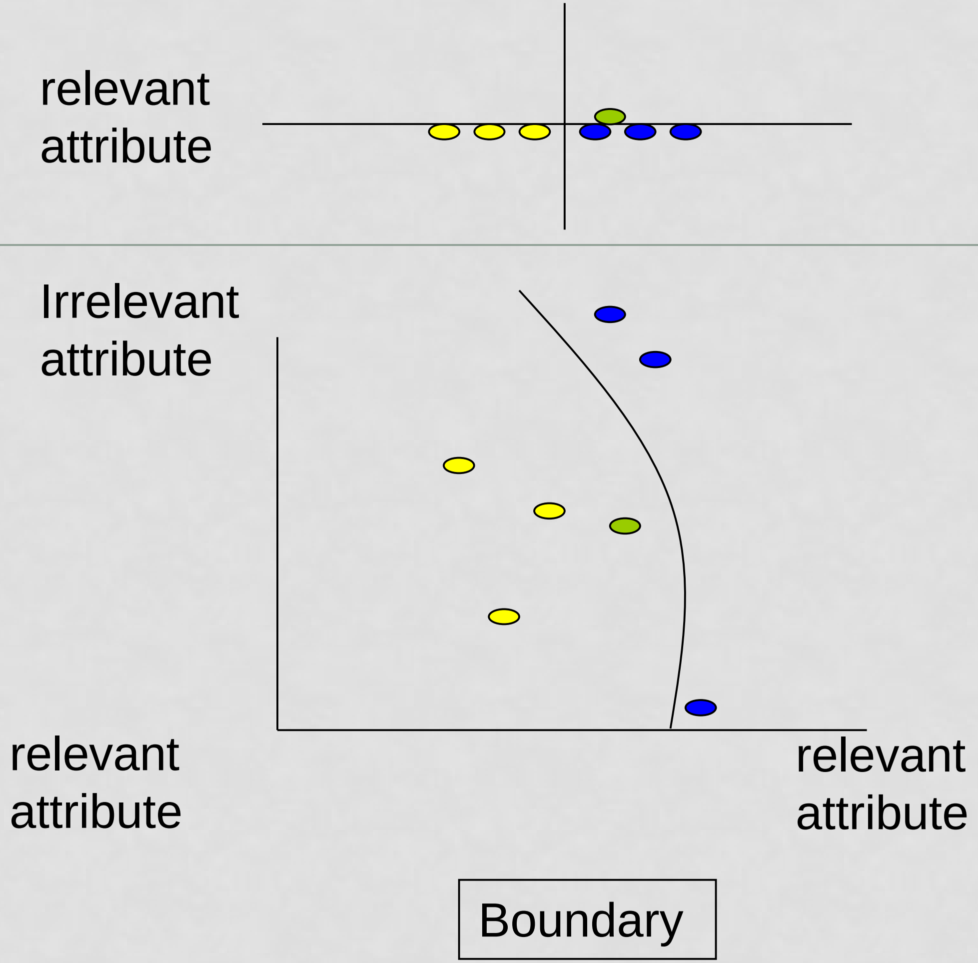
Attribute selection: motivation

- Some attributes can be **irrelevant** (such as "eye color" in order to predict payment of a loan)
- Some attributes can be **redundant** to some extent (such as "salary" and "social class") for predicting payment of a loan
- **Curse of dimensionality**: The number of required instances for learning a model that generalizes well, can grow very quickly with the number of dimensions (attributes)

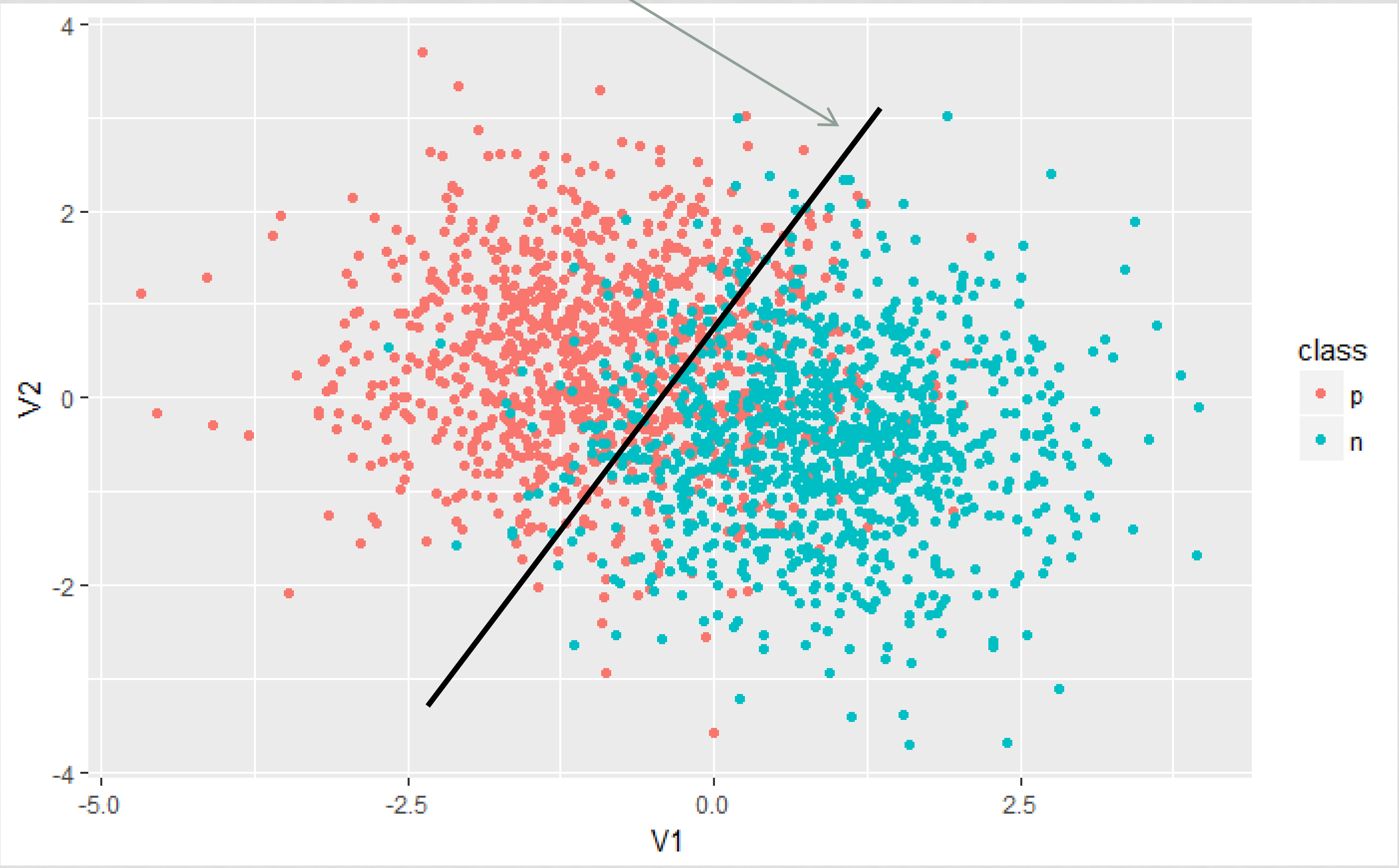
- Learning is slower if there are many attributes (e.g.: Decision trees is $O(n*m^2)$ SVM is $O(n*m)$) where m = number of attributes
- Some classifiers can get confused by irrelevant / redundant attributes (e.g. Naive Bayes and KNN)
- Having too many attributes (specially irrelevant ones) may result in **overfitting**, because it provides the model with extra arbitrary degrees of freedom to fit the data (i.e. it increases the complexity of the model)

- Sometimes it is useful to know which attributes are relevant (e.g. which genes are able to predict cancer?). This is for the benefit of the analyst (person).

