COMP5310: Principles of Data Science

W10: Decision trees

Presented by

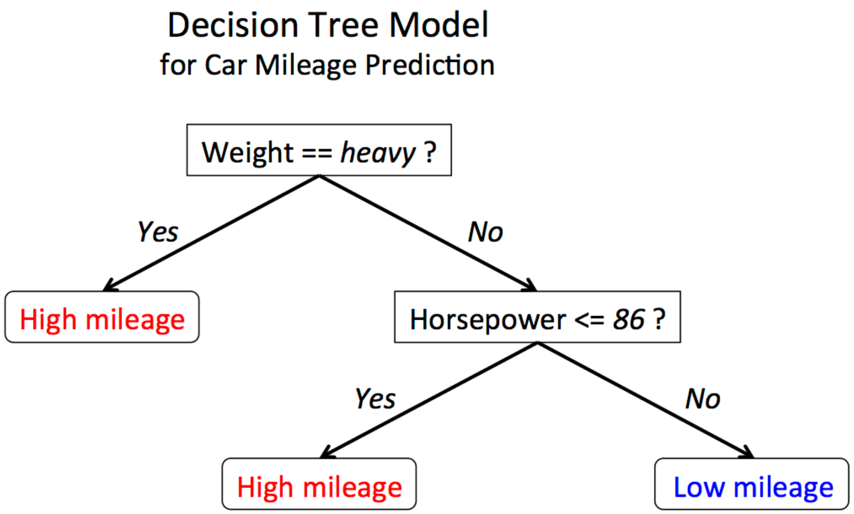
Ali Anaissi

School of Computer Science

THE UNIVERSITY OF SYDNEY

Decision trees

Decision tree classification



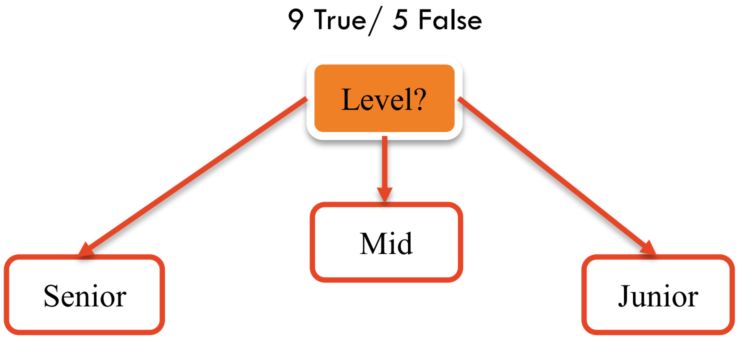
https://databricks.com/blog/2014/09/29/scalable-decision-trees-in-mllib.html

– Maps observations to a target value
– Can be viewed as hierarchy of if/else statements
– Resulting model is intuitive and interpretable
– Ensembles of simple trees can do very well

Algorithm for Decision Tree Induction

- Basic algorithm (a greedy ID3 algorithm)
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - Examples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., Information Gain (IG))
- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf

