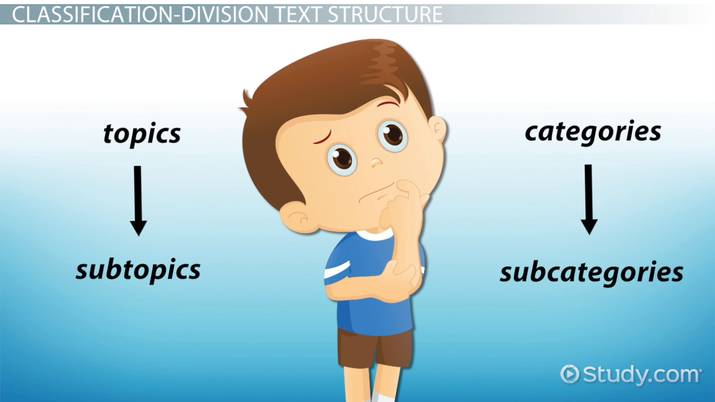
ي

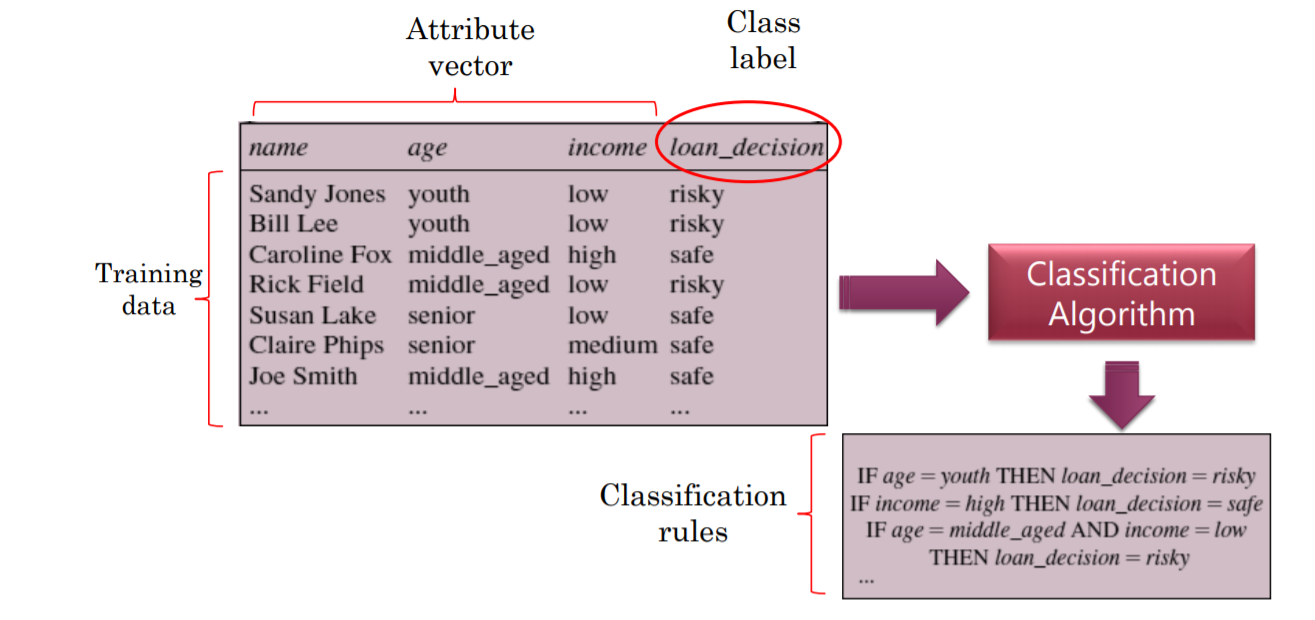
Classification

# What is The Classification

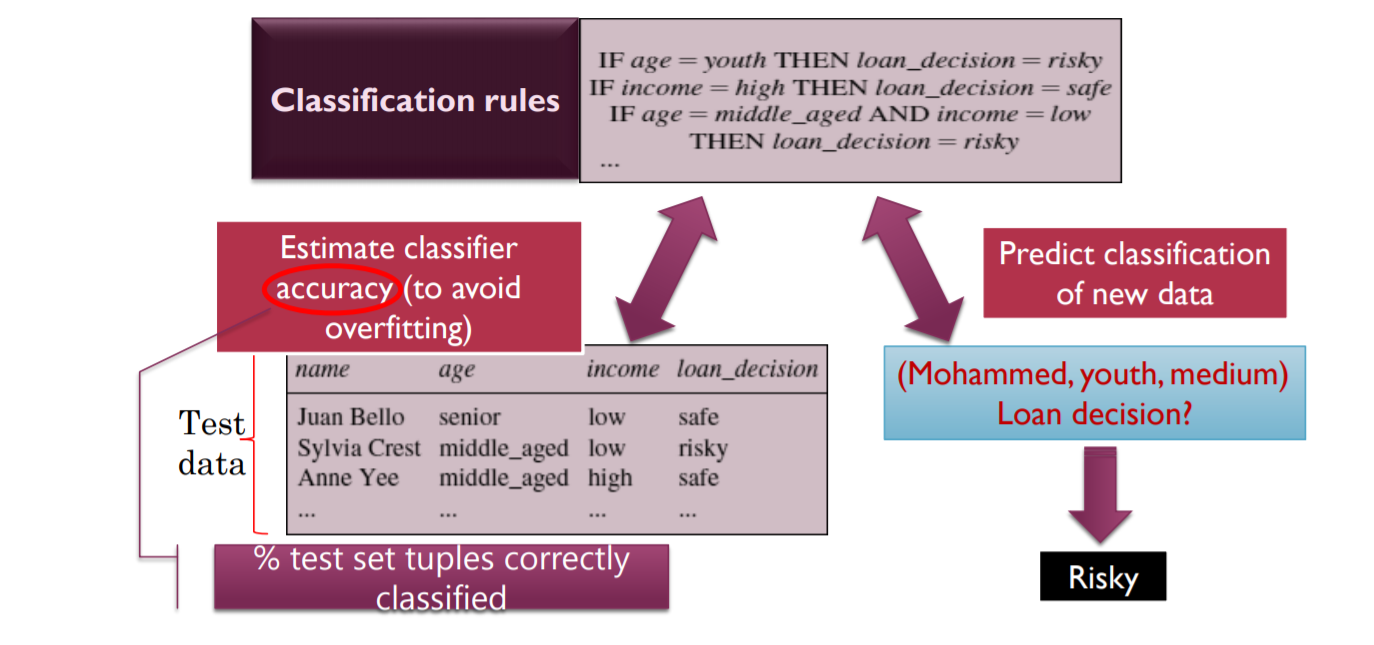
* Classification is a data analysis task where a model is constructed to predict class labels (categories)
* Motivation: Prediction
* المحرك الأساسى للتصنيف هو التنبؤ رغم الأغراض التانية زى التوصيف للبيانات
* **Descriptive vs Predictive Tasks** Chapter 1
* Is a bank loan applicant “safe” or “risky”?
* 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)



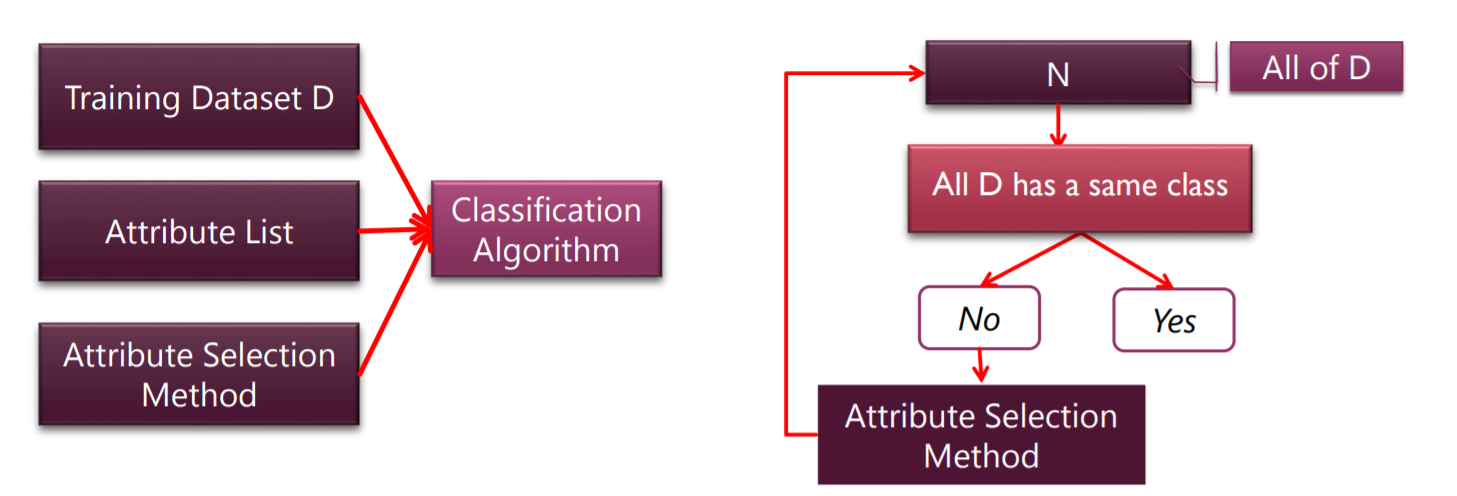
* بيحتاج داتا يتدرب عليها فيصنع مودل عبارة عن مجموعة من القواعد عن طريق الجوريزم معين
* Age and Income is the attributes that can judge about loan

