Data Mining Outputs

Decision Table for Weather Data Set

Outlook	Humidity	Play
Sunny	High	No
Sunny	Normal	Yes
Overcast	High	Yes
Overcast	Normal	Yes
Rainy	High	No
Rainy	Normal	No

Naïve Bayes Classifier

Using the weather database, determine the likelihood that they play or not on a :

sunny

hot day

normal humidity

no wind

Naïve Bayes Classifier

Want:

Pr[play='yes' | outlook = 'sunny' & temp = 'hot' & humidity = 'normal' & windy = 'no']

and

Pr[play='no' | outlook = 'sunny' & temp = 'hot' & humidity = 'normal' & windy = 'no']

Therefore I need values for:

Pr[outlook = 'sunny' | play = 'yes']
Pr[outlook = 'sunny' | play = 'no']
Pr[temp = 'hot' | play = 'yes']
Pr[temp = 'hot' | play = 'no']
Pr[humidity = 'normal' | play = 'yes']
Pr[humidity = 'normal' | play = 'no']
Pr[windy = 'false' | play = 'yes']
Pr[windy = 'false' | play = 'no']

Naïve Bayes Classifier

Naive Bayes Classifier

Attribute	Class yes (0.63)	no (0.38)
outlook		
sunny	3.0	4.0
overcast	5.0	1.0
rainy	4.0	3.0
[total]	12.0	8.0
temperature		
hot	3.0	3.0
mild	5.0	3.0
cool	4.0	2.0
[total]	12.0	8.0
humidity		
high	4.0	5.0
normal	7.0	2.0
[total]	11.0	7.0
windy		
TRUE	4.0	4.0
FALSE	7.0	3.0
[total]	11.0	7.0

Logistic Regression Function for the Numeric Weather Data

Class 0 :
5.57 +
[outlook=sunny] * -0.65 +
[outlook=overcast] * 2.82 +
[temperature] * -0.02 +
[humidity] * -0.06 +
[windy=FALSE] * 1.38

Class 1 :
-5.57 +
[outlook=sunny] * 0.65 +
[outlook=overcast] * -2.82 +
[temperature] * 0.02 +
[humidity] * 0.06 +
[windy=FALSE] * -1.38

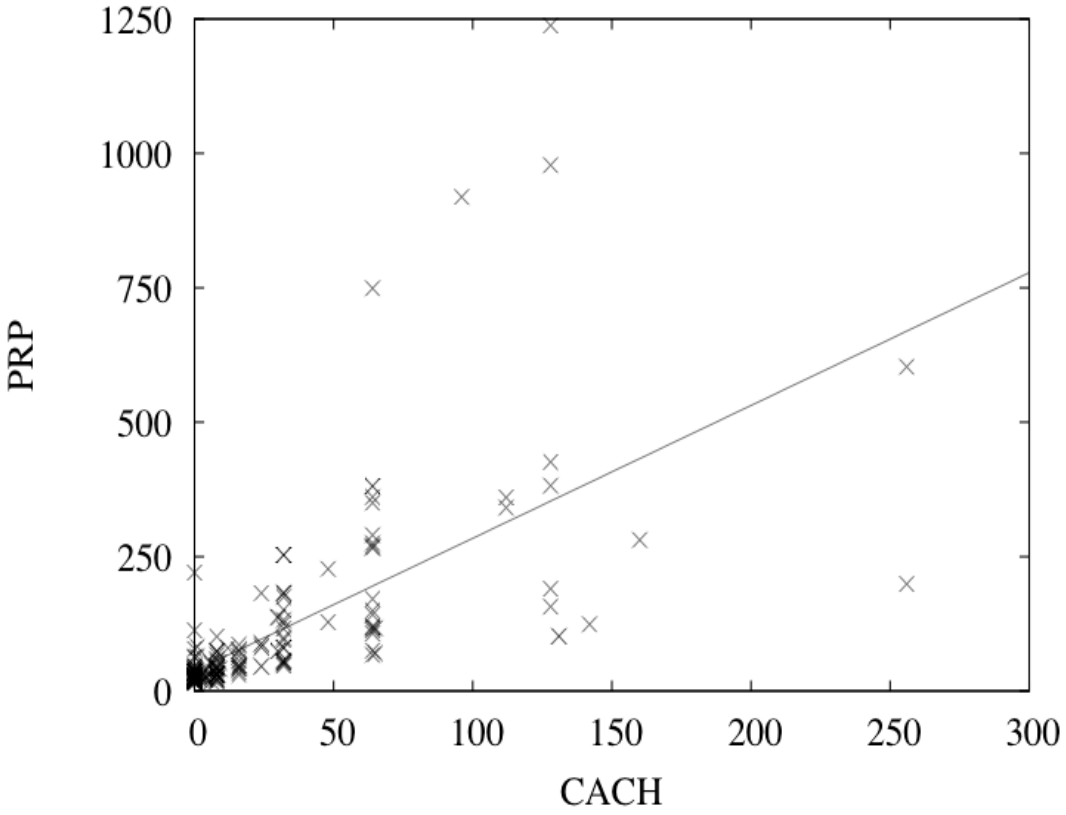
Class 0 is play='yes' Class 1 is play = 'no'

