Lecture 8: Handling Numeric Attributes

Madhavan Mukund

https://www.cmi.ac.in/~madhavan

Data Mining and Machine Learning
August–December 2020

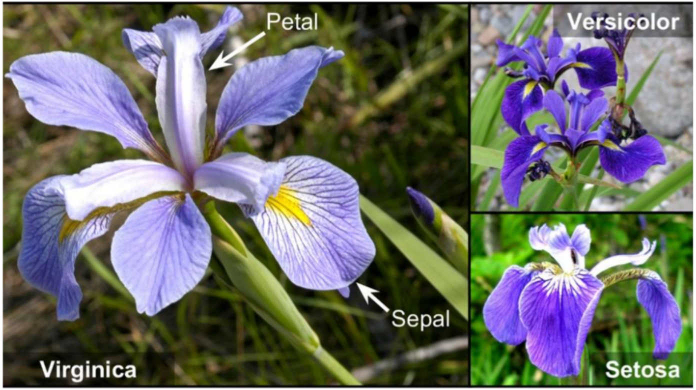
Categorical vs numeric attributes

- So far, all attributes have been categorical
- What age groups make up young, middle, old?
- How are these boundaries defined?
- How do we query numerical attributes?
  - Height, weight, length, income,

ID	Age	Has_job	Own_house	Credit_rating	Class
1	young	false	false	fair	**No**
2	young	false	false	good	**No**
3	young	true	false	good	**Yes**
4	young	true	true	fair	**Yes**
5	young	false	false	fair	**No**
6	middle	false	false	fair	**No**
7	middle	false	false	good	**No**
8	middle	true	true	good	**Yes**
9	middle	false	true	excellent	**Yes**
10	middle	false	true	excellent	**Yes**
11	old	false	true	excellent	**Yes**
12	old	false	true	good	**Yes**
13	old	true	false	good	**Yes**
14	old	true	false	excellent	**Yes**
15	old	false	false	fair	**No**

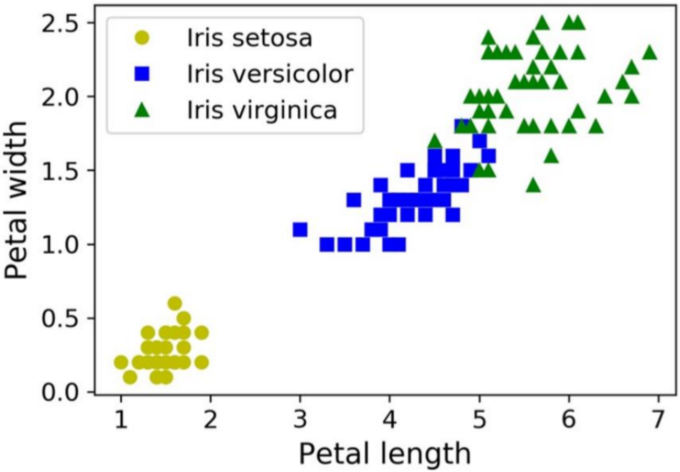
Iris dataset

- Iris is a type of flower
- Three species: *iris setosa*, *iris versicolor*, *iris virginica*
- Dataset has sepal length and width and petal length and width for 150 flowers



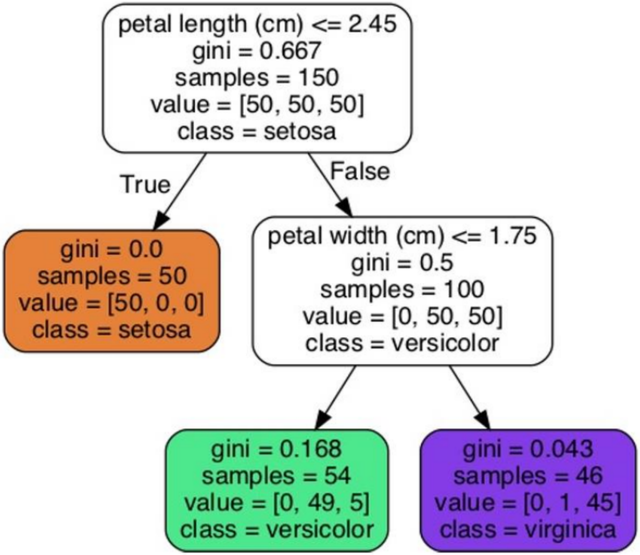
Iris dataset

- Iris is a type of flower
- Three species: *iris setosa*, *iris versicolor*, *iris virginica*
- Dataset has sepal length and width and petal length and width for 150 flowers
- Scatter plot for two attributes, petal length and petal width



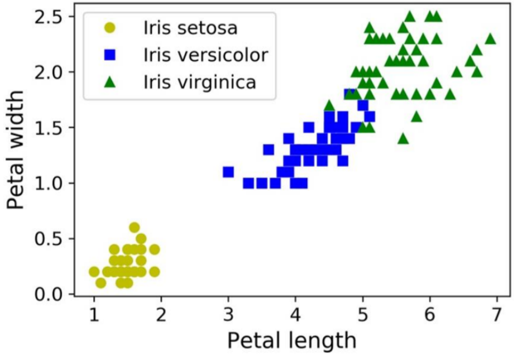
Iris dataset

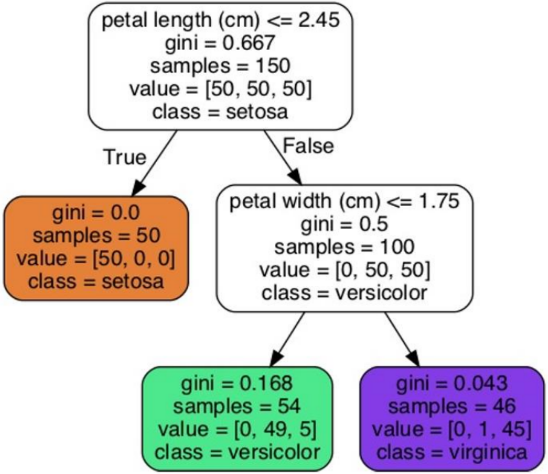
- Iris is a type of flower
- Three species: *iris setosa*, *iris versicolor*, *iris virginica*
- Dataset has sepal length and width and petal length and width for 150 flowers
- Scatter plot for two attributes, petal length and petal width
- Decision tree for this data set