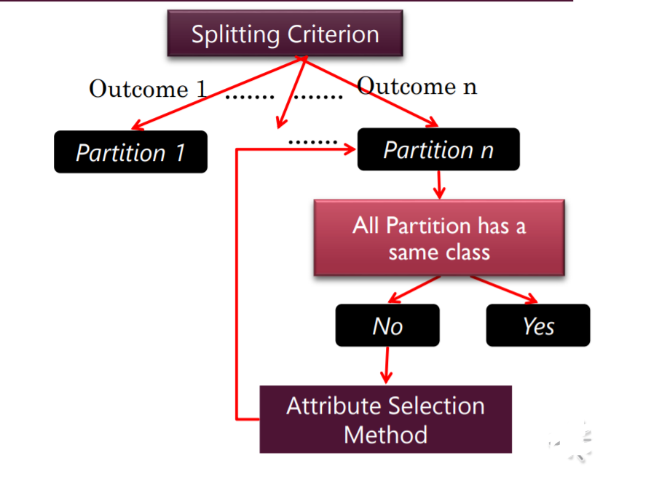


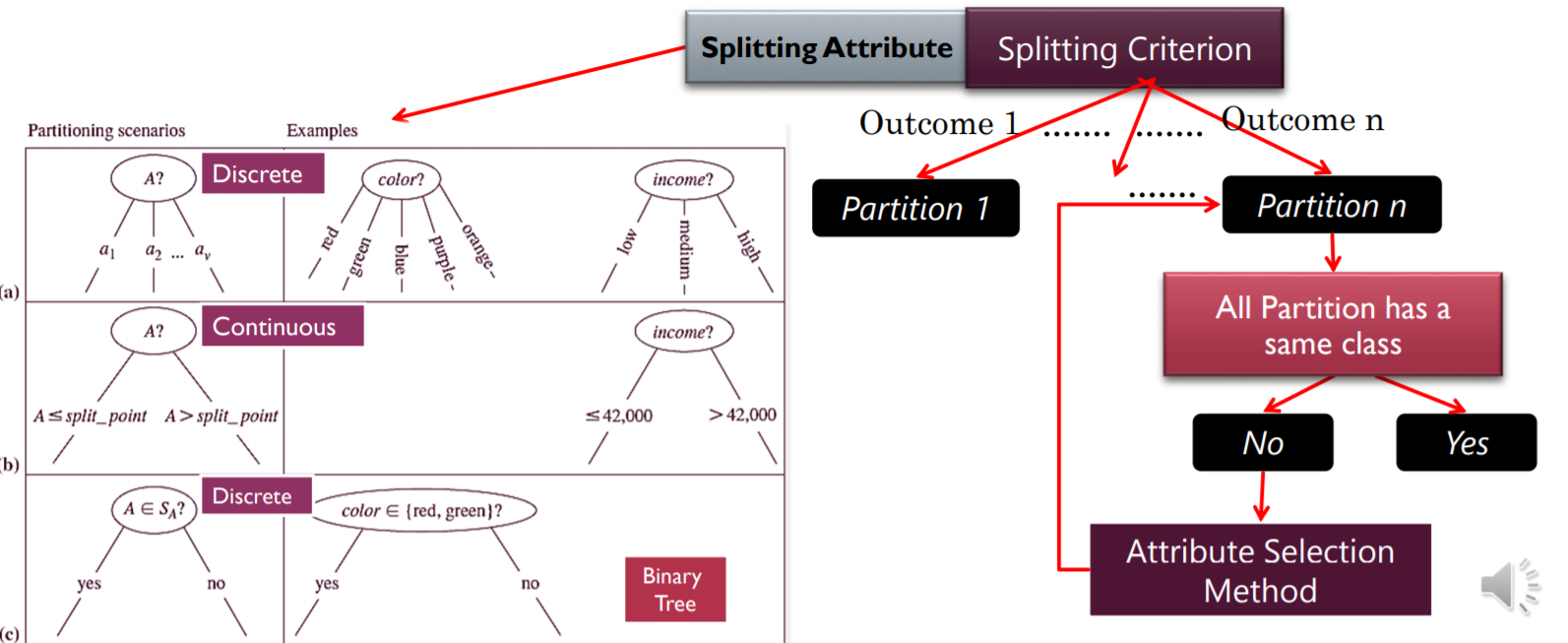
* بعد ما يطلع القواعد بنختبر مدى دقة المودل ده عن طريق انى بجيب داتا تانية بختبر عليها مدى صحة القواعد دى
* و ان كان مدى دقة القواعد سليمة كفاية ,, منكم اتنبأ لو فيه داتا من غير تصنيف اقدر اصنفها بالقواعد دى

# 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

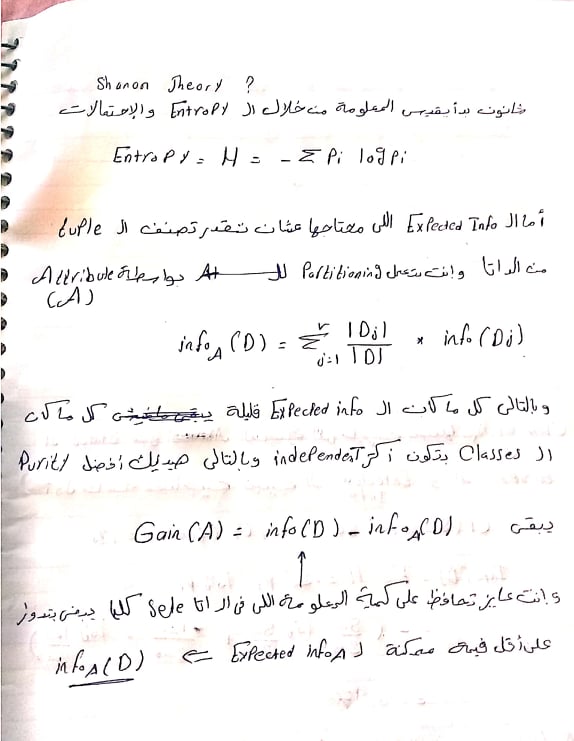
* دى طرق عشان نعرف بيها اى خاصية او عمود هنختاره بالترتيب عشان نعمل التصنيف في اقل عدد من الكلاس الغير متداخلين
* Attributes are ranked according to a measure
  + attribute having the best score is chosen as the splitting attribute
  + split-point for continuous attributes
  + splitting subset for discrete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

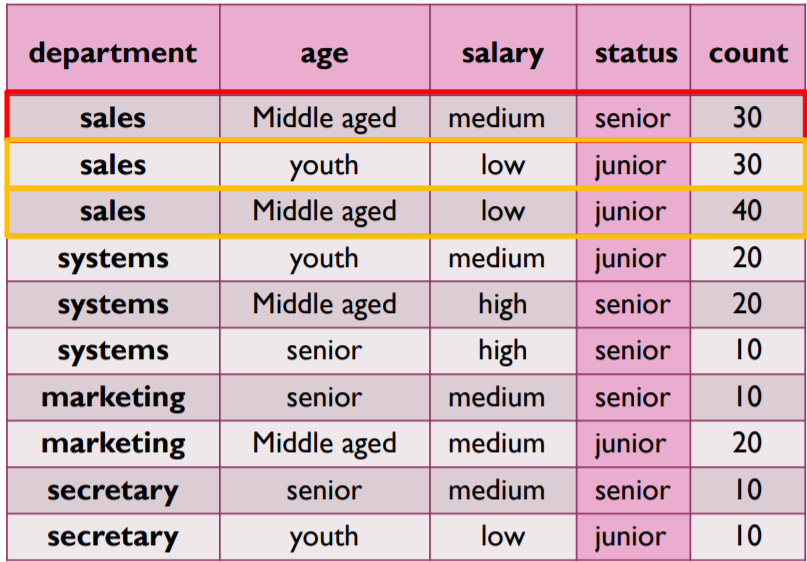
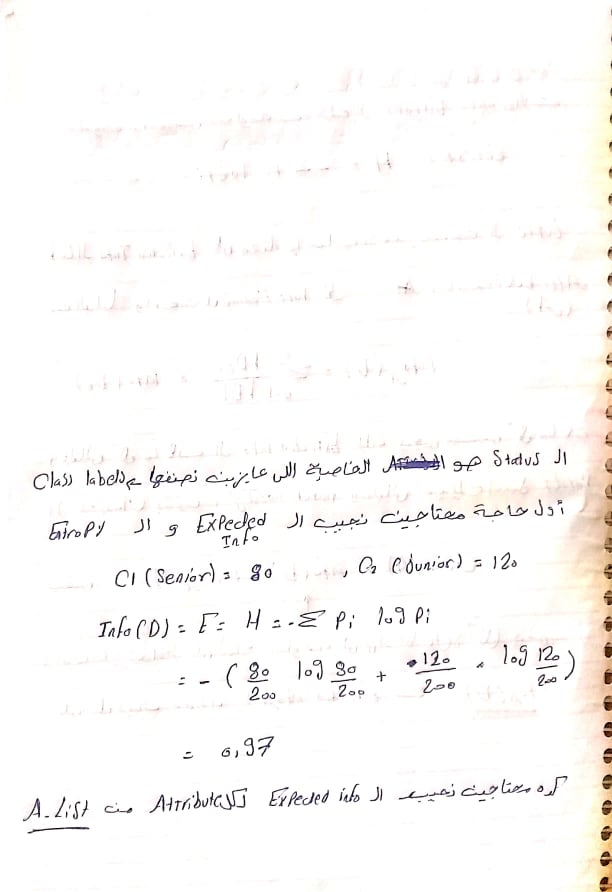
### Information Gain