Linear Regression Function for the CPU Performance Data

class =

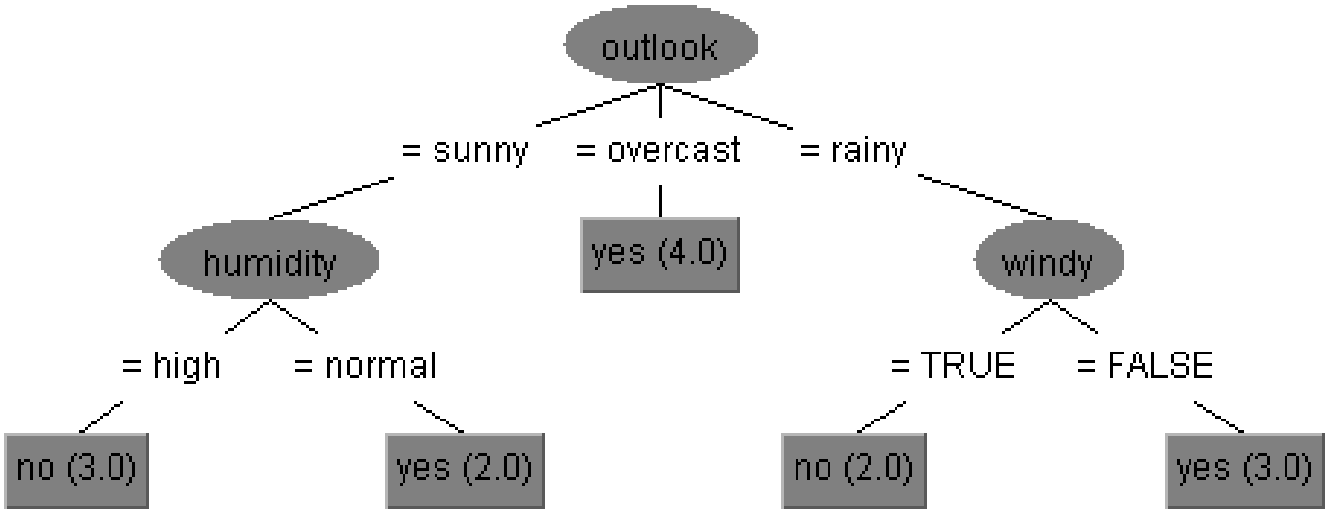
0.0491 * MYCT +
0.0152 * MMIN +
0.0056 * MMAX +
0.6298 * CACH +
1.4599 * CHMAX +
-56.075

A Linear Regression Function for the CPU Performance Data

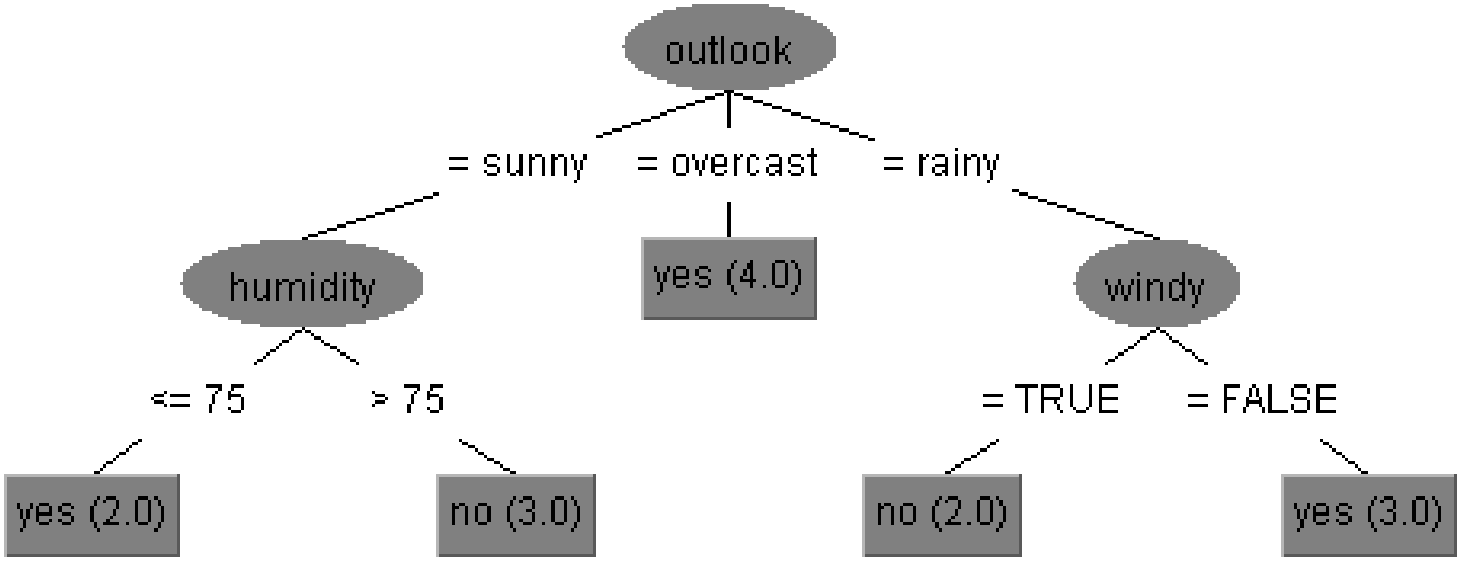


PRP = 37.06 + 2.47CACH

Decision Tree for Nominal Weather Data



Decision Tree for Numeric Weather Data



Classification Rules

Example:

If outlook = sunny and humidity =normal
then play = yes

Association Rules

Example:

If temperature=cool
then humidity=normal

Instance-Based Learning

- No structure is learned
- Given an instance to predict, simply predict the class of its nearest neighbor
- Alternatively, predict the class which appears most frequently for the nearest k neighbors

Clustering

Clustering techniques apply when there is no class to be predicted

Aim: divide instances into “natural” groups

As we've seen clusters can be:

- disjoint vs. overlapping
- deterministic vs. probabilistic
- flat vs. hierarchical