Decision tree for iris dataset

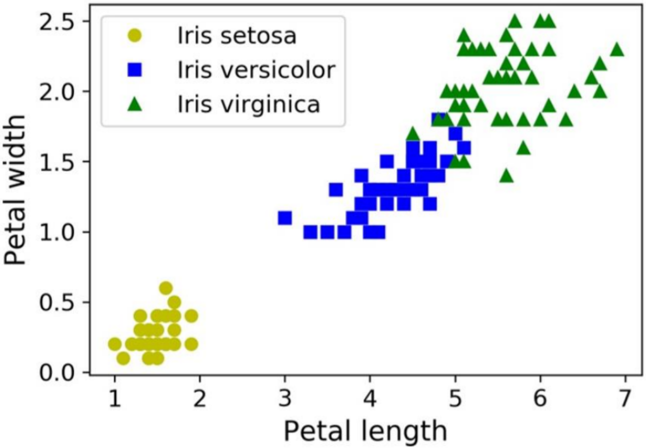
- Queries compare numerical attribute against a value
- How do we find these query values?





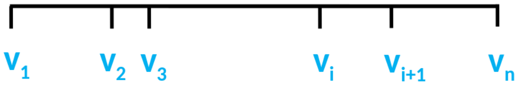
Querying numerical attributes

- Numerical attribute takes values in a range $[L, U]$
  - Petal length : $[1, 7]$
  - Petal width : $[0, 2.5]$
- Pick a value v in the range and check if $A \leq v$
- Infinitely many choices for v
- How do we pick a sensible one?



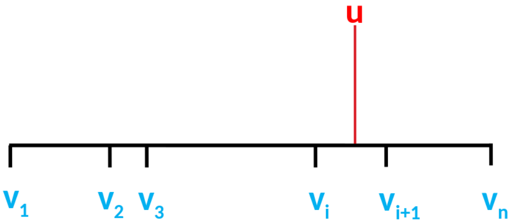
Querying numerical attributes

- Only n values for A in training data
  - Sort as $v_1 < v_2 < \cdots < v_n$



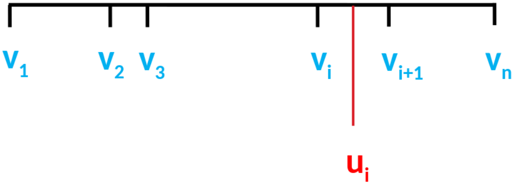
Querying numerical attributes

- Only n values for A in training data
  - Sort as $v_1 < v_2 < \cdots < v_n$
- Consider interval $[v_i, v_{i+1}]$
- For each $v_i \leq u < v_{i+1}$, query $A \leq u$ gives the same answer
- Only $n-1$ useful intervals to check



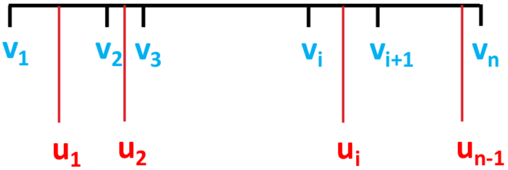
Querying numerical attributes

- Only n values for A in training data
  - Sort as $v_1 < v_2 < \cdots < v_n$
- Consider interval $[v_i, v_{i+1}]$
- For each $v_i \leq u < v_{i+1}$, query $A \leq u$ gives the same answer
- Only $n-1$ useful intervals to check
- Pick midpoint $u_i = (v_i + v_{i+1})/2$ as query value for each interval



Querying numerical attributes

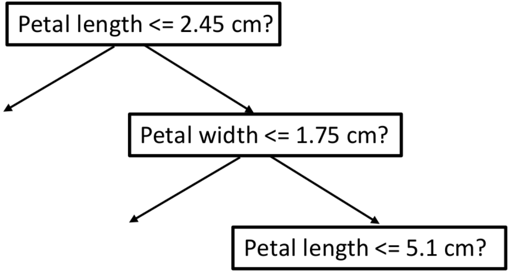
- Pick midpoint $u_i = (v_i + v_{i+1})/2$ as query value for each interval
- Each query $A \leq u_i$ partitions training data
- Choose the query $A \leq u_i$ with maximum information gain
- Assign this as the information gain for this attribute
- Compare across all attributes and choose best one



- Any point within an interval can be used
- May prefer endpoints — midpoints may not be meaningful values

Building a decision tree

- For each numerical attribute, choose query $A \leq v$ with maximum information gain
- Across all categorical and numerical attributes, choose the one with best information gain
- Categorical attrbutes can be queried only once on a path
- Numerical attributes can be queried repeatedly — interval to query keeps shrinking



Summary

- For numerical attributes, $n-1$ intervals in training data generate $n-1$ comparison queries
- Each query partitions the training data
- Choose the query that maximizes information gain
- The same numerical attribute may be queried repeatedly along a path