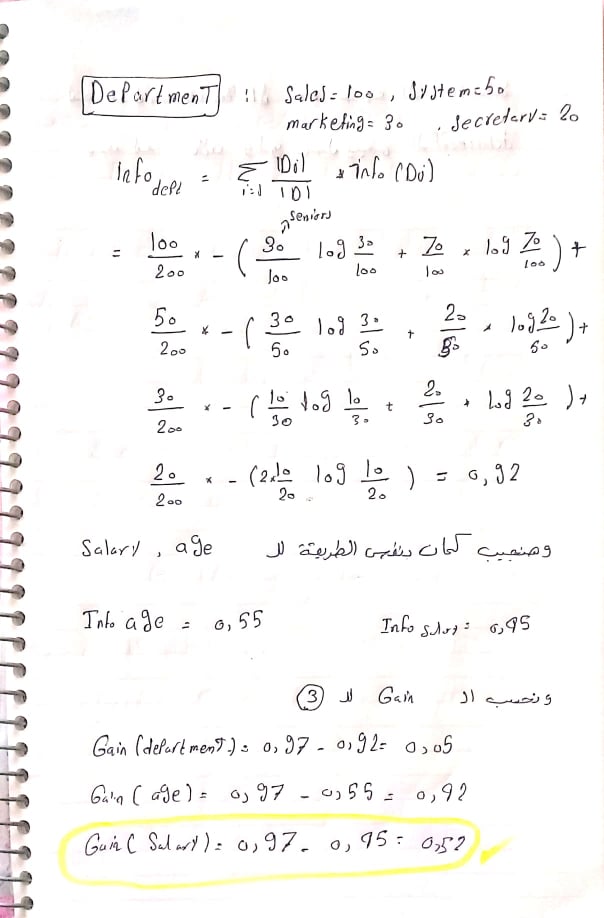
Based on **Shannon’s information theory** ′Goal is to minimize the expected number of tests needed to classify a tuple

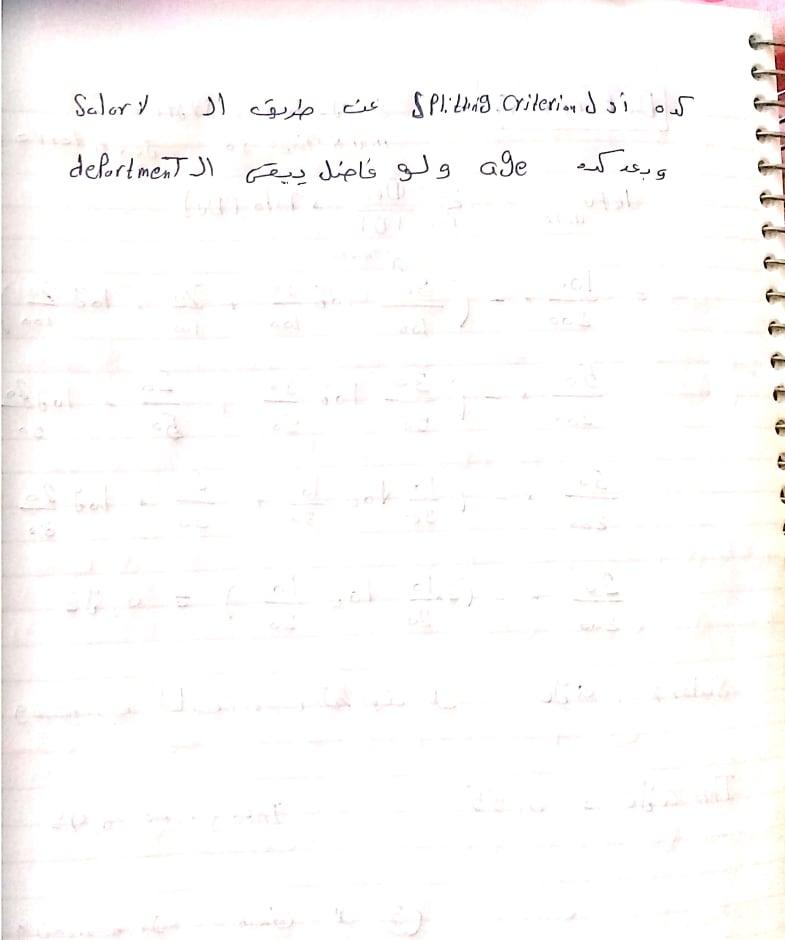
* guarantee that a simple tree is found ′Attribute with the highest information gain is chosen as the splitting attribute
* الinformation gain بتحاول ان هي تلاقى ابسط tree ممكنة عن طريق تقليل كمية المعلومات اللى بنحتاجها عشان ن classify
* minimizes information needed to classify tuples in resulting partitions
* 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

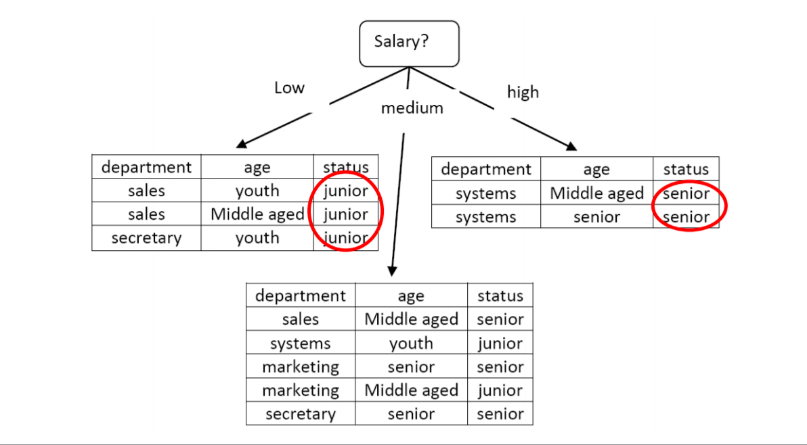
* Ci, D → set of tuples having class label Ci in partition D

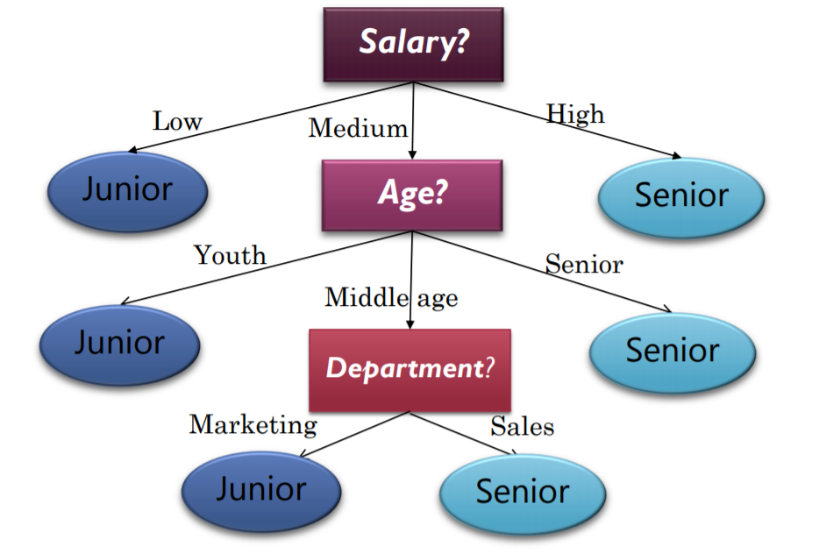
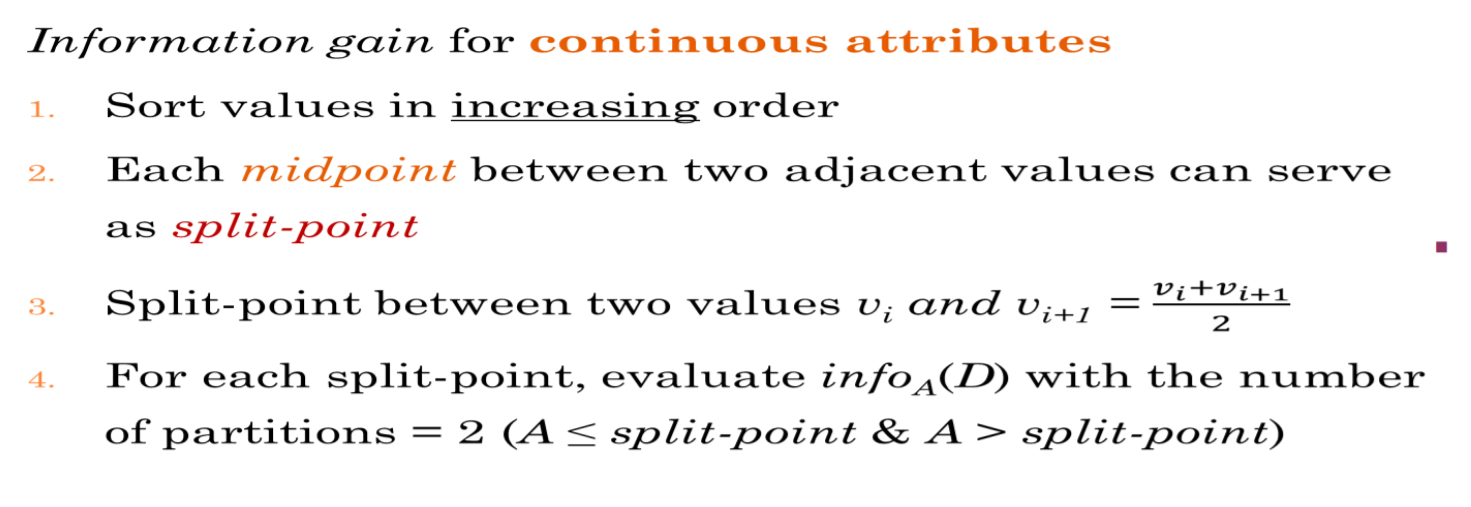