Irrelevant attributes

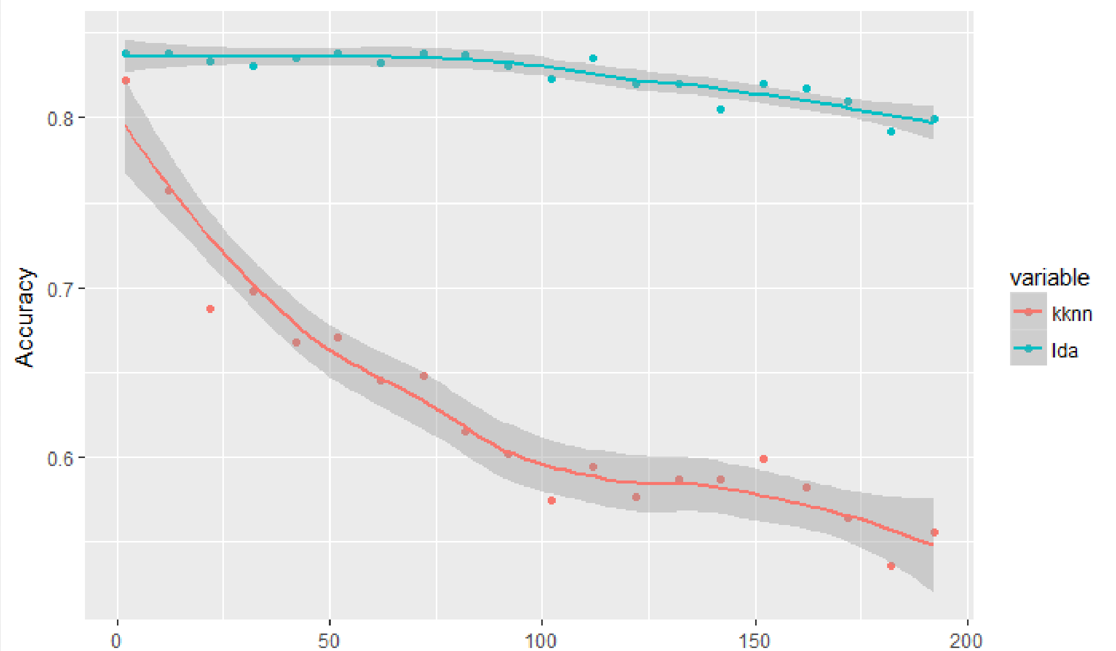


- Artificial data generated from two bi-variate gaussian distributions
- 1000 instances for the red class, 1000 instances for blue
- Optimal boundary is a line



EFFECT OF IRRELEVANT ATTRIBUTES ON KNN AND LINEAR MODELS

- Irrelevant attributes are added to the data matrix as columns with random values.