Predict if A15 belong to True or False

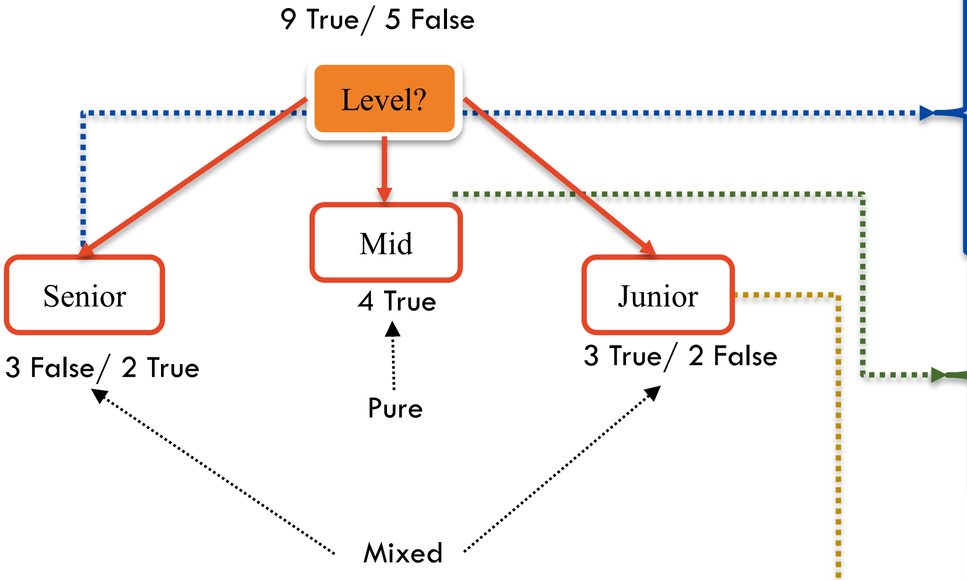


Training examples: 9 True/ 5 False

Class label

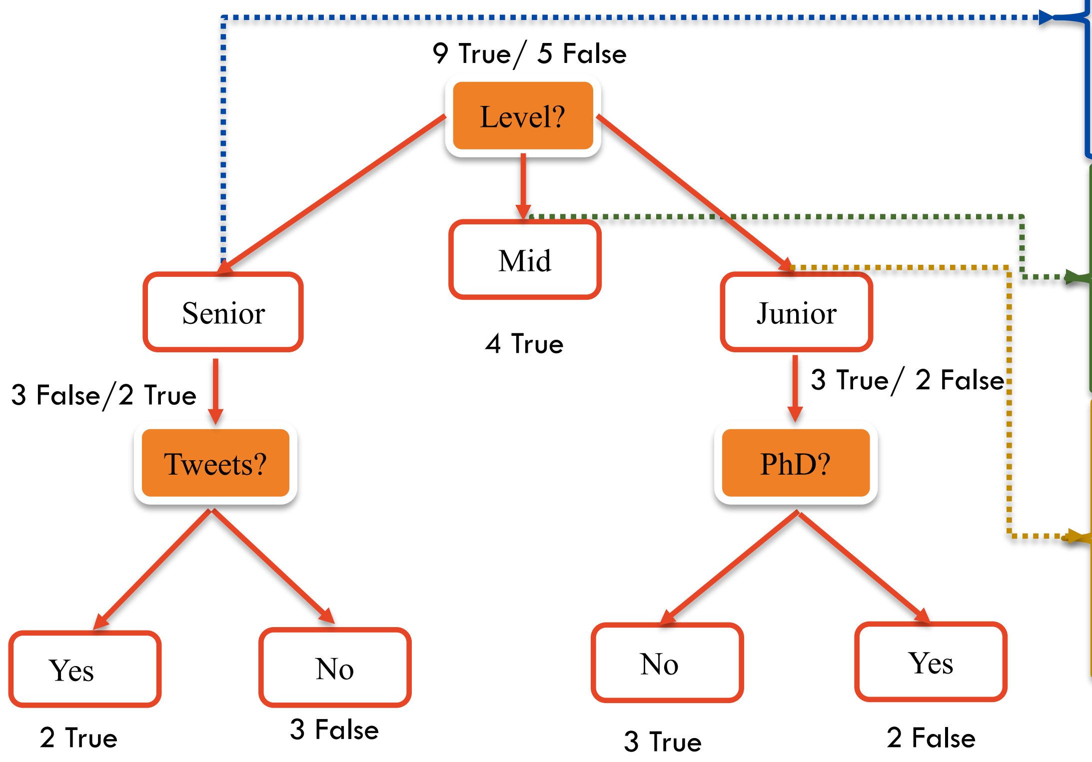
Applicant	Level	Lang	Tweets	PhD	Interviewed well
A1	Senior	Java	No	No	False
A2	Senior	Java	No	Yes	False
A3	Mid	Java	No	No	True
A4	Junior	Python	No	No	True
A5	Junior	R	Yes	No	True
A6	Junior	R	Yes	Yes	False
A7	Mid	R	Yes	Yes	True
A8	Senior	Python	No	No	False
A9	Senior	R	Yes	No	True
A10	Junior	Python	Yes	No	True
A11	Senior	Python	Yes	Yes	True
A12	Mid	Python	No	Yes	True
A13	Mid	Java	Yes	No	True
A14	Junior	Python	No	Yes	False
New data: A15	Senior	R	No	No	?

Decision Tree



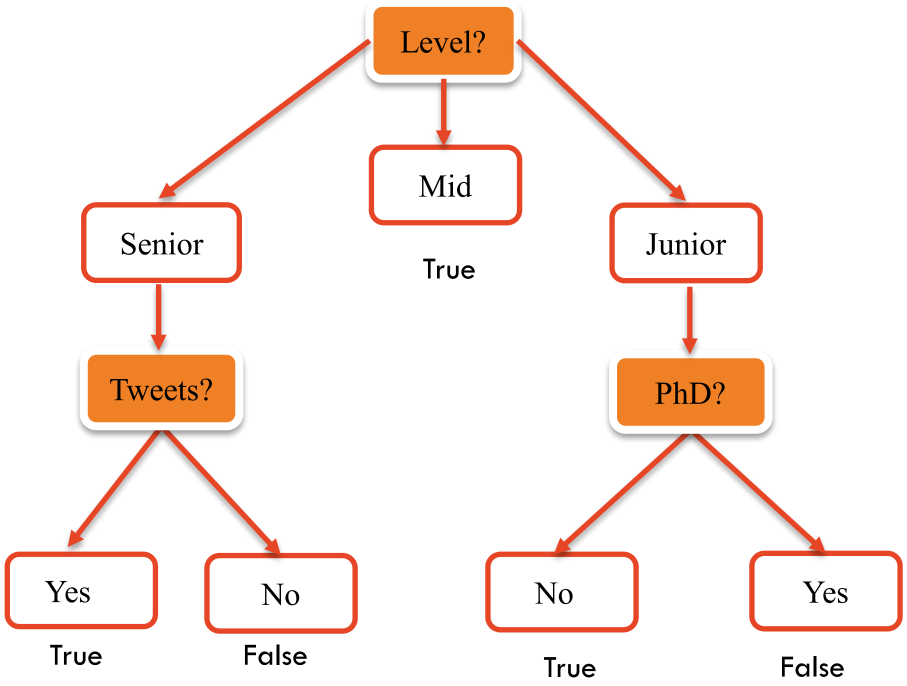
Level	Lang	Tweets	PhD	Interviewed well
Senior	Java	No	No	False
Senior	Java	No	Yes	False
Senior	Python	No	No	False
Senior	R	Yes	No	True
Senior	Python	Yes	Yes	True
Mid	Java	No	No	True
Mid	R	Yes	Yes	True
Mid	Python	No	Yes	True
Mid	Java	Yes	No	True
Junior	Python	No	No	True
Junior	R	Yes	No	True
Junior	Python	Yes	No	True
Junior	R	Yes	Yes	False
Junior	Python	No	Yes	False