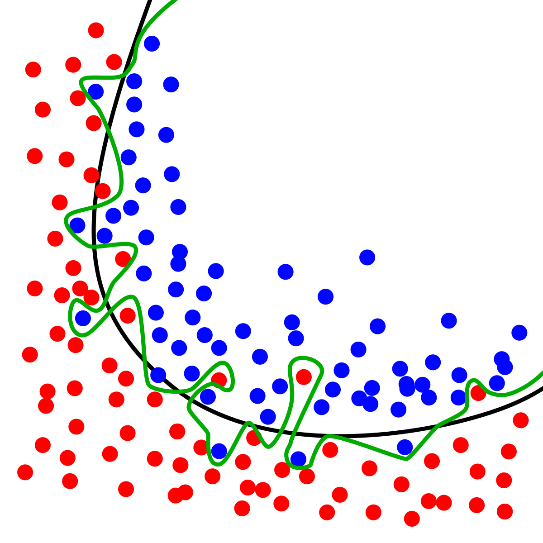
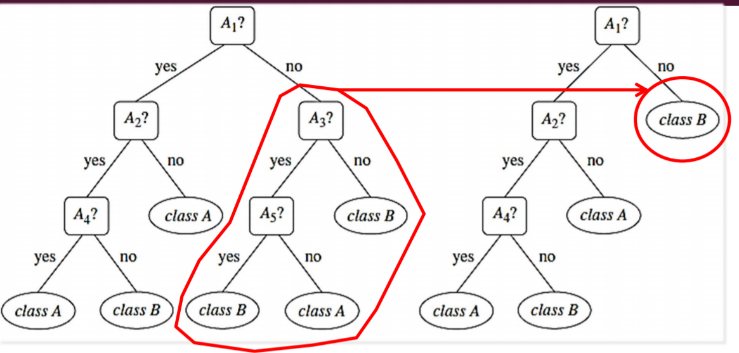


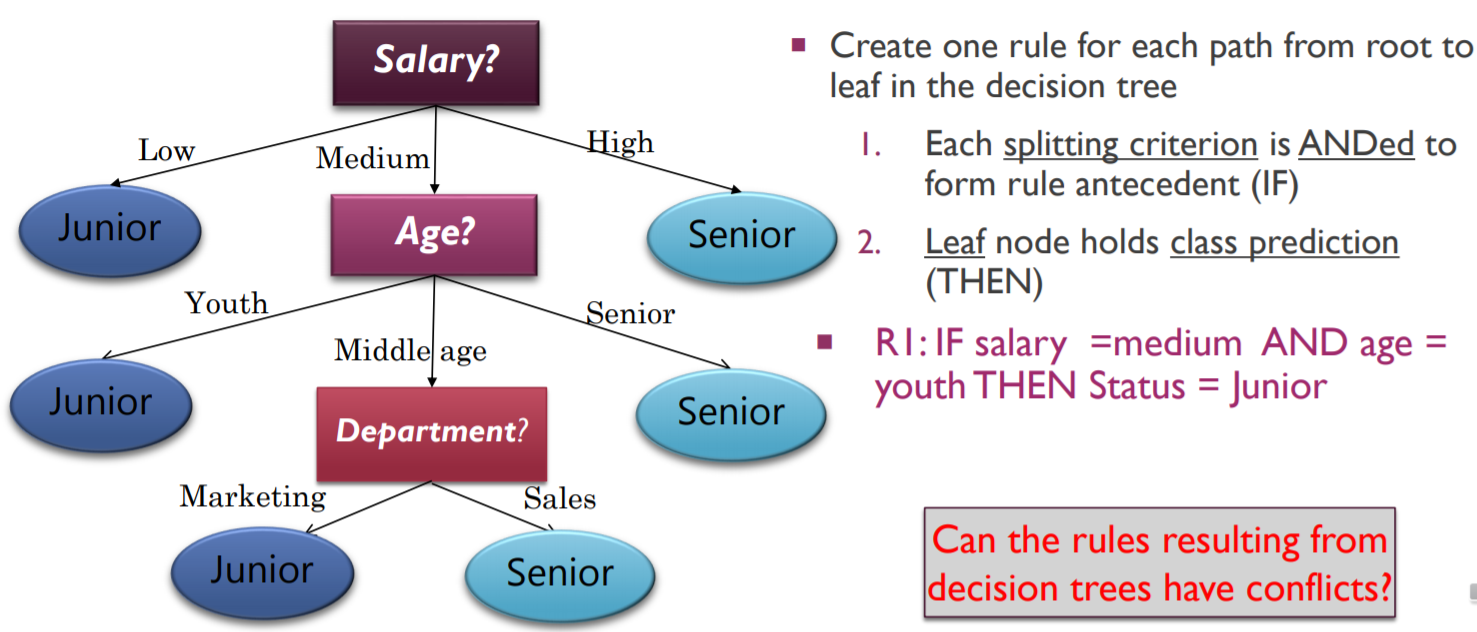




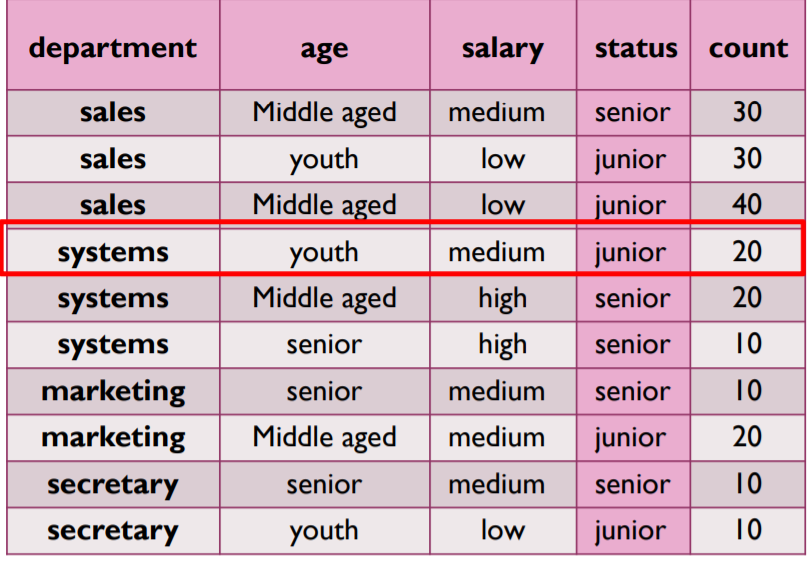
* أول حاجة اما نبدأ Classification نحدد أولا ال salary بتكون descrete و لها 3 Partitions
* نبدأ نقسم الداتا على ال Classes دى هنلاقى ان مازال ال partition Medium مالهوش pure class
* نبدأ ناخد تانى criterion عشان ن classify من خلاله
* و هكذا لغيت أما الاقى ال Termination point لكل الداتا
* لو انت عندك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