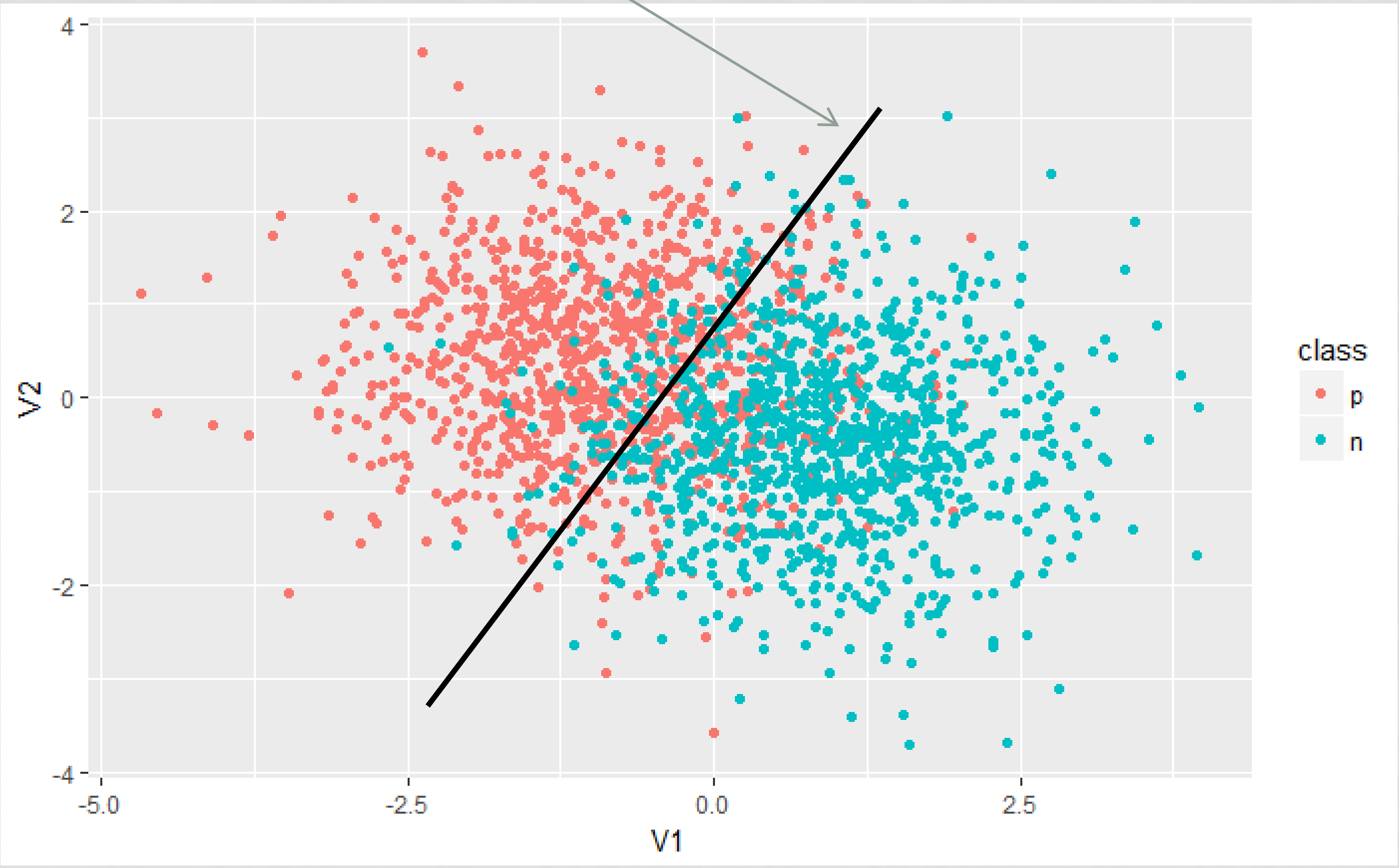
Number of irrelevant attributes

LDA = Linear Discriminant Analysis
KNN = K-nearest neighbour

THE CURSE OF DIMENSIONALITY

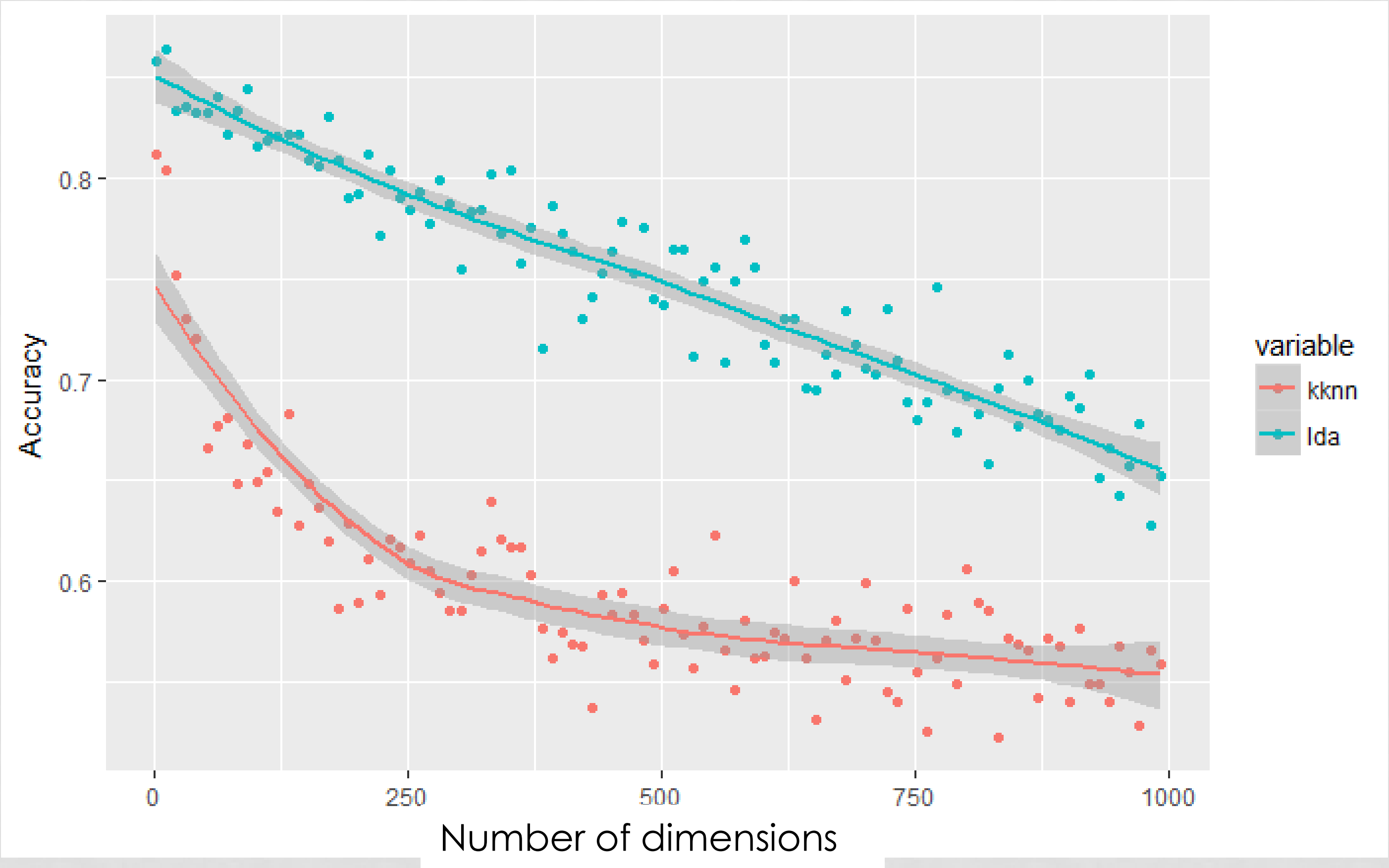
- The number of instances required to obtain a good model grows quickly with the number of attributes.
- It is not that the attributes are irrelevant, but that there are too many, in relation to the number of instances.

- Artificial data generated from two bi-variate gaussian distributions
- 1000 instances for the red class, 1000 instances for blue
- Optimal boundary is a line



- Number of dimensions d (attributes) grows from 2 to 1000
- Data is always generated from two d-variate gaussians, similar to the previos slide
- But the amount of data is kept constant: 1000 instances for the red class, 1000 instances for blue
- Accuracy decreases as dimension increase, despite all attributes being relevant.

- (for linear models, a rule of thumb is that you should have at least 5 instances for every attribute).



LDA = Linear Discriminant Analysis
KNN = K-nearest neighbour

THE CURSE OF DIMENSIONALITY

- Conclusion: if there is not a good relation of available data to the number of attributes, classification may not be accurate, **even if all the attributes are relevant**

ADVANTAGES OF ATTRIBUTE / FEATURE SELECTION

- Improve generalization of the classifier (removing irrelevant and redundant attributes)
  - However, bear in mind that some learning methods are able to deal with irrelevant attributes indirectly via hyper-parameters. For instance, shallow decision trees indirectly force the algorithm to choose the most relevant attributes. Linear models, such as Lasso or Elastic Net, can select important attributes via “regularization”.
- Speed up the learning process (but it is necessary to add the time for the attribute selection phase)

ATTRIBUTE SELECTION METHOD TYPES

Methods

- **Filter**: feature importance is evaluated for each attribute individually, using a simple statistical / mathematical method. Then, features are ranked and the worse ones are discarded.

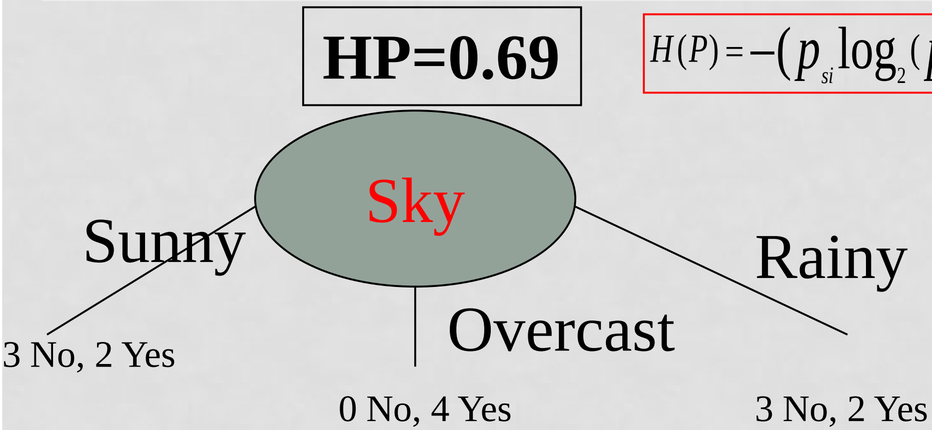
- **Wrapper**: **subsets** of attributes are evaluated (rather than individual attributes).

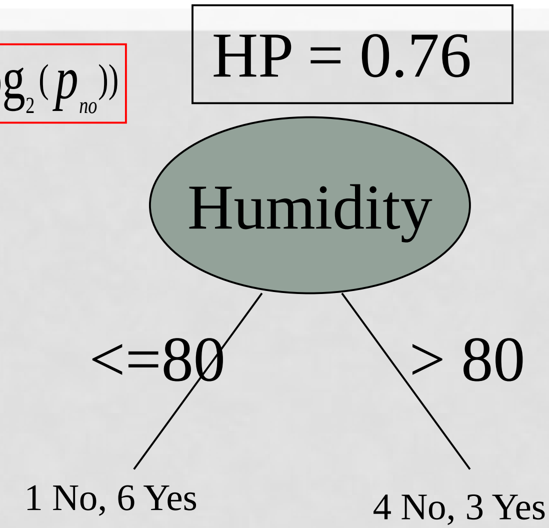
Ranking

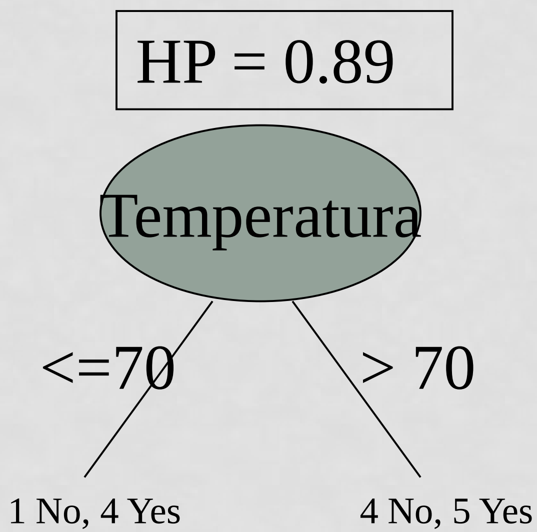
- Given input attributes $A_1, A_2, \ldots, A_n$, each A_i is evaluated **individually**, computing its correlation or dependency with the class, independently of the rest of attributes (i.e. attributes are considered individually, rather than subsets)
- An attribute A_1 is correlated with the class, if knowing its value implies that the class can be predicted more accurately
  - For instance, car speed is correlated with having an accident. But the Social Security Number of the driver is not.
  - For instance, salary is (inversely) correlated with credit default
- How to evaluate / rank attributes (attribute/class correlation):
  - Entropy (information gain), like in decision trees
  - Chi-square
  - Mutual information
  - …
- Once evaluated and ranked, the worst attributes can be removed (according to a threshold)