Decision Tree



Level	Lang	Tweets	PhD	Interviewed well
Senior	Java	No	No	False
Senior	Java	No	Yes	False
Senior	Python	No	No	False
Senior	R	Yes	No	True
Senior	Python	Yes	Yes	True
Mid	Java	No	No	True
Mid	R	Yes	Yes	True
Mid	Python	No	Yes	True
Mid	Java	Yes	No	True
Junior	Python	No	No	True
Junior	R	Yes	No	True
Junior	Python	Yes	No	True
Junior	R	Yes	Yes	False
Junior	Python	No	Yes	False

The resulting tree:



Applicant	Level	Lang	Tweets	PhD	Interviewed well
A1	Senior	Java	No	No	False
A2	Senior	Java	No	Yes	False
A3	Mid	Java	No	No	True
A4	Junior	Python	No	No	True
A5	Junior	R	Yes	No	True
A6	Junior	R	Yes	Yes	False
A7	Mid	R	Yes	Yes	True
A8	Senior	Python	No	No	False
A9	Senior	R	Yes	No	True
A10	Junior	Python	Yes	No	True
A11	Senior	Python	Yes	Yes	True
A12	Mid	Python	No	Yes	True
A13	Mid	Java	Yes	No	True
A14	Junior	Python	No	Yes	False
A15	**Senior**	**R**	**No**	**No**	**False**

An Example

- Training data: interviewee data
- Four features:
  - Level , Lang, Tweets, PhD
- Class label:
  - Interviewed well
- I have new applicant A15 (Level, Lang, Tweets, PhD)
- Want to predict whether Interviewed well is True or False
- Hard to guess for A15!

Training examples: 9 True/ 5 False

Class label

Applicant	Level	Lang	Tweets	PhD	Interviewed well
A1	Senior	Java	No	No	False
A2	Senior	Java	No	Yes	False
A3	Mid	Java	No	No	True
A4	Junior	Python	No	No	True
A5	Junior	R	Yes	No	True
A6	Junior	R	Yes	Yes	False
A7	Mid	R	Yes	Yes	True
A8	Senior	Python	No	No	False
A9	Senior	R	Yes	No	True
A10	Junior	Python	Yes	No	True
A11	Senior	Python	Yes	Yes	True
A12	Mid	Python	No	Yes	True
A13	Mid	Java	Yes	No	True
A14	Junior	Python	No	Yes	False

New data:

Applicant	Level	Lang	Tweets	PhD	Interviewed well
A15	Senior	R	No	No	?

Predict if A15 belong to True or False

– Divide-and-conquer
  – Choose attributes to split the data into subsets
  – Are they pure?(all True or all False)
  – If yes: stop
  – If no: repeat
– Which attributes to choose?
  – Information Gain

Training examples: 9 True/ 5 False

Class label

Applicant	Level	Lang	Tweets	PhD	Interviewed well
A1	Senior	Java	No	No	False
A2	Senior	Java	No	Yes	False
A3	Mid	Java	No	No	True
A4	Junior	Python	No	No	True
A5	Junior	R	Yes	No	True
A6	Junior	R	Yes	Yes	False
A7	Mid	R	Yes	Yes	True
A8	Senior	Python	No	No	False
A9	Senior	R	Yes	No	True
A10	Junior	Python	Yes	No	True
A11	Senior	Python	Yes	Yes	True
A12	Mid	Python	No	Yes	True
A13	Mid	Java	Yes	No	True
A14	Junior	Python	No	Yes	False
New data: A15	Senior	R	No	No	?

Information Gain

Information Gain (IG)

– IG calculates effective change in entropy after making a decision based on the value of an attribute.

$$IG(Y|X) = H(Y) - H(Y|X)$$

where

Y is a class label

X is an attribute

$H(Y)$ *is the entropy of* Y

$H(Y|X)$ *is the* conditional entropy of Y given X

Entropy

– To measure the uncertainty associated with data

$H(Y) = -\sum_{i=i}^{m} p_i \log_2 (p_i)$, where p_i = P(Y =y_i) and

m is the number of classes

– Interpretation:
  – Higher entropy => higher uncertainty
  – Lower entropy => lower uncertainty

– Example, I have input X and want to predict Y

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

P(Y = Yes) P(Y = No)

– $H(Y) = -(0.5*\log_2 (0.5) + 0.5*\log_2 (0.5)) = 1$

Compare with: H(Y) = -(1.0*log2 (1.0) + 1.0*log2 (1.0)) = 0

Specific Conditional Entropy $H(Y|X=v)$

X	Y
Math	Yes
Math	No
Math	No
Math	Yes

X	Y
History	No
History	No

X	Y
CS	Yes
CS	Yes

– $H(Y|X=v)$ = entropy of Y among only those records in which X has value v

P(Y = Yes)

P(Y = No)

$H(Y|X=Math) = -(0.5*\log_2(0.5) + 0.5*\log_2(0.5)) = 1$

$H(Y|X=History) = 0$

$H(Y|X=CS) = 0$

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Conditional Entropy $H(Y|X)$

- H(Y | X) = the average conditional entropy of Y

$$H(Y|X) = \sum_i p(X = vi)\ H(Y|X = vi)$$

- From data we estimate

P(X=Math) = 4/8 = 0.5

vi	$p(X = vi)$	$H(Y\|X = vi)$
Math	0.5	?
History	0.25	?
CS	0.25	?

