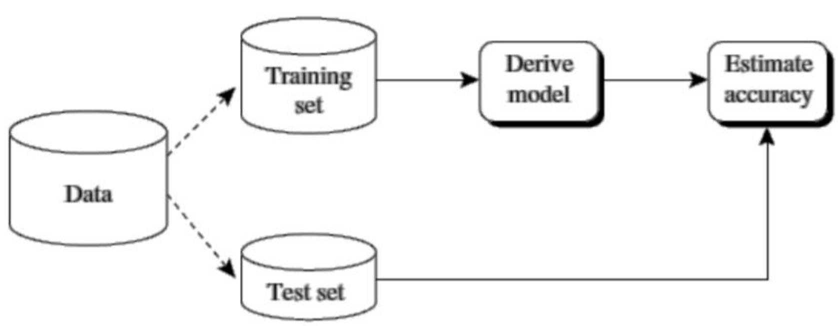
Actual \ Predicted	Yes	No	Total	Accuracy (%)
Yes	90	210	300	30
No	140	9560	9700	98.56
Total	230	9770	10000	96.4

Example Cancer Confusion Matrix

Use *sensitivity* (TPs or *recall*) and *specificity*

- صحيح ال Accuracyبتاعت ال cancer اعلى من ال computers بس بسبب التباين الجزرى اللى حصل لل Positives اللى من حيث sensitivity اللى مش كويسة لدى ال cancer أدت ان الاعتماد على ال classifier ده غلط

- **Holdout** → RANDOMLY allocate 2/3 of data for training and remaining 1/3 for testing
- **Random Subsampling** → Repeat holdout *k* times and take average accuracy

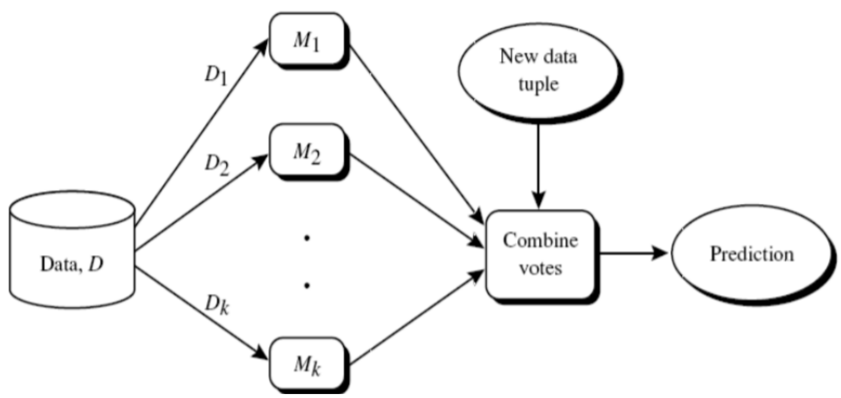


- من طريق تحسين عملية ال classification و هو اختيار ايه الداتا اللى هتعمل training و ايه اللى هيتعمل بيه testing ففي ال holdout بقسم البيانات ل 3 أثلاث ,, ثلثان لل training و ثلث لل testing
- ال random subsampling انى اختبر الأقسام دى بعشوائية عدد من المرات لغيت ما أتأكد انه Balanced

- **k-fold cross-validation** →randomly partition dataset into k mutually exclusive ***folds*** of approximately equal size
- In iteration i, $fold_i$ is test set and all other folds are training set
- Accuracy = $\frac{\sum correct\ classifications\ for\ all\ k\ iterations}{dataset\ size}$
- **Stratified k-fold cross-validation** → class distribution in each fold is approximately the same as in initial dataset
  - **Stratified 10-fold cross-validation** is recommended

- بعشوائية بحاول اقسم الداتا لمجموعة من ال folds اللى تقريبا قد بعض
- و على قد ال folds و احدة منهم بتكون test و الباقى training
- ال startified sampling ان في كل flod لازم يكون نسب ال classes بتساوى النسب الأصلية لل Whole dataset

- **Ensemble** → a set of classifiers, each with a vote for a class label
  - Each base classifier is produced from a different partition of the dataset
  - Majority voting is used to compose an **aggregate classification**



- في ال ensemple بشتغل بأكتر من classifier و أخليهم يشتغلوا على الداتا و اقارن بين النتائج بتاعتهم و أعمل aggregation لافضل نتيجة في ال accuracy لل classifier و اللى يطلع أفضل أختاره انه يبقى ال classifierبتاعى

Algorithm: Bagging. The bagging algorithm—create an ensemble of classification models for a learning scheme where each model gives an equally weighted prediction.

Input:

- D, a set of d training tuples;
- k, the number of models in the ensemble;
- a classification learning scheme (decision tree algorithm, naïve Bayesian, etc.).

Output: The ensemble—a composite model, $M*$.

Method:

(1) **for** $i = 1$ **to** k **do** // create k models:
(2) create bootstrap sample, D_i, by sampling D with replacement;
(3) use D_i and the learning scheme to derive a model, M_i;
(4) **endfor**

To use the ensemble to classify a tuple, X:

let each of the k models classify X and return the majority vote;

Bootstrap
→ same size as dataset, sampling with replacement

3
5
3
7
4
3
7
9
6
10

- في ال Bagging بستخدم ال SRSWR Sampling بس Bootstrap يعنى من نفس المقاس لأنه ب replacement و بعملهم ensempling عادى و بفاضل بينهم