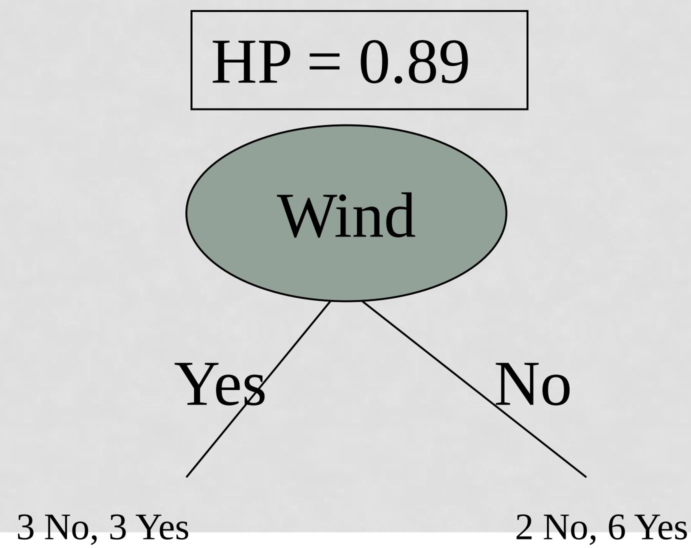
Entropy / Information Gain for ranking attributes

$$H(P) = -(p_{si} \log_2(p_{si}) + p_{no} \log_2(p_{no}))$$









Entropy / Information Gain for ranking attributes

Sky generates as many partitions as values (3: sunny, outcast, rainy)

Sky

Sunny

S	T	H	W	Ten
Sunny	85	85	No	No
Sunny	80	90	Yes	No
Sunny	72	95	No	No
Sunny	69	70	No	Yes
Sunny	75	70	Yes	Yes

"3 No, 2 Yes"

"no" majority

Overcast

S	T	H	W	Ten
Outcast	83	86	No	Yes
Outcast	64	65	Yes	Yes
Outcast	72	90	Yes	Yes
Outcast	81	75	No	Yes

"0 No, 4 Yes"

Perfect partition

Rainy

S	T	H	W	Ten
Rainy	70	96	No	No
Rainy	68	80	No	Yes
Rainy	75	80	No	Yes
Rainy	65	70	Yes	No
Rainy	71	91	Yes	No

"3 No, 2 Yes"

"no" majority

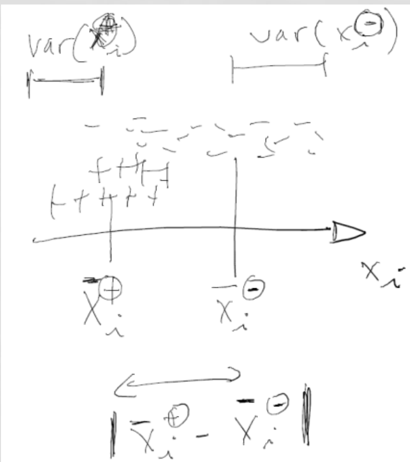
RANKING WITH MUTUAL INFORMATION

If x and y are independent,
p(x,y)=p(x)*p(y)

$$I(X,Y) = \sum_{i=1}^{n_x}\sum_{j=1}^{n_y} P(X=x_i, Y=y_j)\log\frac{P(X=x_i, Y=y_j)}{P(X=x_i)\,P(Y=y_j)}$$

- i means the values of attribute x, j means the values of class y
- I(x,y)=0 if x and y are independent (log(1) = 0)
- I(x,y)>= 0 (the more correlated, the larger is mutual information)
- If x or y are continuous, discretize them

F-SCORE (FISHER SCORE)



- Fisher score of the ith attribute in a 2-class problema (classes (0) (1))

$$\mathrm{FiR}_i = \frac{\left|\overline{X}_i^{(0)} - \overline{X}_i^{(1)}\right|}{\sqrt{\mathrm{var}(X_i)^{(0)} + \mathrm{var}(X_i)^{(1)}}},$$

Useful for continuous features and classification problems (2-class)

- Directly proportional to the distance between the two class means.
- and inversely proportional to the sum of variances.
- The farther away the means are, and the smaller the class variances, the better (the more is going the attribute to be able to separate instances belonging to the two classes).

RANKING WITH CHI SQUARE

- Liu, H., & Setiono, R. (1995, November). Chi2: Feature selection and discretization of numeric attributes. In *tai* (p. 388). IEEE.

CHI-SQUARE

- Class Y and attribute X (with two discrete values: a and b). $Obs_{x,y}$ is the number of instances observed for which class is y and attribute X=x.
- Total number of instances is:
  - $n = Obs_{a,p} + Obs_{b,p} + Obs_{a,n} + Obs_{b,n}$

Attribute X

Class Y

Data	X= a	X= b
Y= positive	$Obs_{a,p}$	$Obs_{b,p}$
Y= negative	$Obs_{a,n}$	$Obs_{b,p}$

Useful for classification problems and categorical attributes.

CHI-SQUARE

- If class Y and attribute X are independent, then: prob(X,Y) = prob(X) * prob(Y)
- If X and Y were independent:

  Prob(X=x, Y=y) = Prob(X=x)*Prob(Y=y)
- Eg: P(X=a, Y=p) = P(X=a)*P(Y=p)
  - Prob(X=a) = ($Obs_{a,p}$+ $Obs_{a,n}$) / n
  - Prob(X=b) = ($Obs_{b,p}$+ $Obs_{b,n}$) / n
  - Prob(Y=p) = ($Obs_{a,p}$+ $Obs_{b,p}$) / n
  - Prob(Y=n) = ($Obs_{a,n}$+ $Obs_{b,n}$) / n
- The expected number of instances with X=x and Y=y, assuming X and Y are independent, is:

  $Esp_{x,y}$ = n*P(X=x,Y=y) = n*P(X=x)*P(Y=y)

Data	X= a	X= b
Y= positive	$Obs_{a,p}$ $Exp_{a,p}$	$Obs_{b,p}$ $Exp_{b,p}$
Y= negative	$Obs_{a,n}$ $Exp_{a,p}$	$Obs_{b,p}$ $Exp_{b,p}$

CHI-SQUARE

- Under the assumption of independence, X^2 follows a chi-squared distribution with 1 degree of freedom.