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Conditional Entropy $H(Y|X)$

– H(Y | X) = the average conditional entropy of Y

$$H(Y|X) = \sum_i p(X = vi)\ H(Y|X = vi)$$

vi	$p(X = vi)$	$H(Y\|X = vi)$
Math	0.5	1
History	0.25	0
CS	0.25	0

H(Y | X) = 0.5*1+0.25*0+0.25*0 = 0.5

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Information Gain (IG)

$$IG(Y|X) = H(Y) - H(Y|X)$$

- Example:

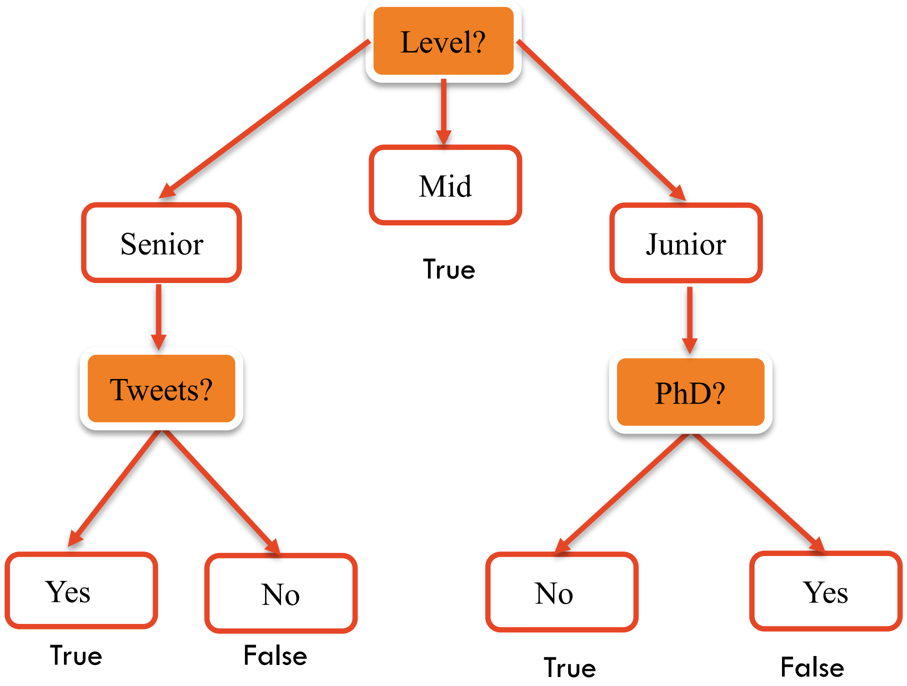
  H(Y) = 1

  H(Y|X) = 0.5

  Thus:

  IG(Y|X) = 1 – 0.5 = 0.5

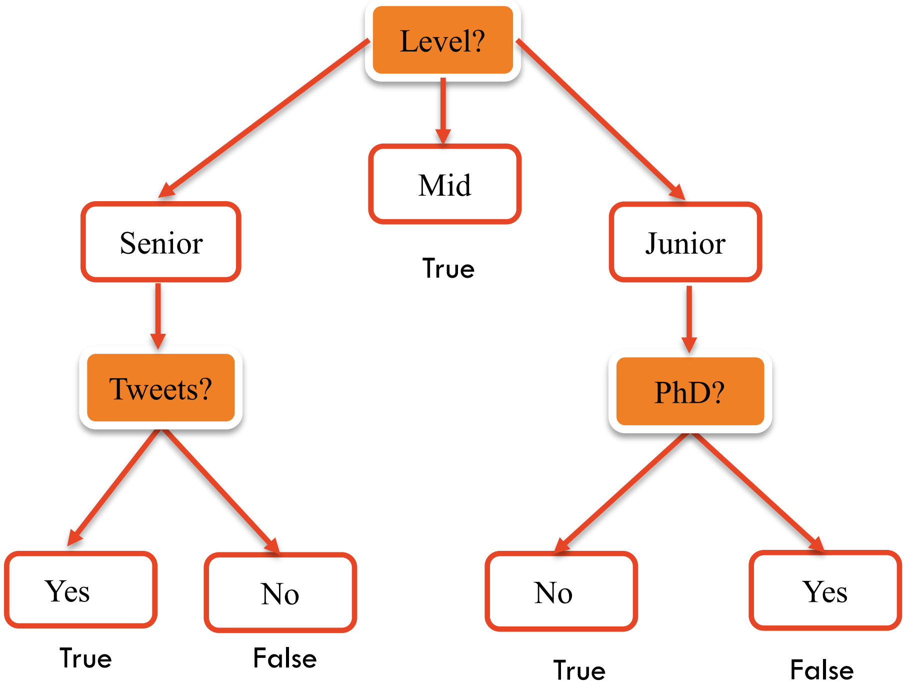
Our previous tree:



Applicant	Level	Lang	Tweets	PhD	Interviewed well
A1	Senior	Java	No	No	False
A2	Senior	Java	No	Yes	False
A3	Mid	Java	No	No	True
A4	Junior	Python	No	No	True
A5	Junior	R	Yes	No	True
A6	Junior	R	Yes	Yes	False
A7	Mid	R	Yes	Yes	True
A8	Senior	Python	No	No	False
A9	Senior	R	Yes	No	True
A10	Junior	Python	Yes	No	True
A11	Senior	Python	Yes	Yes	True
A12	Mid	Python	No	Yes	True
A13	Mid	Java	Yes	No	True
A14	Junior	Python	No	Yes	False

Is my decision tree correct?

- Let's check whether the split on Level attribute is correct.
- We need to show that Level attribute has the highest information gain.



$H(Y) = -\sum_{i=i}^{m} p_i \log_2 (p_i)$, where p_i = P(Y $=y_i$) and m is the number of classes

$$H(Y|X) = \sum_i p(X = vi)\ H(Y|X = vi)$$

$$IG(Y|X) = H(Y) - H(Y|X)$$

Applicant	Level	Lang	Tweets	PhD	Interviewed well
A1	Senior	Java	No	No	False
A2	Senior	Java	No	Yes	False
A3	Mid	Java	No	No	True
A4	Junior	Python	No	No	True
A5	Junior	R	Yes	No	True
A6	Junior	R	Yes	Yes	False
A7	Mid	R	Yes	Yes	True
A8	Senior	Python	No	No	False
A9	Senior	R	Yes	No	True
A10	Junior	Python	Yes	No	True
A11	Senior	Python	Yes	Yes	True
A12	Mid	Python	No	Yes	True
A13	Mid	Java	Yes	No	True
A14	Junior	Python	No	Yes	False

Calculation

P(Y = True)

P(Y = False)

– H(Interviewed)=H(9,5)= $-\frac{9}{14}log_2(\frac{9}{14})$ - $\frac{5}{14}log_2(\frac{5}{14})$ = 0.94

– H (Interviewed$|_{X=level}$) $= \frac{5}{14}\ H(2,3) + \frac{4}{14}\ H(4,0) + \frac{5}{14}\ H(3,2)$ = 0.7

$level$	$p(X = level)$	$H(Y\|X = level)$
Senior	0.365	$H(2,3)$ =0.971
Mid	0.27	$H(4,0)$=0
Junior	0.365	$H(3,2)$ = 0.971

– IG (level) = H(Interviewed) - H (Interviewed$|_{X=level}$) = **0.24**

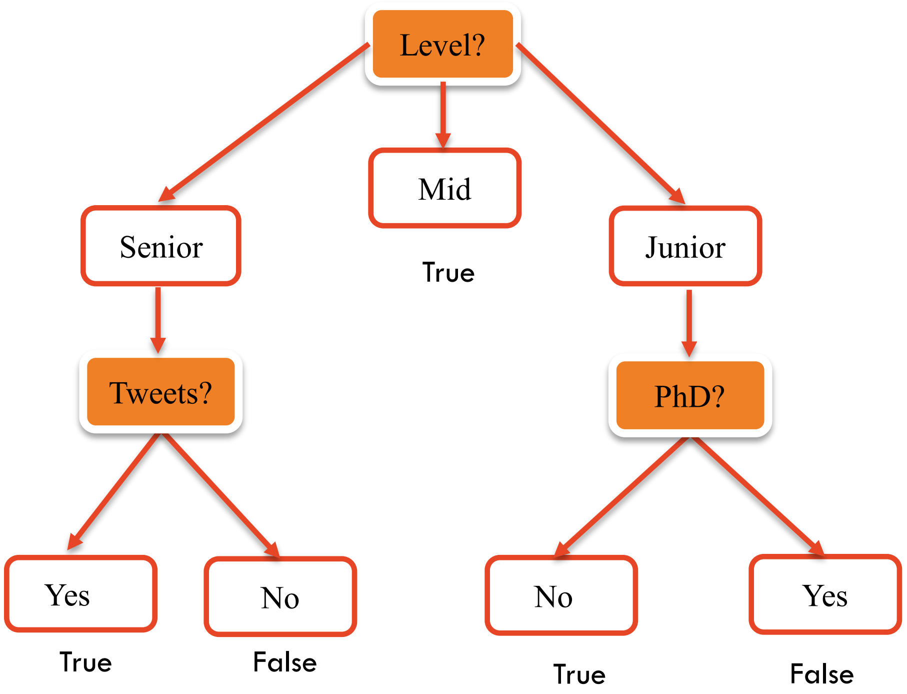
– IG (tweets) = H(Interviewed) - H (Interviewed$|_{X=tweets}$) = 0.15

– IG (PhD) = H(Interviewed) - H (Interviewed$|_{X=PhD}$) = 0.048

– IG (lang) = H(Interviewed) - H (Interviewed$|_{X=lang}$)= 0.029

Is my decision tree correct?