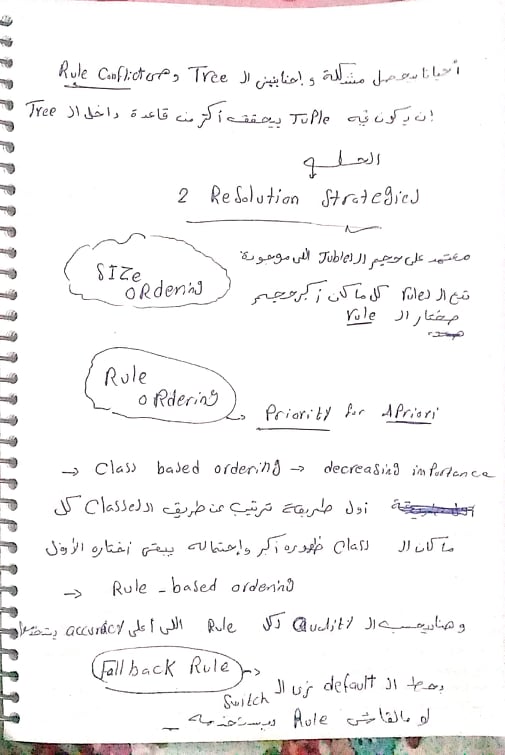
## Tree Pruning

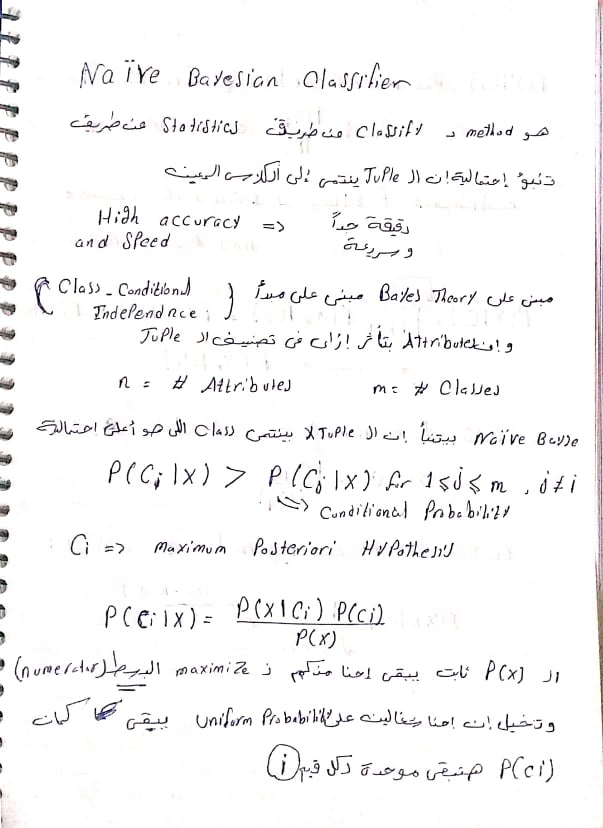
* تخيل ان عندك 2 class labels {Red & Blue}
* و لقيت ان وانت بت classify data ان معظما ال partition هيطلع أغلب الداتا ب Class {Blue} و احتمالية بسيطة جدا ان هو يبقى الكلاس التانى و المشكلة دى اسمها ال overfitting ,, فبدل ما أرهق الوقت و المجهود هقول ان ال partition ده الكلاس بتاعه Blue و هعمل pruning 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 منطقى هيبقى 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