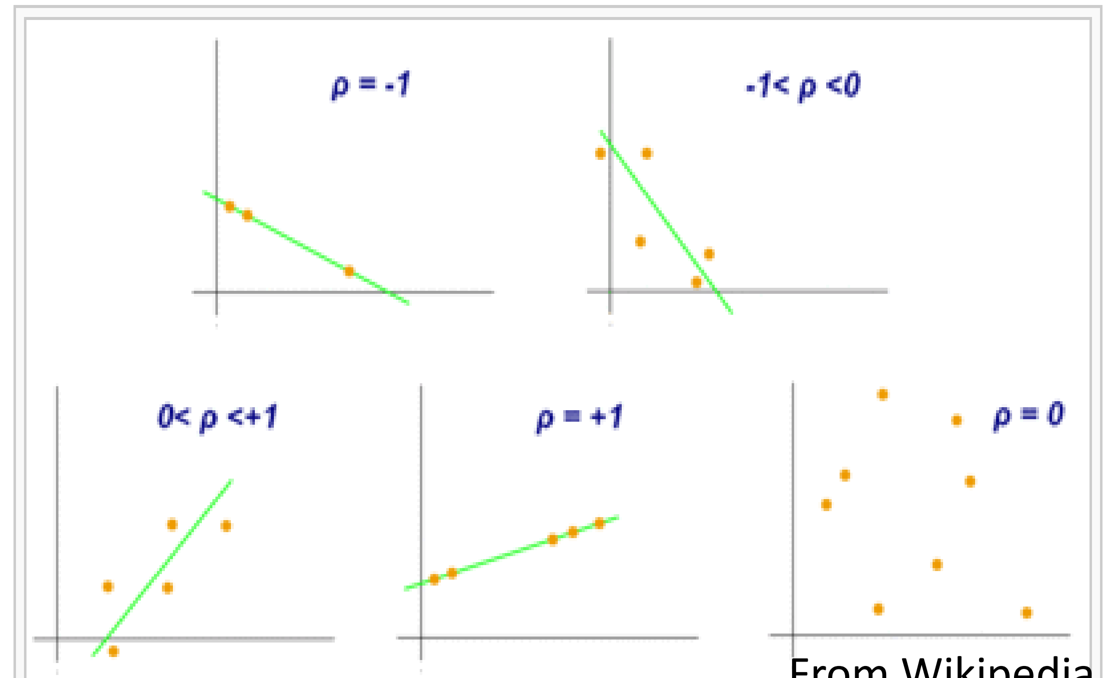
$$X^2 = \sum_{y=p,n} \sum_{x=a,b} \frac{(\mathrm{Obs}_{x,y} - \mathrm{Esp}_{x,y})^2}{\mathrm{Esp}_{x,y}}$$

- For feature selection, we just order attributes by X^2, and select the top k ones.

Data	X= a	X= b
Y= positive	$\mathrm{Obs}_{a,p}$ $\mathrm{Exp}_{a,p}$	$\mathrm{Obs}_{b,p}$ $\mathrm{Exp}_{b,p}$
Y= negative	$\mathrm{Obs}_{a,n}$ $\mathrm{Exp}_{a,p}$	$\mathrm{Obs}_{b,p}$ $\mathrm{Exp}_{b,p}$

LINEAR CORRELATION

- Pearson coefficient -1 <= r <= +1
- Covariance(X,Y) = $E((X-\mu_X)(Y-\mu_Y))$
- Correlation(X,Y)= $r = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}$



From Wikipedia

FILTER IN SCIKIT

- sklearn.feature_selection.SelectKBest

`f_classif`

ANOVA F-value between label/feature for classification tasks.

`mutual_info_classif`

Mutual information for a discrete target.

`chi2`

Chi-squared stats of non-negative features for classification tasks.

`f_regression`

F-value between label/feature for regression tasks.

`mutual_info_regression`

Mutual information for a continuous target.

Filter

- Advantages: fast
- Disadvantages:
  - Redundant attributes are not removed
  - Attributes are evaluated individually. Therefore, attribute interaction is not detected
  - Attribute interaction: subsets of attributes that work well together but not individually. In fact, they are likely to be discarded.

WHAT IS ATTRIBUTE INTERACTION?

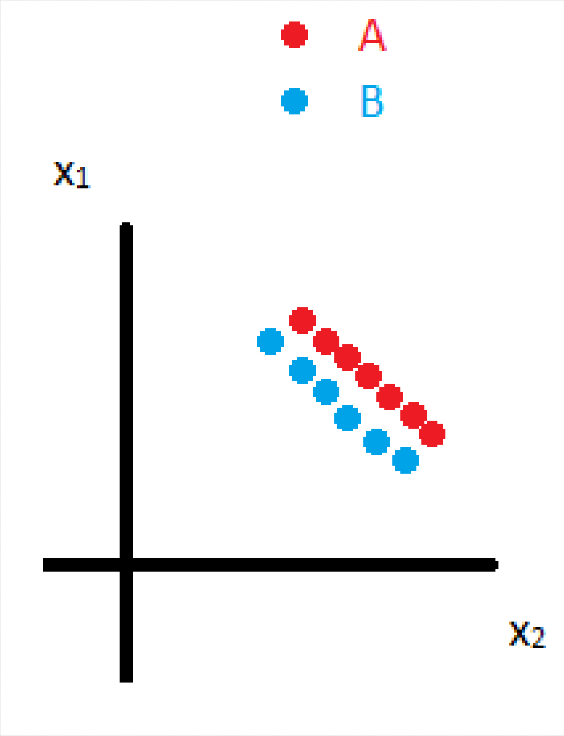
- Sometimes, two attributes are not predictive separately, but they are if they are used together (**attribute interaction**)
- Example:
  - Classification problem into two classes: computer science and anthropology
  - Binary attributes "intelligence" and "artificial" which are true if these words appear in the text and false otherwise
  - Separately, they do not allow to differenciate between computer science and anthropology textbooks, because both words appear in both types of books:

    IF intelligence=yes **THEN** ?; **IF** artificial=yes **THEN** ?
  - But together they can

    IF artificial=yes **AND** intelligence=yes **THEN** "computer science"
- Therefore, in the general case, the aim of attribute selection is to find the smallest **subset** of attributes that "work well" together. In this case, the subset would be {"artificial", "intelligence"}