- Let's also check whether the split on PhD attribute is correct.
- We need to show that PhD attribute has the highest information gain.



PhD attribute – subset of 5 records with Junior level

	Level	Lang	Tweets	PhD	Interviewed well
1	Senior	Java	No	No	False
2	Senior	Java	No	Yes	False
3	Mid	Python	No	No	True
4	Junior	Python	No	No	True
5	Junior	R	Yes	No	True
6	Junior	R	Yes	Yes	False
7	Mid	R	Yes	Yes	True
8	Senior	Python	No	No	False
9	Senior	R	Yes	No	True
10	Junior	Python	Yes	No	True
11	Senior	Python	Yes	Yes	True
12	Mid	Python	No	Yes	True
13	Mid	Java	Yes	No	True
14	Junior	Python	No	Yes	False

	Level	Lang	Tweets	PhD	Interviewed well
4	Junior	Python	No	No	True
5	Junior	R	Yes	No	True
6	Junior	R	Yes	Yes	False
10	Junior	Python	Yes	No	True
14	Junior	Python	No	Yes	False

Calculation

Entropy:

- H(Interviewed)=H(3,2)= $-\frac{3}{5}\,log_2(\frac{3}{5})$ - $\frac{2}{5}\,log_2(\frac{2}{5})$ = 0.971
- $H\,(\text{Interviewed}|_{X=PhD}) = \frac{2}{5}\ H(2,0) + \frac{3}{5}\ H(0,3) = 0$

	Level	Lang	Tweets	PhD	Interviewed well
4	Junior	Python	No	No	True
5	Junior	R	Yes	No	True
6	Junior	R	Yes	Yes	False
10	Junior	Python	Yes	No	True
14	Junior	Python	No	Yes	False

PhD	$p(X = \boldsymbol{PhD})$	$H(Y\|X = \boldsymbol{PhD})$
Yes	0.4	$H(2,0)$ =0
No	0.6	$H(0,3)$=0

Calculation

- Information Gain:
  - IG (PhD) = H(Interviewed) - H (Interviewed$|_{X=PhD}$) = **0.971**
  - IG (Tweets) = H(Interviewed) - H (Interviewed$|_{X=Tweets}$) = 0.01997
  - IG (Lang) = H(Interviewed) - H (Interviewed$|_{X=lang}$)= 0.01997

	Level	Lang	Tweets	PhD	Interviewed well
4	Junior	Python	No	No	True
5	Junior	R	Yes	No	True
6	Junior	R	Yes	Yes	False
10	Junior	Python	Yes	No	True
14	Junior	Python	No	Yes	False

Train a decision tree classifier in scikit-learn

```
from sklearn.tree import DecisionTreeClassifier

# Let's fit a model
tree = DecisionTreeClassifier(max_depth=2)
_ = tree.fit(X_train, Y_train)
```

Some decision tree parameters in scikit-learn

- max_depth
  - the maximum depth of the tree
- criterion
  - entropy: choose splits that minimise total uncertainty
  - gini: choose splits that minimise misclassification
- splitter
  - best: choose the optimal threshold for each feature
  - random: choose the best random threshold for each feature

Exercise: Decision trees

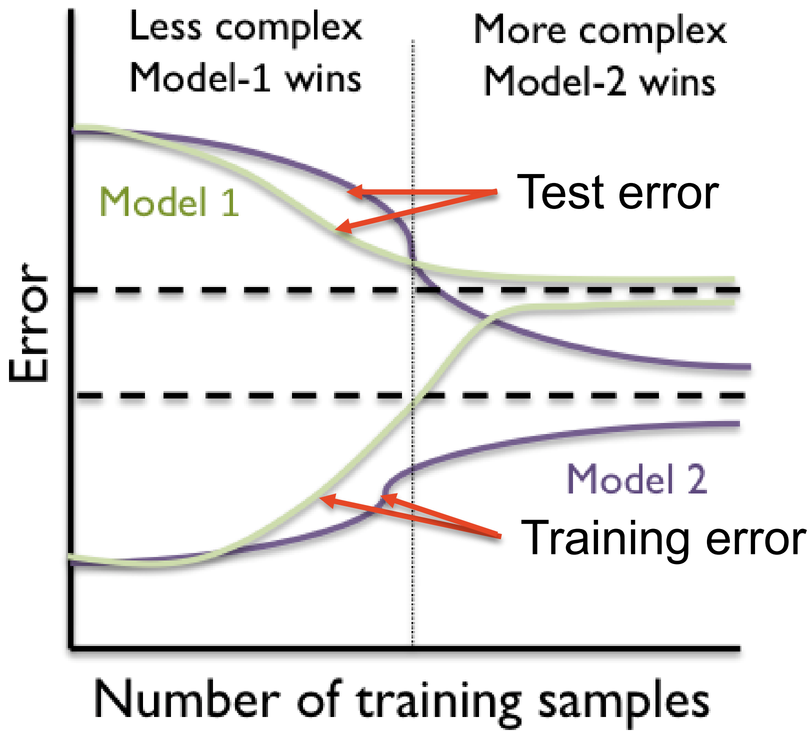
- Decision trees in scikit-learn
  -  code cell after “Train and view a tree”
  -  code cell after “McNemar’s test”
- Comparing classifiers
  - Which classifier is better?
  - Is the difference reliable?