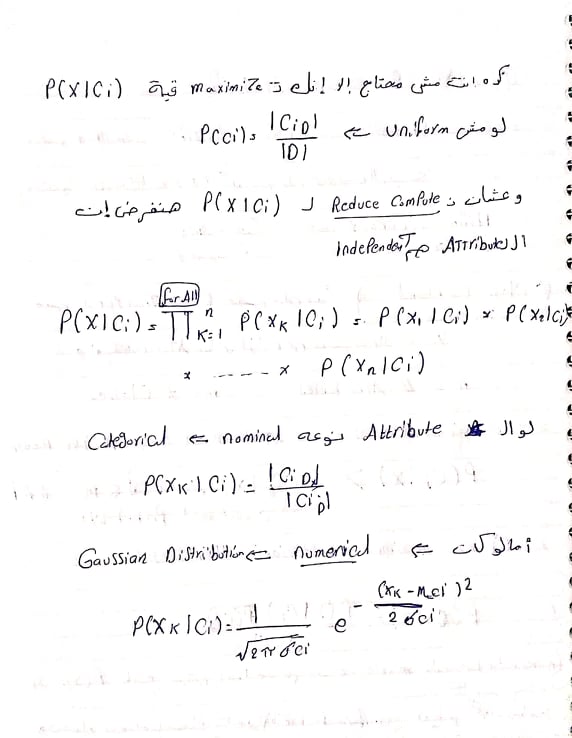


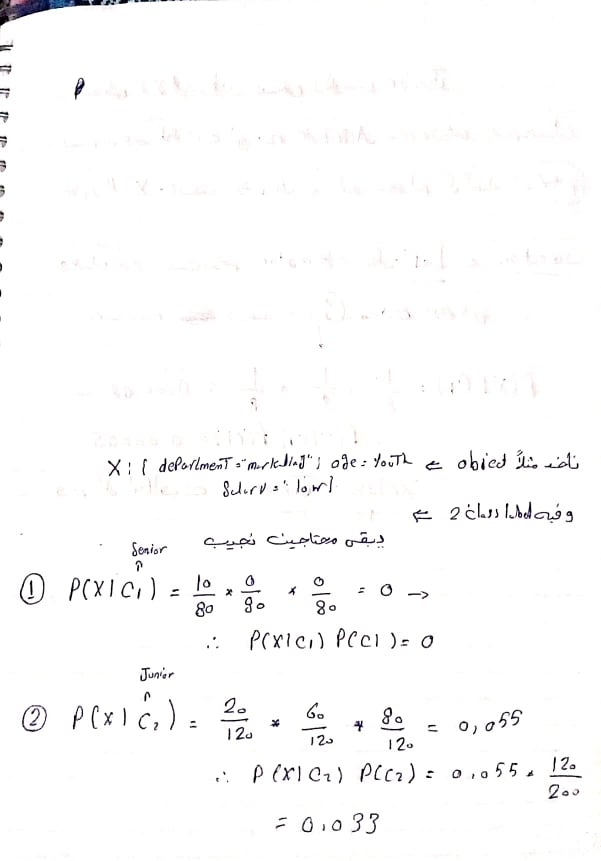
## RULE EXTRACTION FROM A DECISION TREE

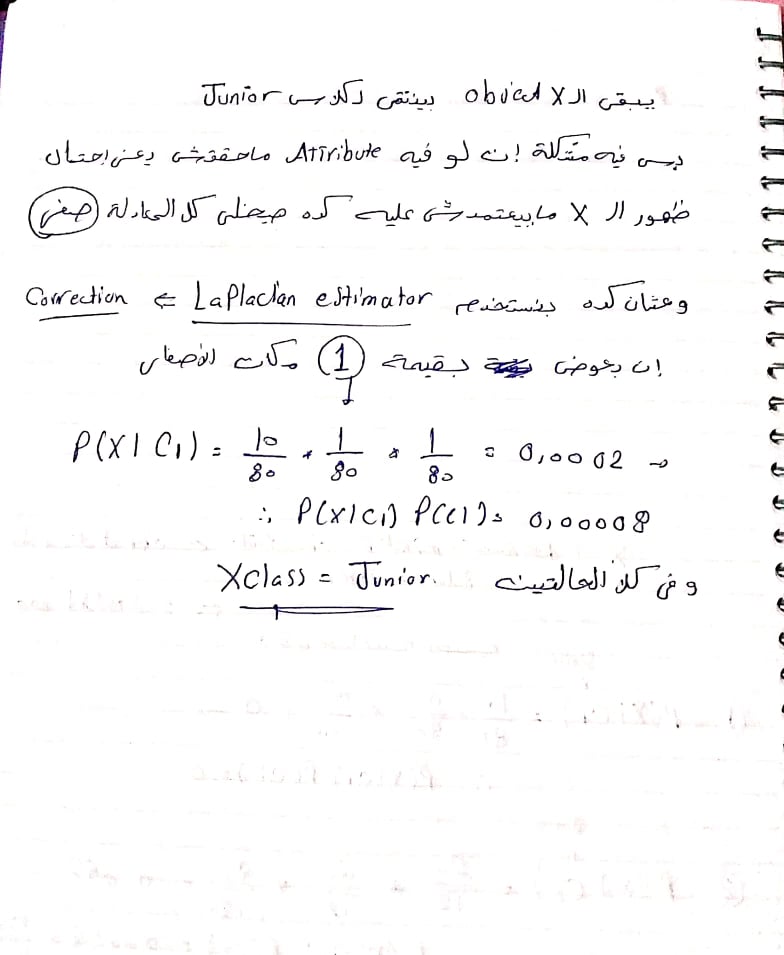


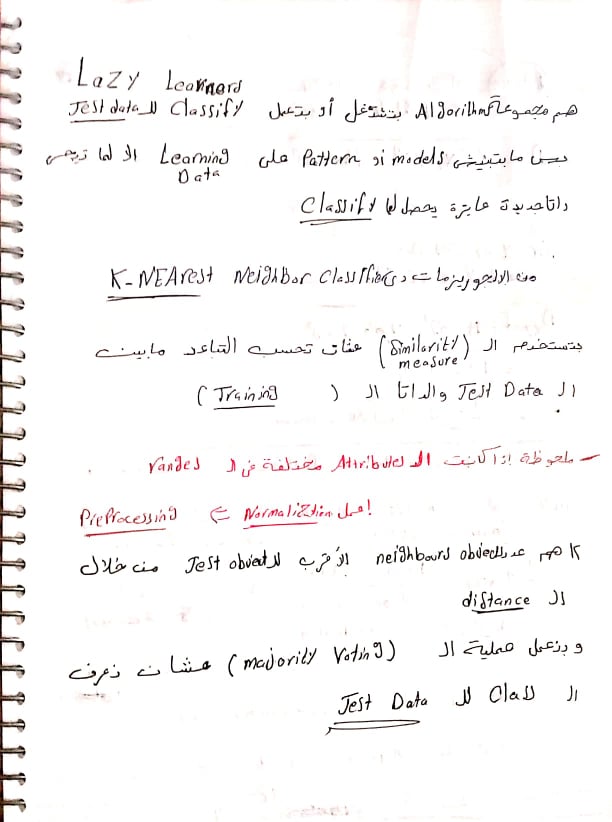


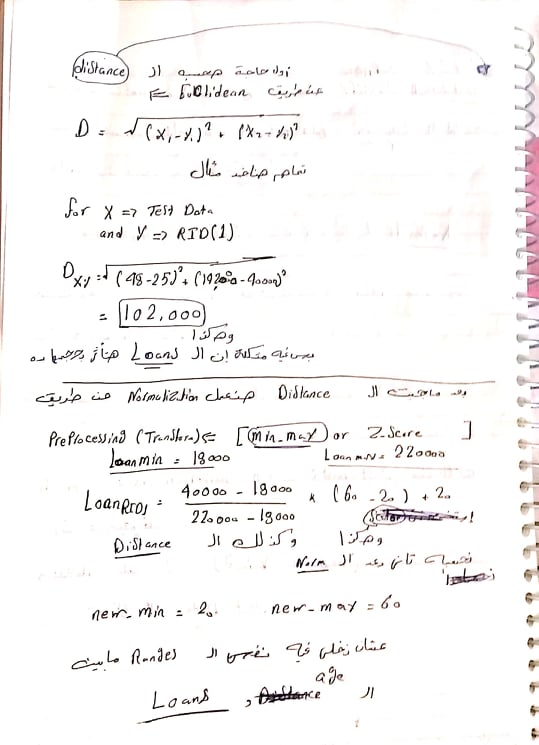
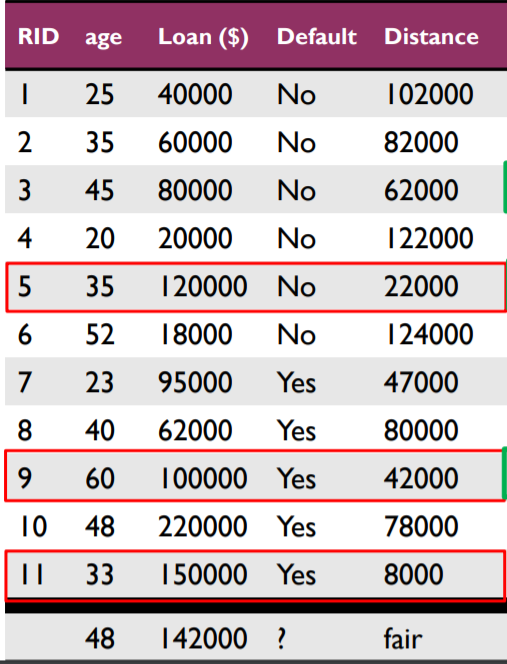


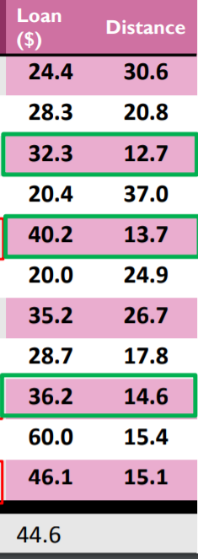


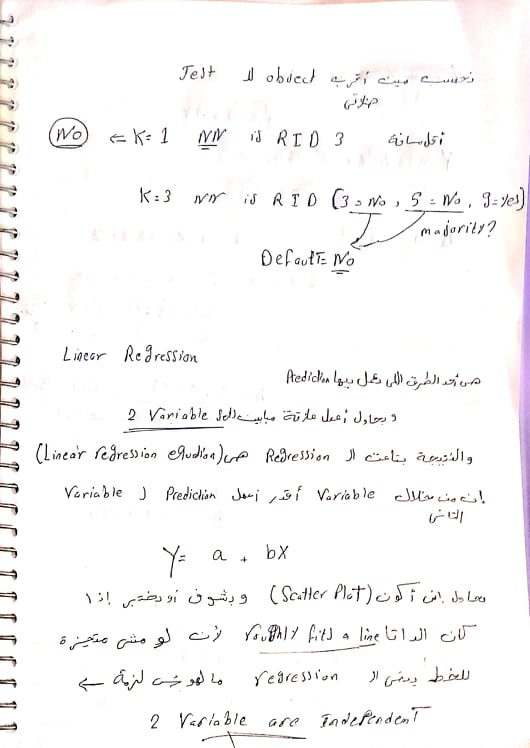
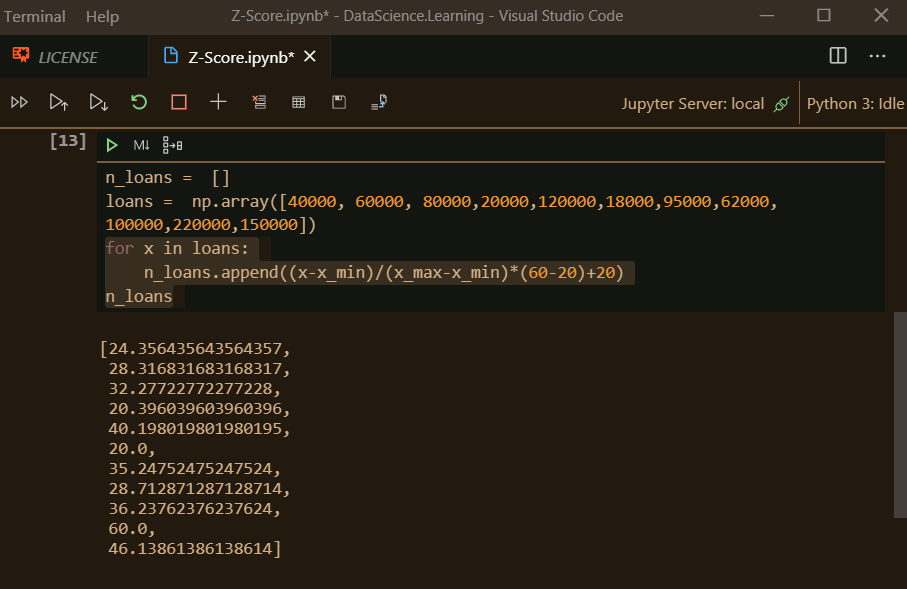