Example of attribute interaction

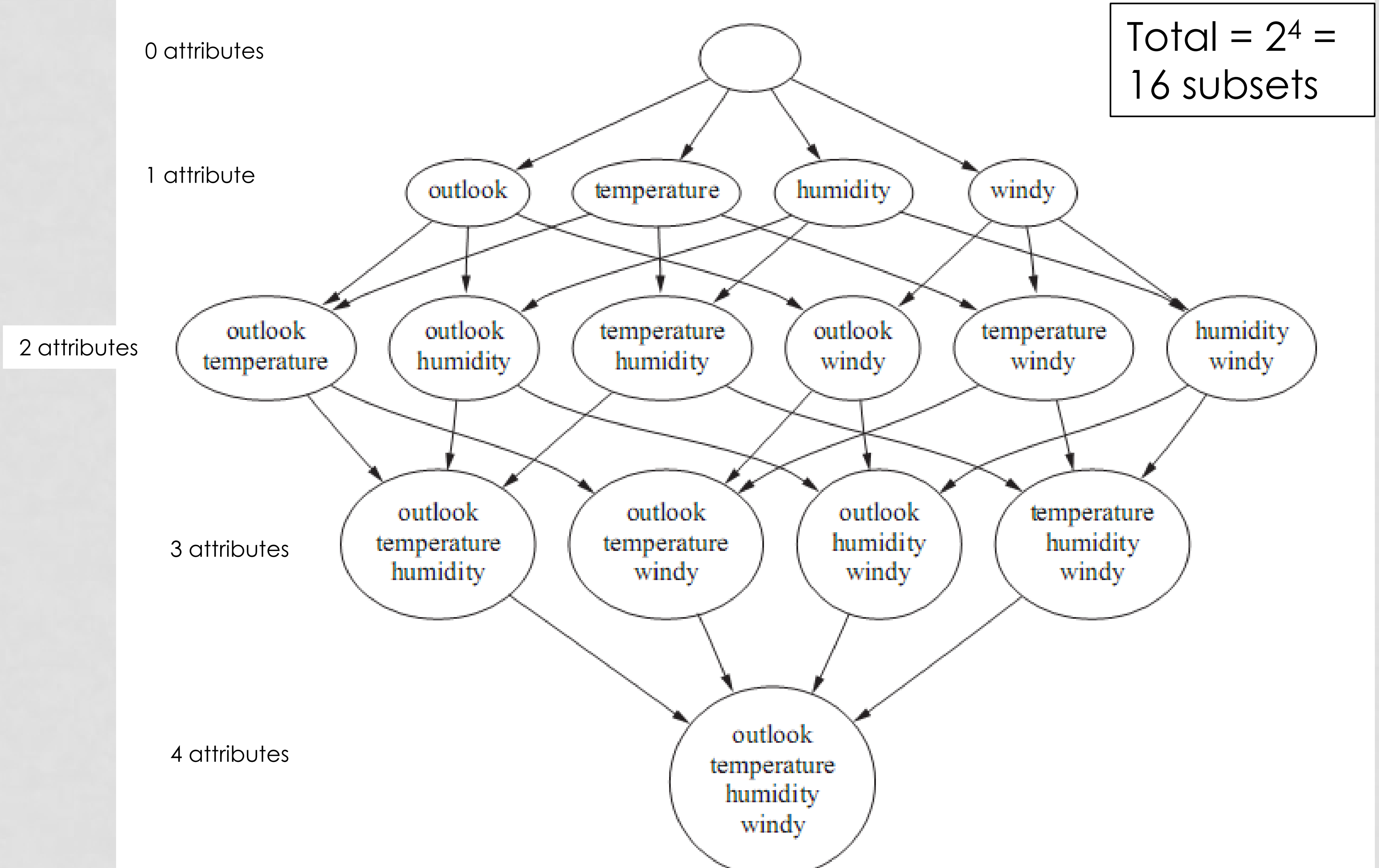
- x1 and x2, individually, cannot separate class A from class B, therefore filter methods would rank them poorly
- But together, they can (with a linear classifier, for instance)



EXHAUSTIVE SEARCH

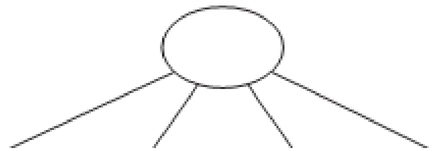
- Test all possible subsets of attributes
- If there are 4 input attributes A, B, C, D
- The list of possible subsets to try is $2^4=16$: {A, B, C, D}, {A, B, C}, {A, B, D}, {B, C, D}, {A, C, D}, {A, B}, {A, C}, ..., {A}, {B}, {C}, {D}
- For n large, this is not feasible:
  - n = 10 => 2^{10} = 1024 subsets
  - n = 20 => 2^{20} = 1048576 subsets
  - n = 30 => 2^{30} = 1073741824 subsets
  - …

Define the space of subsets of attributes

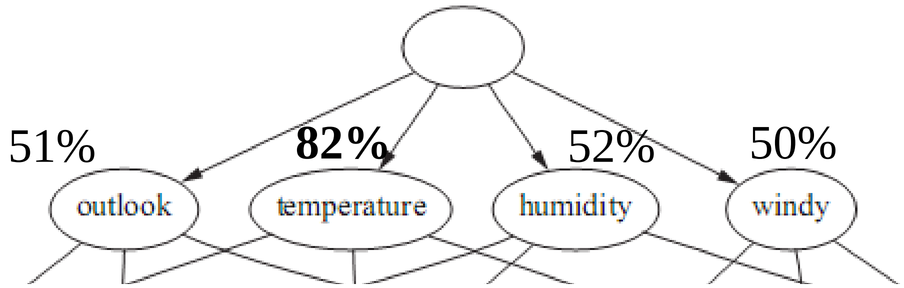


Total = 2^4 = 16 subsets

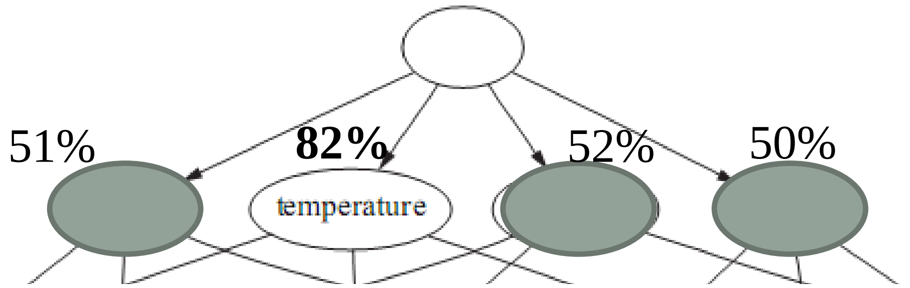
Forward search in the space of subsets of attributes



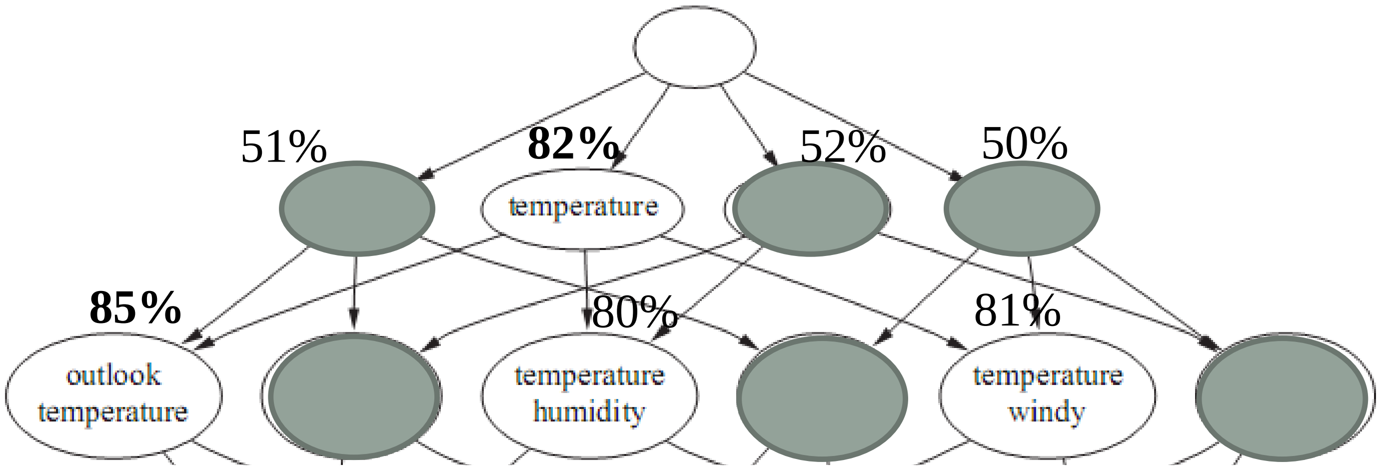
Forward search in the space of subsets of attributes