Evaluation setup

Setting up a reliable evaluation

- Aim is to create an experiment setup that
  - Is fair for approaches/participants
  - Prevents overfitting
  - Allows reliable comparison

Model choice depends on amount of data available

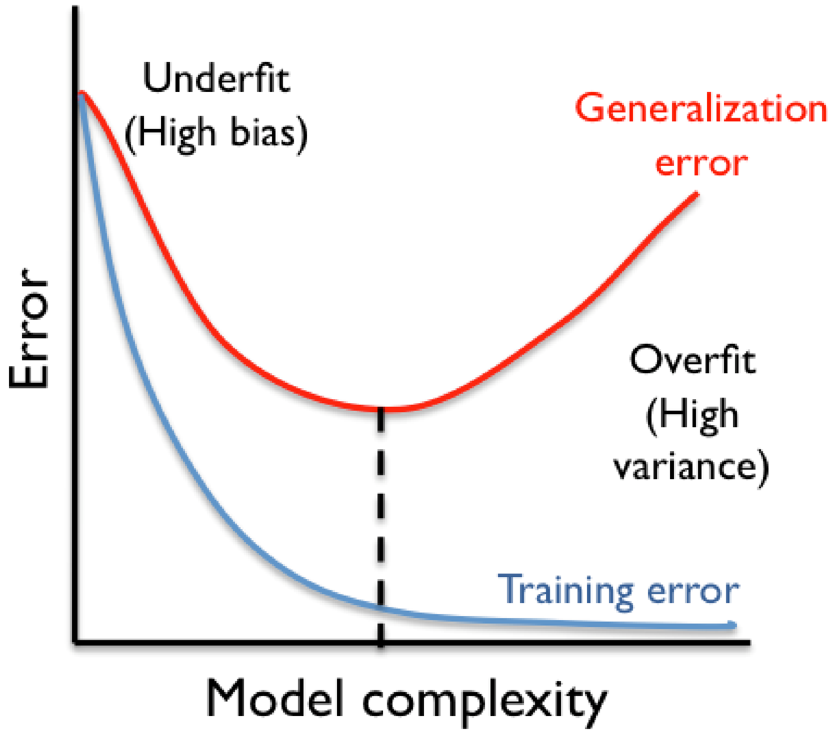


- Training error increases
- Test error decreases
- Two converge to asymptote
- If the amount of training data available is less than a certain threshold, then the less complex model 1 wins.
- If we can get more data, model 2 eventually wins
- Neither model will improve much with more data than we already have

https://thebayesianobserver.wordpress.com/2012/02/07/debugging-machine-learning-algorithms/

Finding a model that generalizes

- The dashed line on right shows point where we switch form under-fitting to overfitting
- Goal: Find this dotted line
- Generalization error should model application as closely and reliably as possible
  - Sample must be representative
  - Larger sample better



https://thebayesianobserver.wordpress.com/2012/02/07/debugging-machine-learning-algorithms/

Data drift (non-stationary data)

What it is:

– Typical train/test setups assume stationarity
– Should be near-true for train and test samples
– Only near-true in production for a little while

What to do:

– Monitor offline metric on live data
– May require monitoring/annotation
– If there are large changes, then retrain on new data
– Online/incremental learning

Exercise: model selection on test data

- Grid search using cross validation vs test data
  - code cell under “Grid search against test data”
  - code cell under “Grid search with cross validation”
  - Which result should we prefer?

Building a good solution

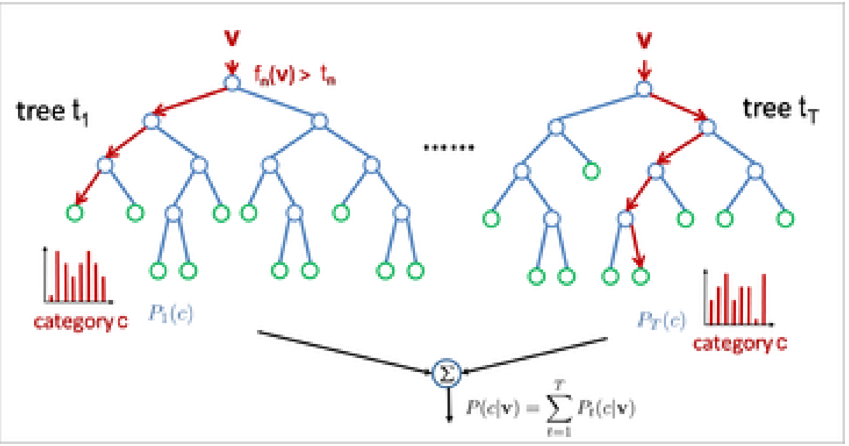
Build a simple model first, evaluate, iterate

- Start by building an end-to-end pipeline and evaluation
- Replicate published benchmarks to sanity check pipeline
- Wash, rinse, repeat:
  - Review the data and problem
  - Hypothesize next best approach in terms of elegance and impact
  - Implement and evaluate approach

Feature engineering is often key

- Relates back to understanding the problem
- Design informative and discriminative features
- Understand and validate features to avoid overfitting
  - Beware if a model weights a feature more than makes sense