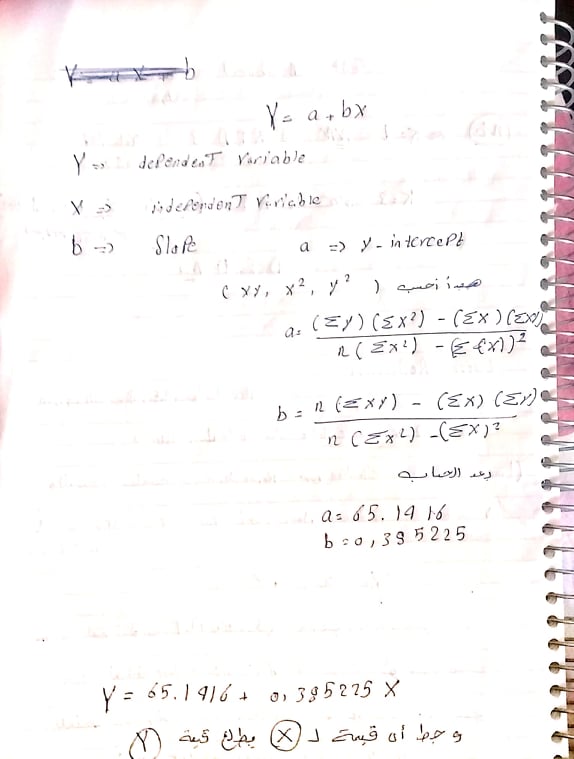


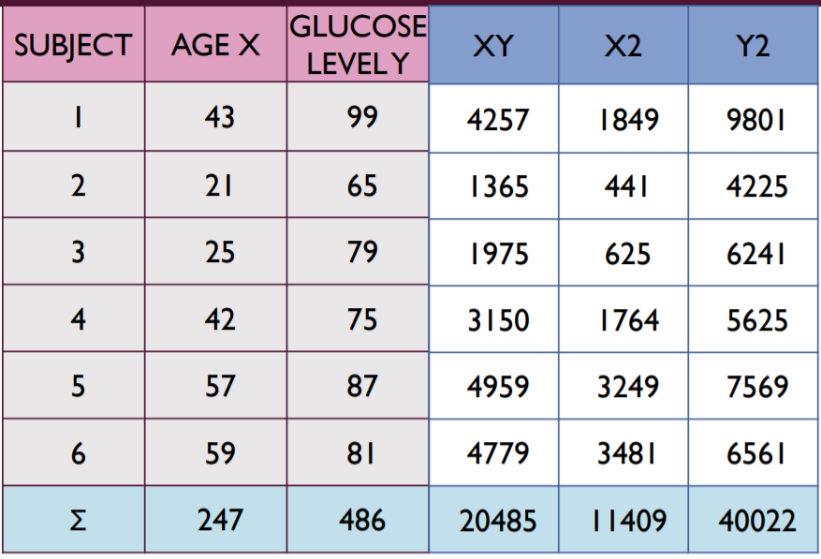


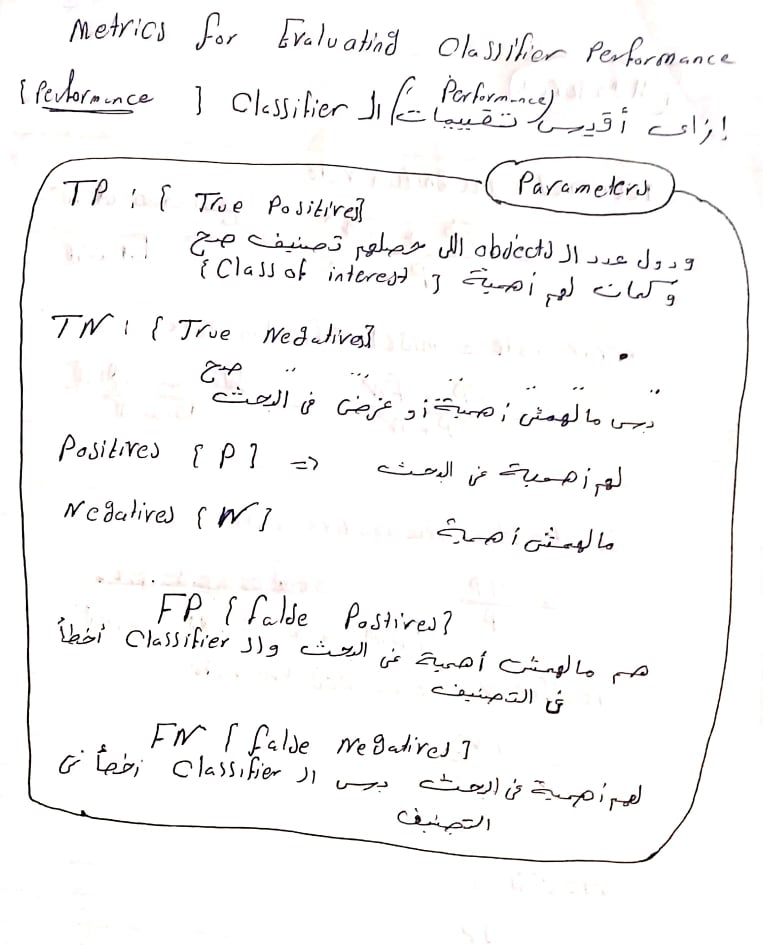


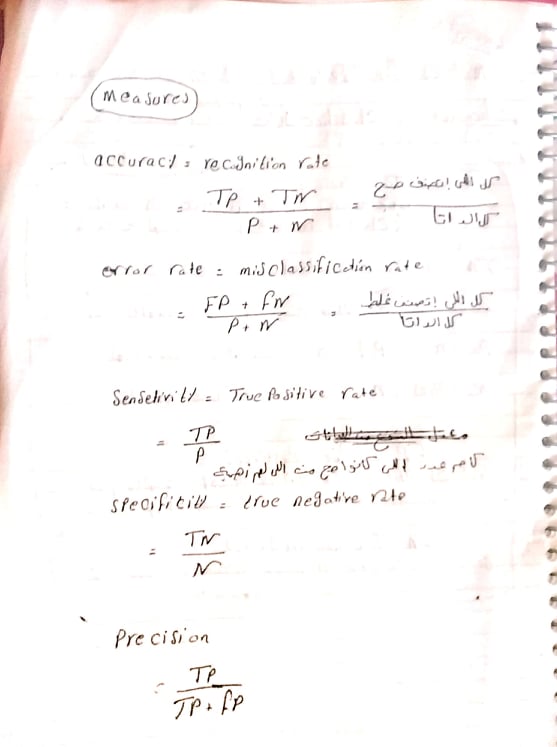


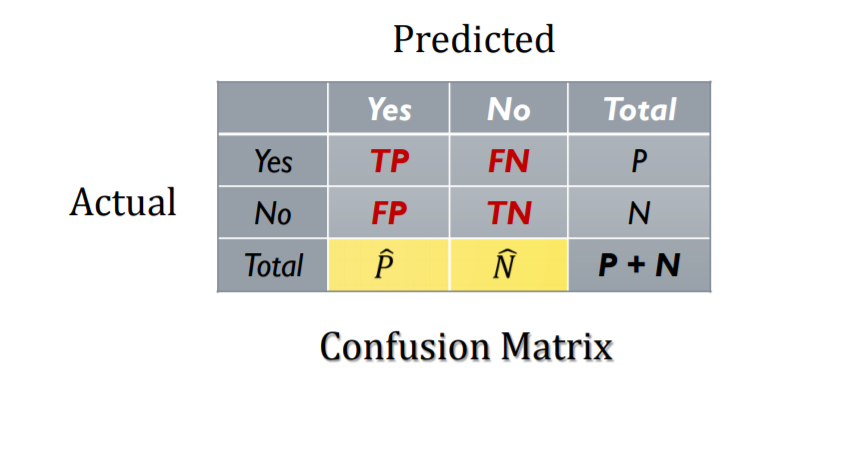


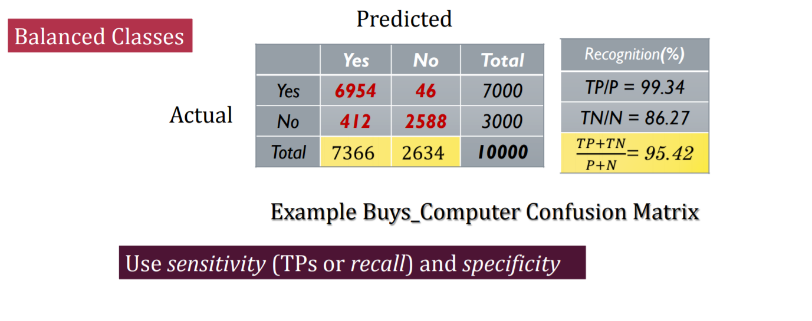
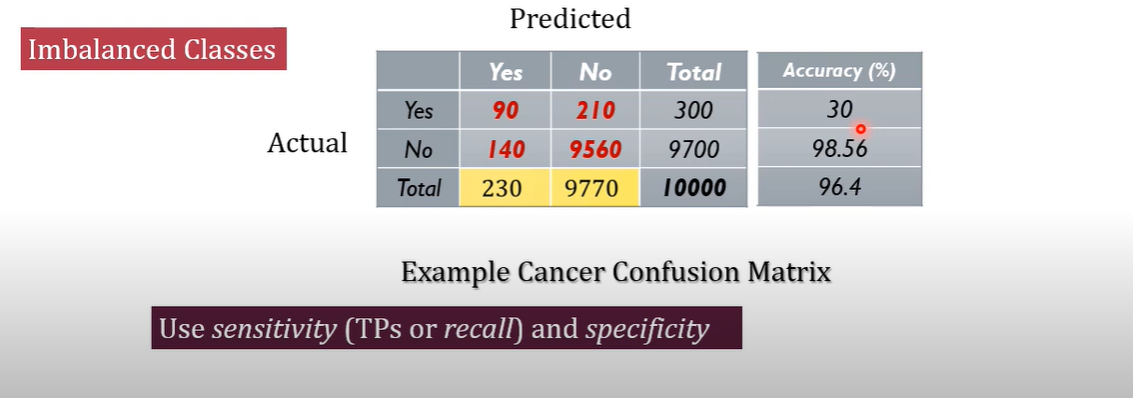