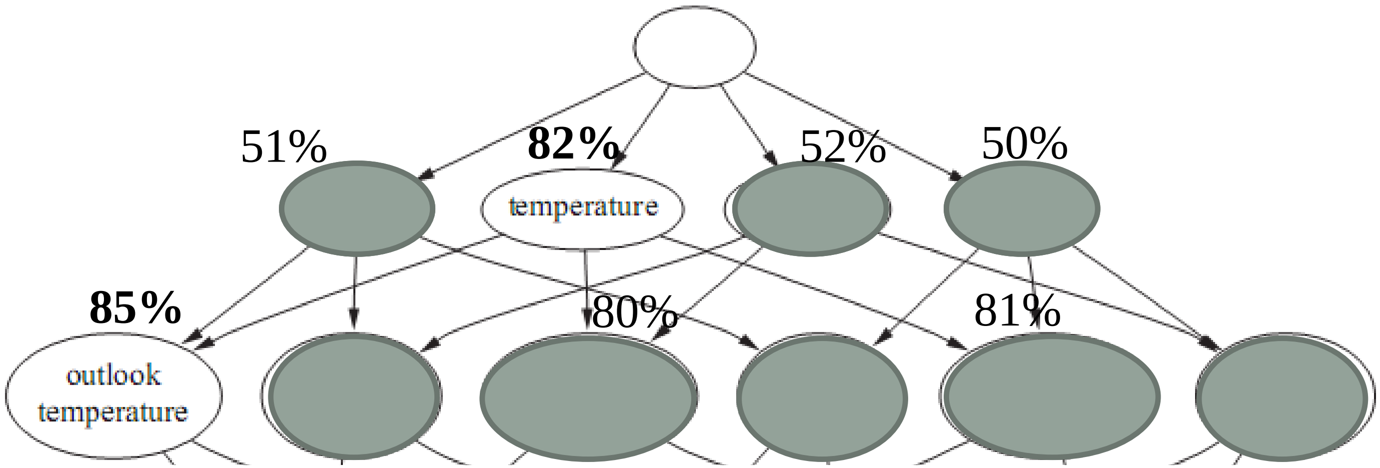
Forward search in the space of subsets of attributes



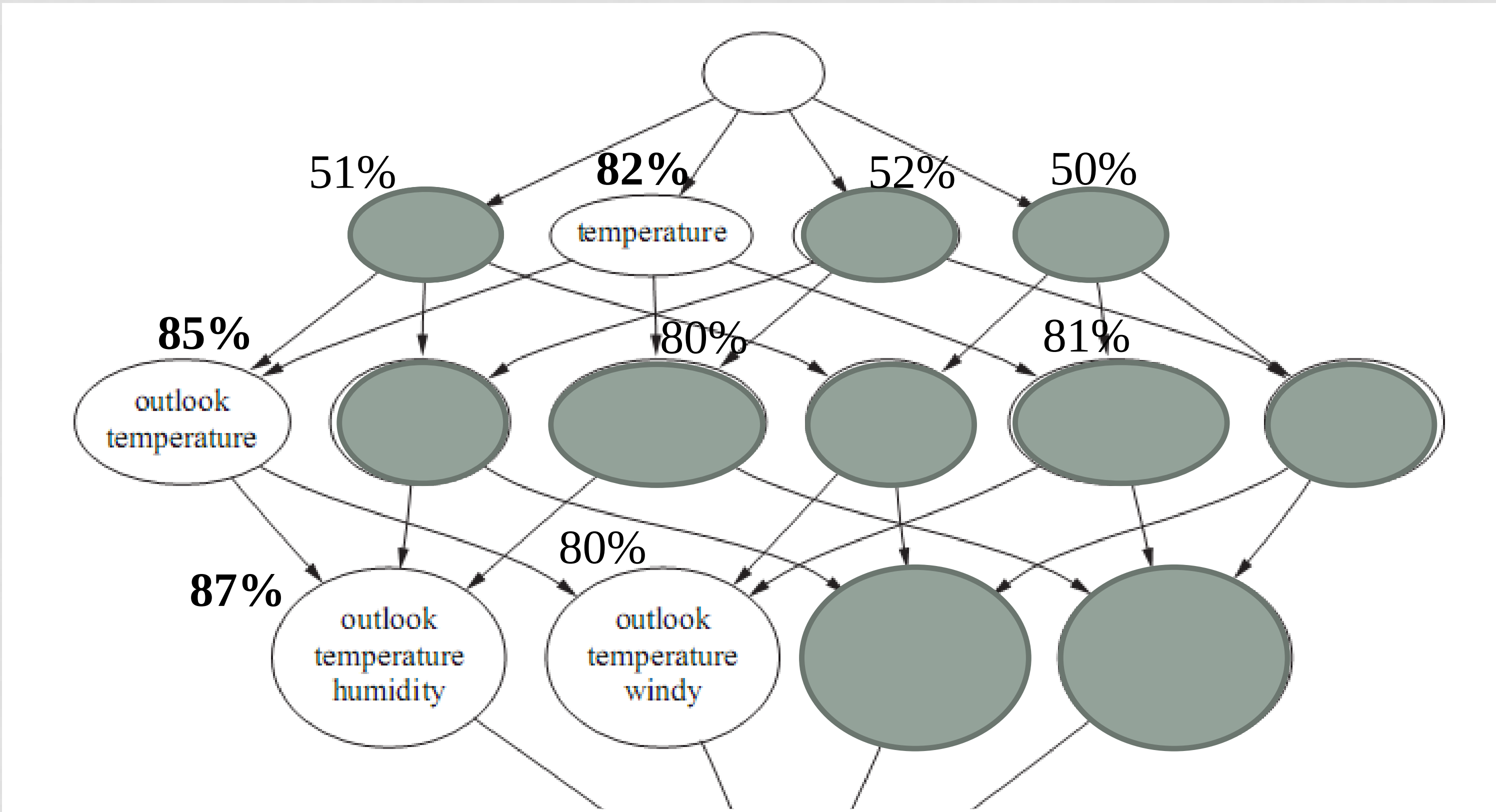
Forward search in the space of subsets of attributes



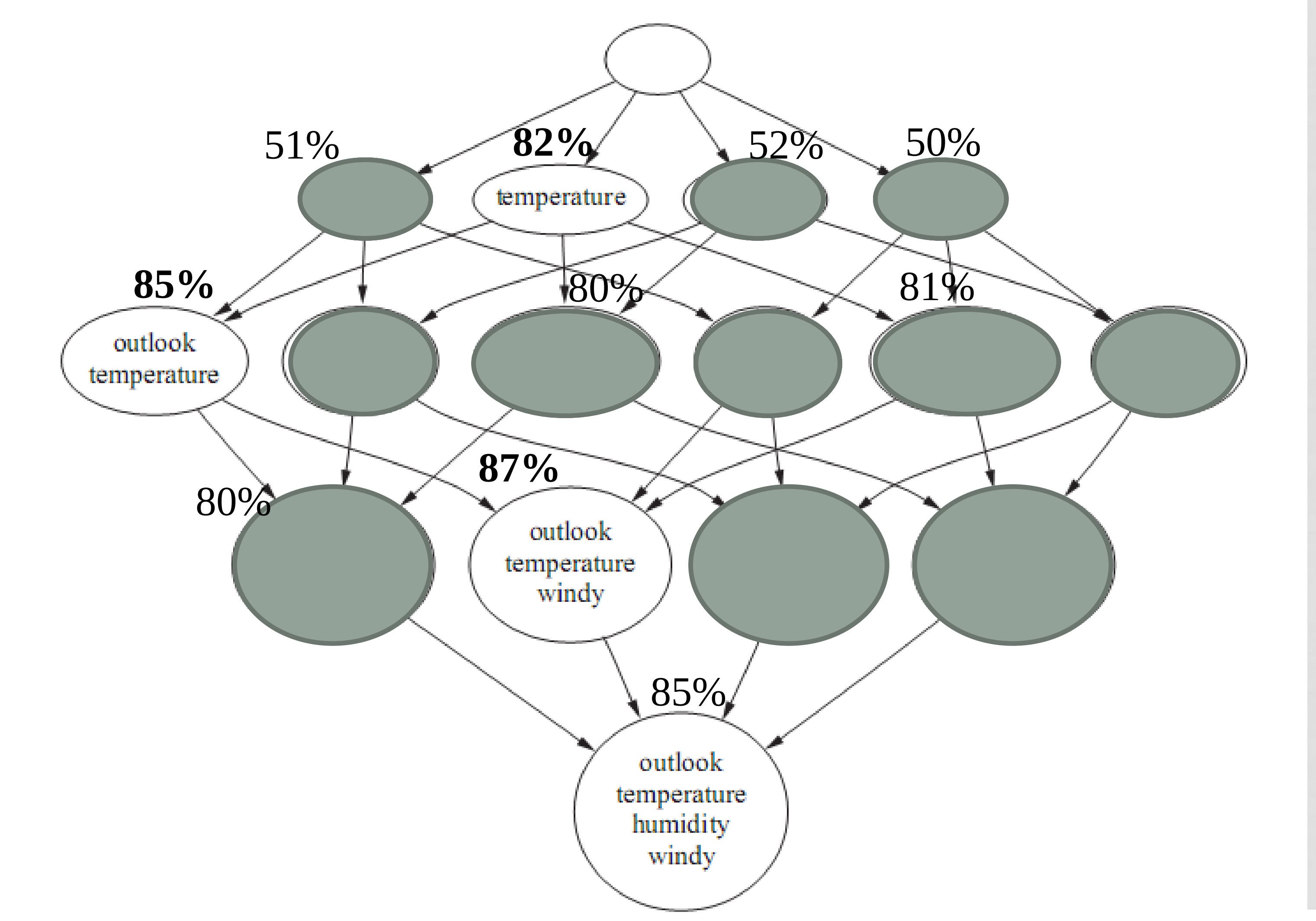
Forward search in the space of subsets of attributes



