Handling Missing Data

So far: data always completely observed

"known x and y values for all data points"

Missing data

Credit	Term	Income	y
excellent	3yrs	high	safe
fair	?	low	risky
fair	3yrs	high	safe
poor	5yrs	high	risky
excellent	3yrs	low	risky
fair	5yrs	high	safe
poor	?	high	risky
poor	5yrs	low	safe
fair	?	high	safe

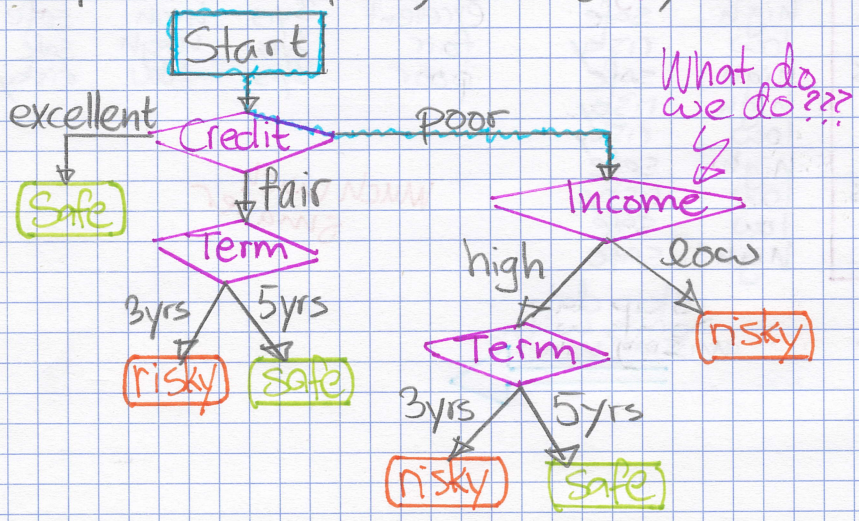
← Loan application may be 3 or 5 years

Missing values impact training and predictions

1. Training data: Contains "unknown" values
2. Predictions: Input at prediction time contains "unknown" values

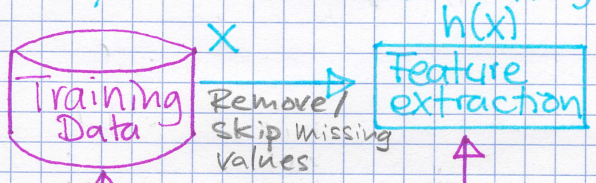
Missing values during prediction

x_i = (credit = poor, Income = ?, Term = 5 years)



Strategy 1: Purification by skipping missing data

Idea 1: Purification by skipping / removing



May contain data points where one (or more) features are missing

h(x) is a subset of the data without any missing values

skip data points with missing values →

X: N = 9, 3 features

Credit	Term	Income	y
excellent	3yrs	high	safe
⋮			

h(x): N = 6, 3 features

Credit	Term	Income	y
⋮			

The challenge with Idea 1

Warning: More than 50% of the loan Terms are unknown

X: N = 9, 3 features

Credit	Term	Income	y
excellent	3yrs	high	safe
fair	?	low	risky
fair	3yrs	high	safe
poor	?	high	risky
excellent	?	low	risky
fair	?	high	safe
poor	3yrs	low	risky
poor	?	low	safe
fair	?	high	safe

h(x): N = 3, 3 features

Credit	Term	Income	y
excellent	3yrs	high	safe
fair	3yrs	high	safe
poor	3yrs	low	risky

much much smaller

skip data points with missing values →

Idea 2: Skip features with missing values

Skip features with many missing values →

h(x)
N-9 2features

Credit	Income	y
excellent	high	safe
fair	low	risky
fair	high	safe
poor	high	risky
excellent	low	risky
fair	high	safe
poor	high	risky
poor	low	safe
fair	high	safe

Missing value skipping: Ideas 1 & 2

Idea 1: Skip data points where any feature contains a missing value
- Make sure only a few data points are skipped

Idea 2: Skip an entire feature if it's missing for many data points
- Make sure only a few features are skipped.

Missing value skipping: Pros and Cons

Pros

- Easy to understand and implement
- Can be applied to any model (decision trees, logistic regression, linear regression, ...)