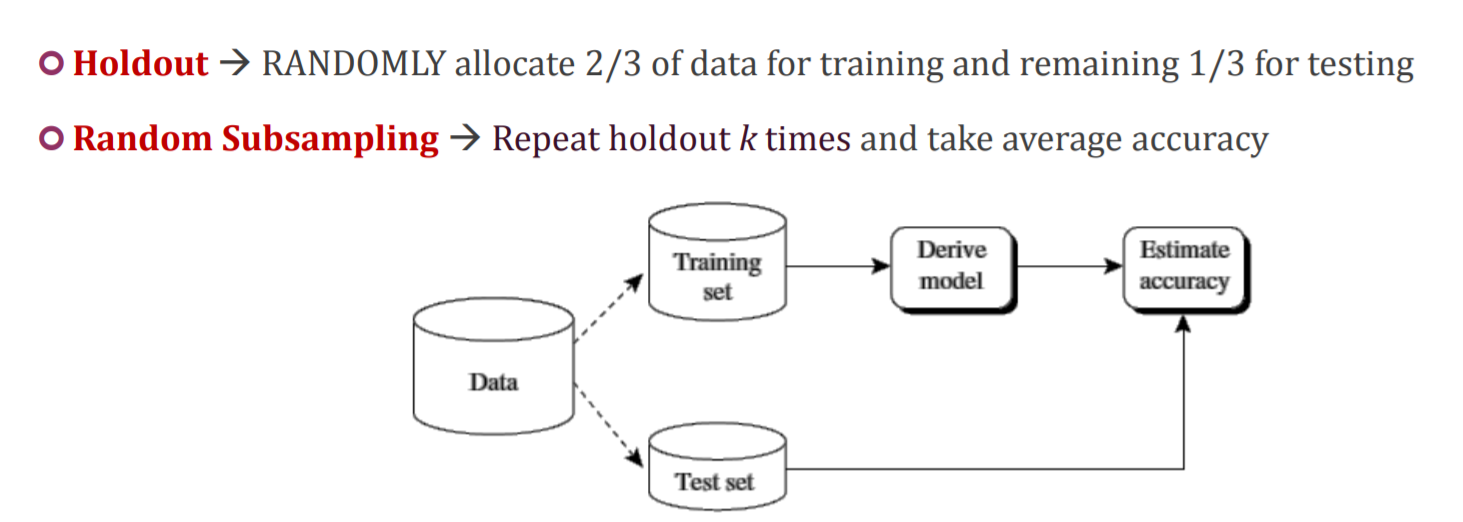


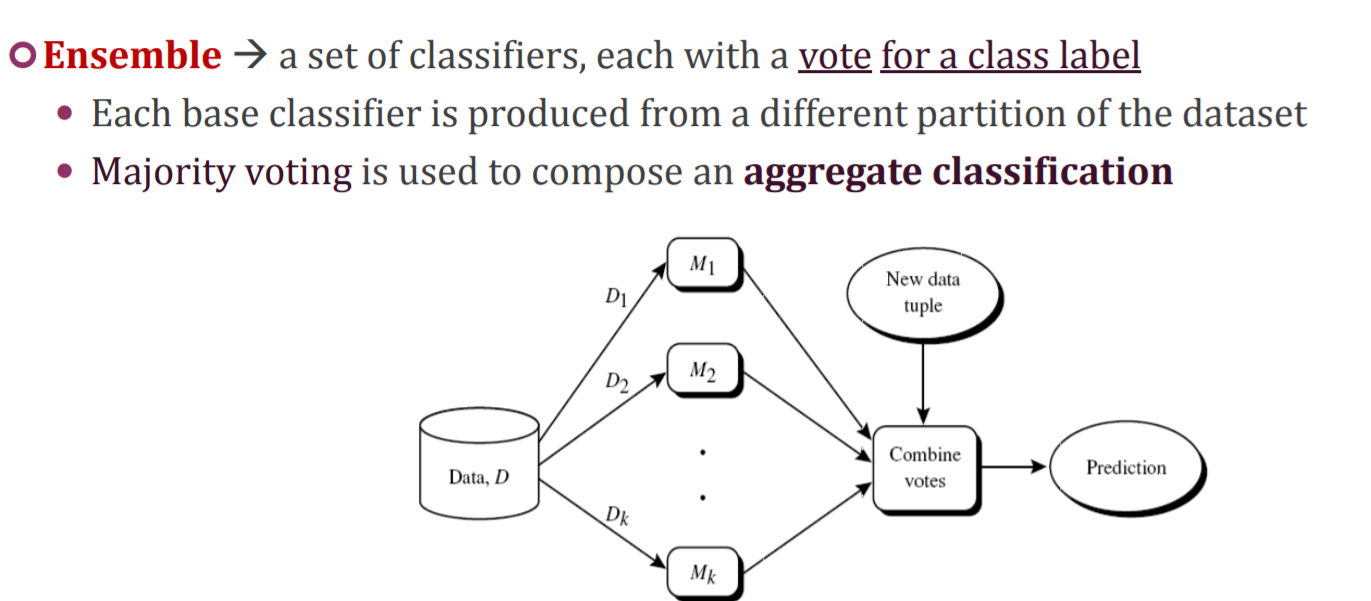
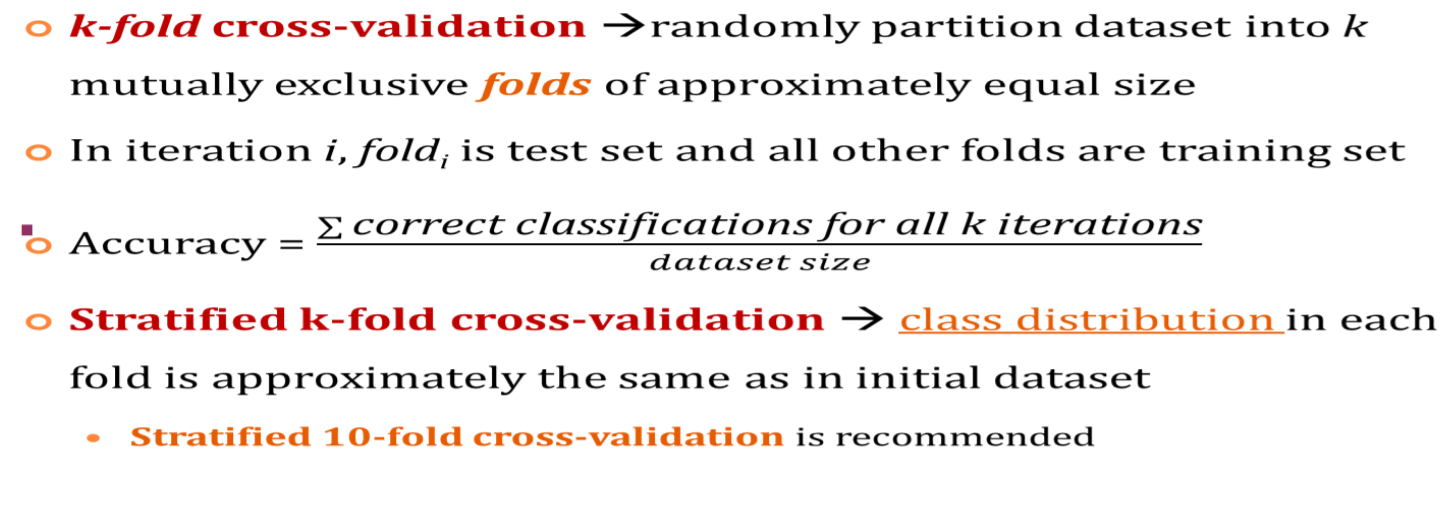


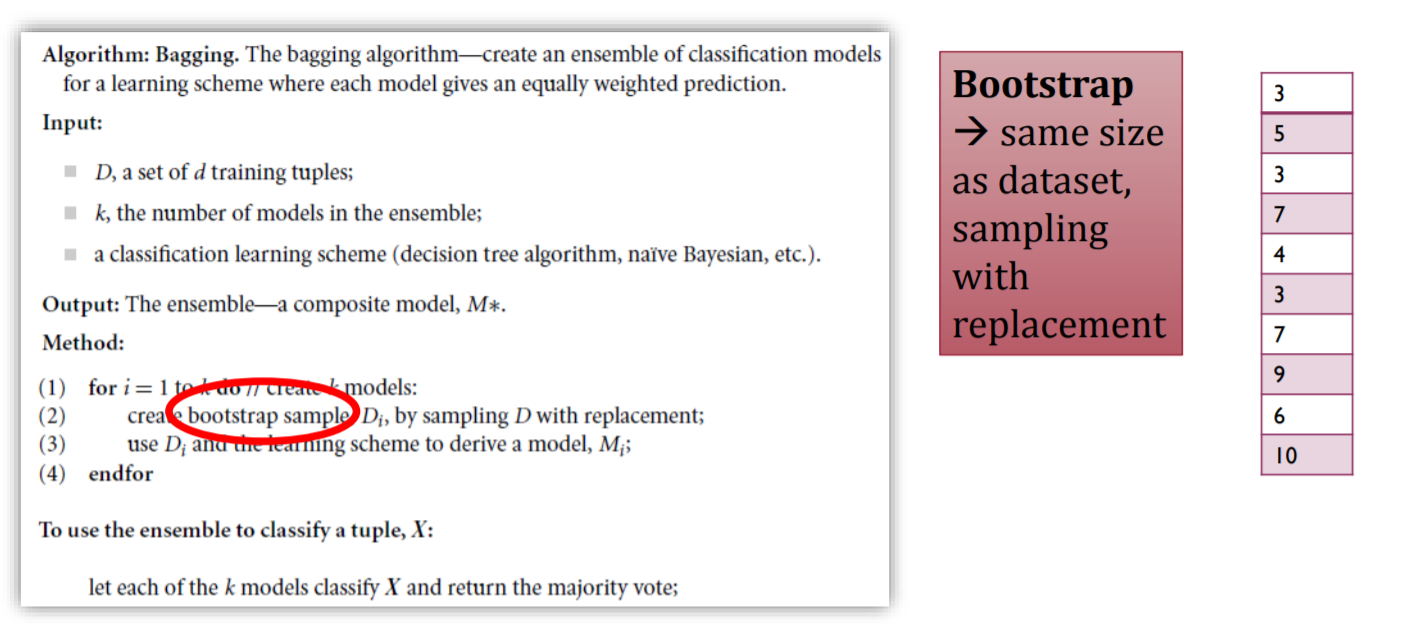
* ال confusion matrix من أهم الاشكال الى بتخلينا نقيس ال performance لل classifirer
* الصفوف بتوضح ايه القيم الحقيقية اللى المفروض ال classifier لو هو مية في المية صحيح يطلعها
* الأعمدة بتوضح ايه القيم الفعلية اللى تنبأ
* بيها ال classifier

* صحيح ال Accuracy بتاعت 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



* 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
*  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
* 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 classifier بتاعى



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