Forward search in the space of subsets of attributes

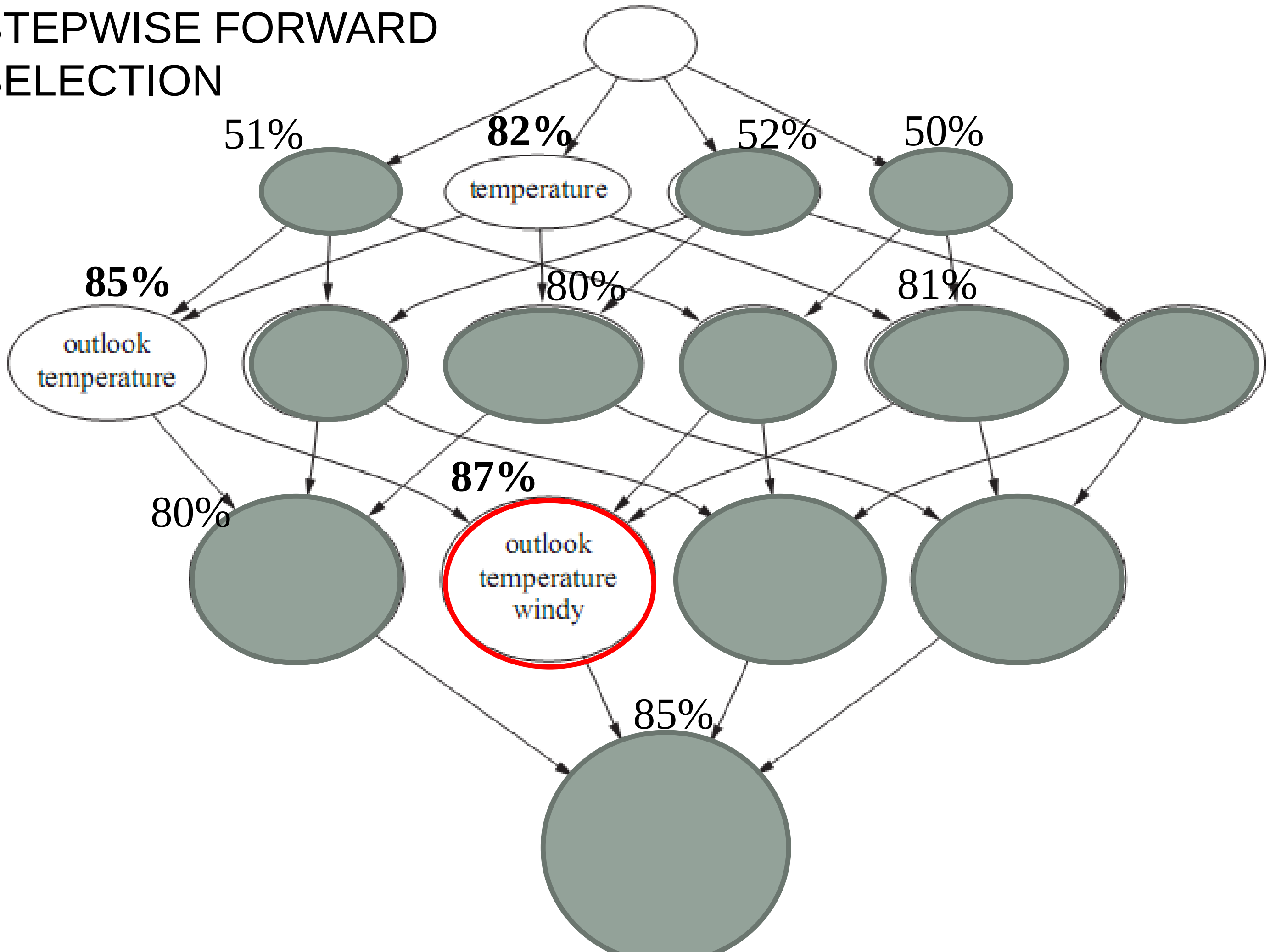


Forward search in the space of subsets of attributes



Forward search in the space of subsets of attributes

STEPWISE FORWARD SELECTION

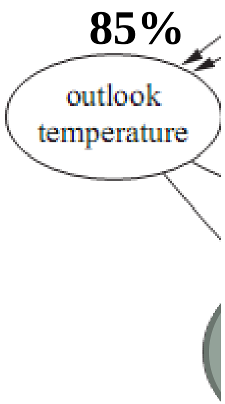


SUBSET EVALUATION

- How to evaluate subsets of attributes?
- We know how to evaluate a single attribute. E.g., Mutual Information in Filter: MI(A_3, Class)
- What about subset {A_2, A_7}?

Wrapper: subset evaluation

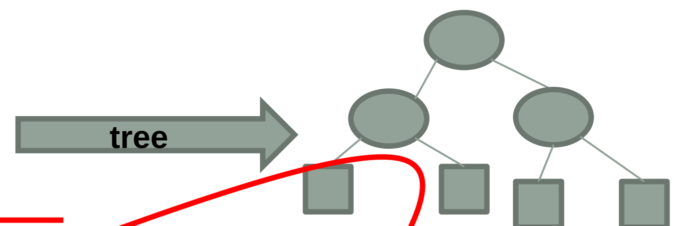
Outlook	Temperature	Humidity	Windy	Play



Outlook	Temperature	Play

Train

Validation



85%

SUBSET EVALUATION: Wrapper

- *Wrapper* methods evaluate a subset of attributes by building a model (e.g. a decision tree) that uses only those attributes and then computing its expected performance (e.g. success rate)

- E.g. in order to evaluate subset {A1, A3, A10}, a model M is trained that uses only those attributes. The evaluation of the subset is the accuracy obtained by model M.

SUBSET EVALUATION: Wrapper

- Advantages:
  - They obtain subsets of attributes for particular machine learning algorithms (like decision trees)
  - They actually evaluate subsets of attributes
- Disadvantages:
  - They obtain subsets of attributes which are too dependent of the particular machine learning algorithm used
  - Very slow (testing different attribute subsets involves building many models from training sets)
  - Although they are based on a good idea, they sometimes overfit

- In scikit-learn: Sequential Feature Selection
- https://scikit-learn.org/stable/modules/feature_selection.html#sequential-feature-selection
- Forward-SFS
- Backward-SFS