Cons

- Removing data points and features may remove important information from data
- Unclear when it's better to remove data points versus features
- Doesn't help if data is missing at prediction time

Strategy 2: Purification by imputing missing data

Main drawback of skipping strategy

X — N=9, 3features

Data is precious don't throw it away!

h(x) — N=6, 3features

Credit	Term	Income	y
excellent	3yrs	high	safe
fair	?	low	risky
fair	3yrs	high	safe
poor	5yrs	high	risky
excellent	3yrs	low	risky
fair	5yrs	high	safe
poor	3yrs	low	risky
poor	?	low	safe
fair	?	high	safe

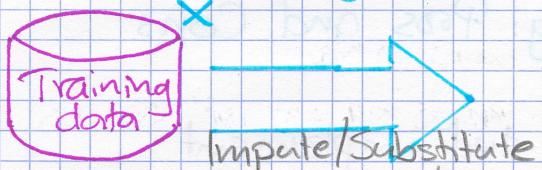
Skip data points with missing values

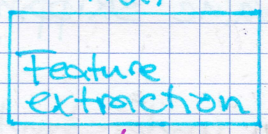
Credit	Term	Income	y
excellent	3yrs	high	safe
fair	3yrs	high	safe
poor	5yrs	high	risky
excellent	3yrs	low	risky
fair	5yrs	high	safe
poor	3yrs	low	risky

Can we keep all the data?

Idea 2: Purification by imputing

Use other data points in X to "guess" the "?"





Same number of data points as the original data

Example: Replace ? with most common value

#3 year loans: 4 ← Best guess

#5 year loans: 2

Purification by imputing

Credit	Term	Income	y
excellent	3yrs	high	safe
fair	3yrs	low	risky
fair	3yrs	high	safe
poor	5yrs	high	risky
excellent	3yrs	low	risky
fair	5yrs	high	safe
poor	3yrs	low	risky
poor	3yrs	low	safe
fair	3yrs	high	safe

Common (simple) rules for purification by imputation

Impute each feature with missing values:

1) Categorical features use mode: Most popular value (mode) of non-missing x_i
2) Numerical features use average or median: Average or median value of non-missing x_i

Many advanced methods exist, e.g. expectation-maximization (EM) algorithm

Missing Value Imputation: Pros and Cons

Pros

- Easy to understand and implement
- Can be applied to any model (decision trees, logistic regression, linear regression)
- Can be used at prediction time: use same imputation rules