Ensembles of predictors often do very well



http://www.iis.ee.ic.ac.uk/icvl/iccv09_tutorial.html

- Vote across many classifiers
- Random forest
  - Bootstrap many trees on samples of training data
  - Become more biased
  - But lower variance
- Lose explainability of trees!
- Generally boosts the performance of the final model

Exercise: Ensembling classifiers

- Decision trees can overfit
  - code cell under "Load and split data
  - code cell under "Plot error vs complexity for decision tree"
  - Assessing fit and checking for overfitting
- Ensembles of decision trees
  - code cell under "Plot error vs complexity for random forest"
  - code cell under "Plot error vs number of training samples"
  - Compare fit and assess data needed

Communicating results

Telling a story

- Construct a **narrative** around the importance of the **problem**
- Briefly explain technical approach (the **solution**)
- Describe results focusing on **impact** and **caveats**

Construct a narrative around the problem

- It should be absolutely clear why the problem matters
- How are you framing the problem in terms of (a) specific research question(s)?
- How will you validate the success of your proposed solution?

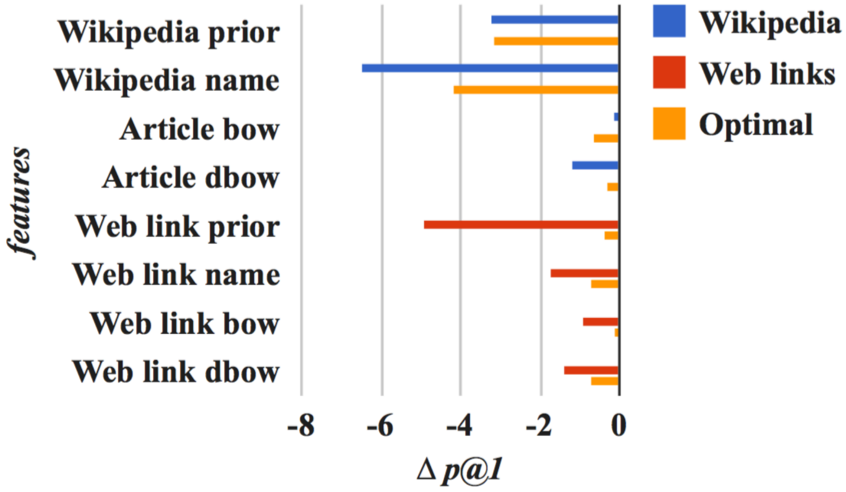
Reporting accuracy and reliability

- Understand the problem and the data
  - Report annotation process and agreement
  - Confusion matrices to assess less frequent categories
  - Report human upper bound as a benchmark where possible
  - http://www.mitpressjournals.org/doi/pdf/10.1162/089120102762671936
- Report simplest reasonable model as a benchmark (baseline)
- Report accuracy numbers with reliability, e.g.:
  - Pairwise significance tests to compare to benchmarks
  - Confidence intervals
  - Training versus generalization performance

Error analysis

- Error analysis seeks to identify systematic problems, e.g.:
  - Sample 20 false positives and 20 false negatives
  - Look at feature vectors and corresponding data
  - Group errors into categories and count
- Requires manual inspection but provides qualitative insight
- Should not be overlooked in favour of parameter tweaking
- Confusion matrices can also help to identify common errors

Subtractive feature analysis



- Assess impact of each feature by removing it
- The more performance goes down, the more critical
- If performance goes up, it's not a good feature

http://www.aclweb.org/anthology/Q15-1011

Deploying machine learning

- Remember the goal is a practical and usable solution
- It does no good to solve a problem if it can’t be deployed
- Some things to keep in mind:
  - Efficiency
  - Reliability of code
  - Monitoring drift

Review

Additional reading (not examinable)

- Stanford ML class lecture on regularization and overfitting
  https://class.coursera.org/ml-003/lecture/39
- A tutorial for learning data science with Python
  http://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-learn-data-science-python-scratch-2/
- Slides on cross validation and bootstrap
  https://lagunita.stanford.edu/c4x/HumanitiesScience/StatLearning/asset/cv_boot.pdf

On good data science

- How to evaluate machine learning modelshttp://blog.dato.com/how-to-evaluate-machine-learning-models-the-pitfalls-of-ab-testing
- Top 10 data science practitioner pitfalls http://www.slideshare.net/0xdata/top-10-data-science-practitioner-pitfalls
- Introduction to Applied Machine Learning: Generalisation http://www.inf.ed.ac.uk/teaching/courses/iaml/slides/eval-2x2.pdf

Next Time

Next week: Unstructured data

Objective

Learn to set up, explain and maintain machine learning tools.

Lecture

– Naïve Bayes
– Text-driven forecasting
– Structured prediction

Readings

– Doing Data Science, Ch. 7, 11, 13

Exercises

– Spam detection
– Predicting box office returns
– Information extraction

TODO in W11

– Analyze and characterize results

Project Stage 2

Suggested timeline for project stage 2

- W7: Define experimental framework
- W8: Implement approach
- *W9: Write first page (framework, approach)*
- **W10: Evaluate and benchmark approach**
- W11: Analyze and characterize results
- W12: Submit full report (W9 + results, analysis, conclusions)
  W12: Deliver presentation