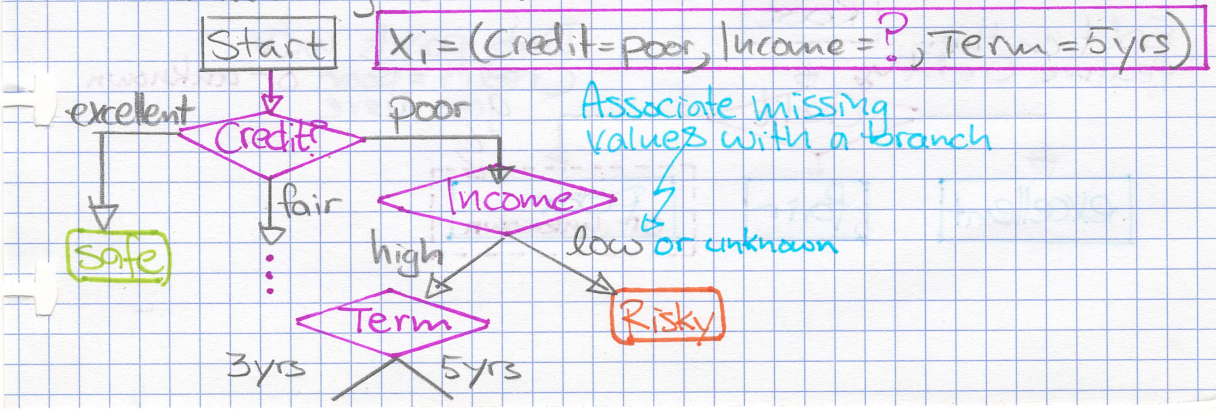
Cons

- May result in systematic errors

Example: Feature "age" missing in all banks in Washington by state law

Strategy 3: Adapt learning algorithm to be robust to missing values

Add missing values to the tree definition



Add missing value choice to every decision node

Explicitly handling missing data by learning algorithm: Pros and Cons

Pros

- Addresses training and prediction time
- More accurate predictions

Cons

- Requires modification of learning algorithm
  - Very simple for decision trees

Feature split selection with missing data

Greedy decision tree learning

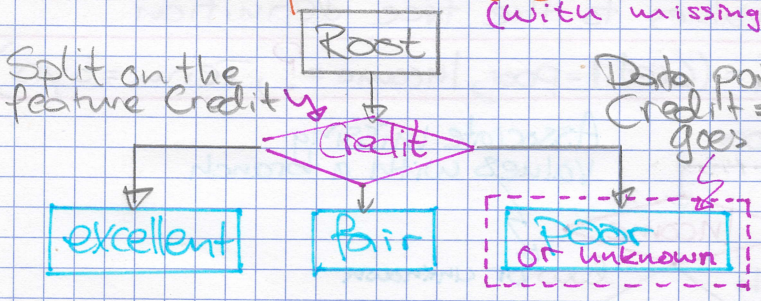
- Step1: Start with an empty tree
- Step2: Select a feature to split data
- For each split of the tree:
  - Step3: If nothing more to make predictions
  - Step4: Otherwise, go to step2 & continue (recurse) on this split

Pick feature split leading to lowest classification error

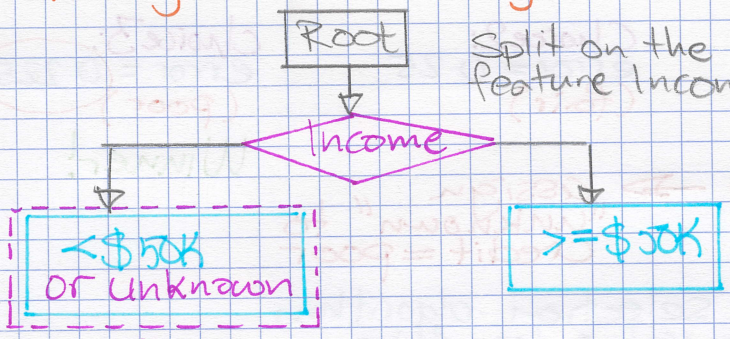
Must select feature & branch for missing values!

Feature split (without missing values)

(with missing values)



Missing value handling in threshold splits



Data points with Income <$30K or unknown

Should missing go left, right or middle?

Choose branch that leads to lowest classification error!

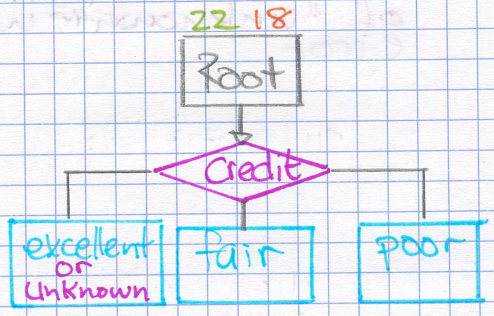
Choice1: Missing values go with Credit=excellent

Choice2: Missing values go with Credit=fair

Choice3: Missing values go with Credit=poor

Computing classification error of decision stump with missing data

Credit	Term	Income	y
excellent	3yrs	high	safe
?	5yrs	low	risky
fair	3yrs	high	safe
poor	5yrs	high	risky
?	3yrs	low	risky
?	5yrs	low	safe
poor	3yrs	high	risky
poor	5yrs	low	safe
fair	3yrs	high	safe
...	...	...	...



9 0 8 4 4 12

2

10 2 8 4 4 12

$$\text{Error} = \frac{2+4+4}{40} = 0.25$$

Observed values "Unknown"

Total

Use classification error to decide

Choice 1:
error = 0.25
(excellent)

Choice 2:
error = 0.25
(fair)

Choice 3:
error = 0.225
(poor)
Winner!

Best choice → assign "unknown" to credit = poor

Feature split selection algorithm with missing value handling

- Given a subset of data M (a node in a tree)
- For each feature $h_i(x)$:
  1) Split data points of M where $h_i(x)$ is not "unknown" according to feature $h_i(x)$
  2) Consider assigning data points with "unknown" value for $h_i(x)$ to each branch
     A) Compute classification error split & branch assignment of "unknown" values
- Chose feature $h^*(x)$ & branch assignment of "unknown" with lowest classification error