COMP5310: Principles of Data Science

W10: Decision Tree

Presented by

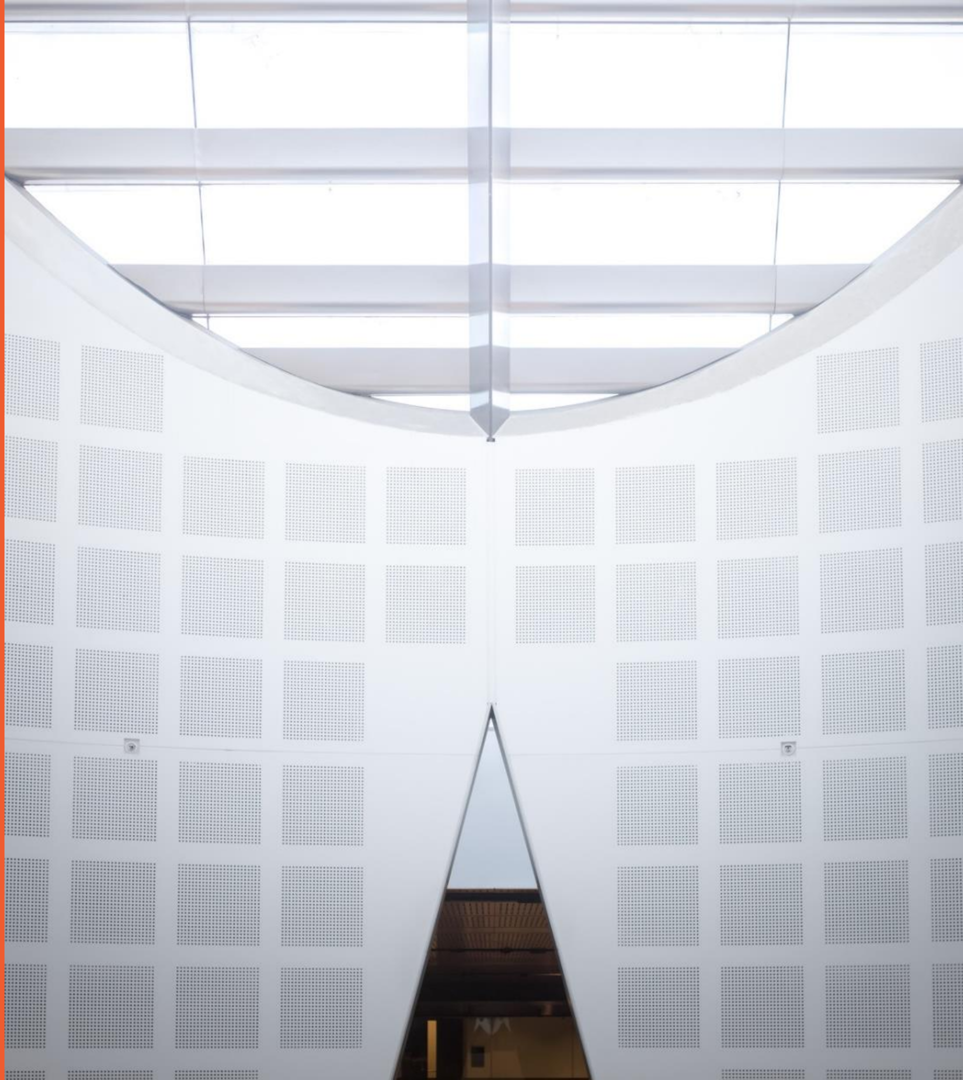
Maryam Khanian

School of Computer Science

Based on slides by previous lecturers of this unit of study

THE UNIVERSITY OF SYDNEY

The University of Sydney



Last week: Linear regression & logistic regression

Objective

Learn techniques for supervised machine learning, with tools in Python.

Lecture

- Simple linear regression
- Multiple linear regression
- Gradient descent
- Logistic regression

Readings

- Data Science from Scratch, Ch. 8, 14, 15, 16

Exercises

- sklearn: regression

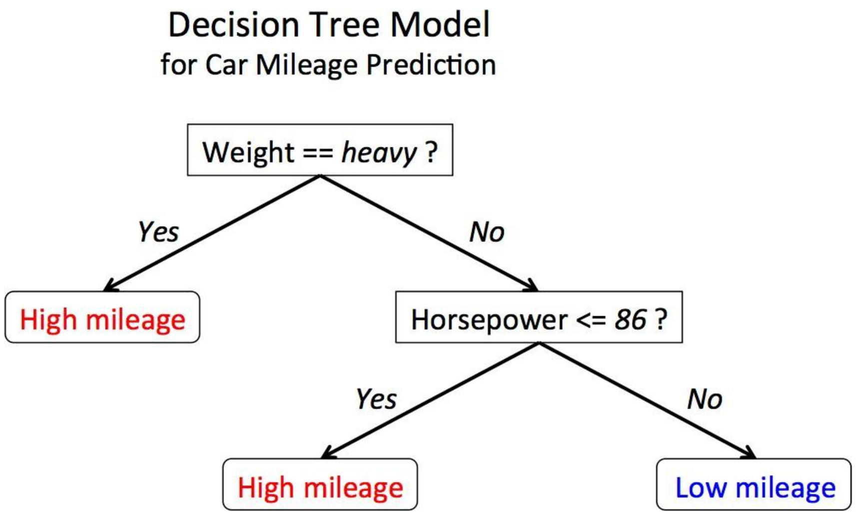
Supervised Learning:

- We'll now focus on supervised machine learning techniques
  - ☑ *Simple linear regression*
  - ☑ *Multiple linear regression*
  - ☑ *Logistic regression*
  - **Decision tree**
  - Naïve Bayes

DECISION TREE

Decision tree classification

- Maps observations to a target value by asking a series of questions
- Can be viewed as hierarchy of if/else statements.
  - Each **non-leaf** node corresponds to a test for the values of an attribute
- Resulting model is **intuitive** and **interpretable.**
- Ensembles of simple trees can do very well.



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

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.

Example

- **Training data:** interviewee data.
- **Features:** Level , Lang, Tweets, PhD.
- **Class label:** Interviewed well.
- **New applicant:** A15 (Senior, R, No, No).
- We want to **predict whether A15 Interviewed well or not**

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

- Divide-and-conquer:
  - Choose attributes to split the data into subsets
  - Are they pure?(all True or all False)
    - If yes: stop
    - Otherwise: repeat
- Which attributes to choose?
- Let's try selecting "Level" attribute first.

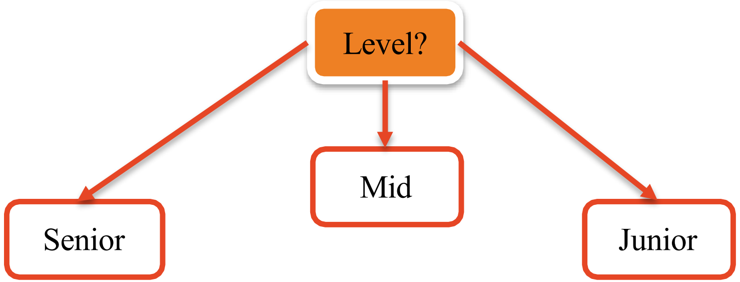
Training examples: 9 True/ 5 False

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

9 True/ 5 False



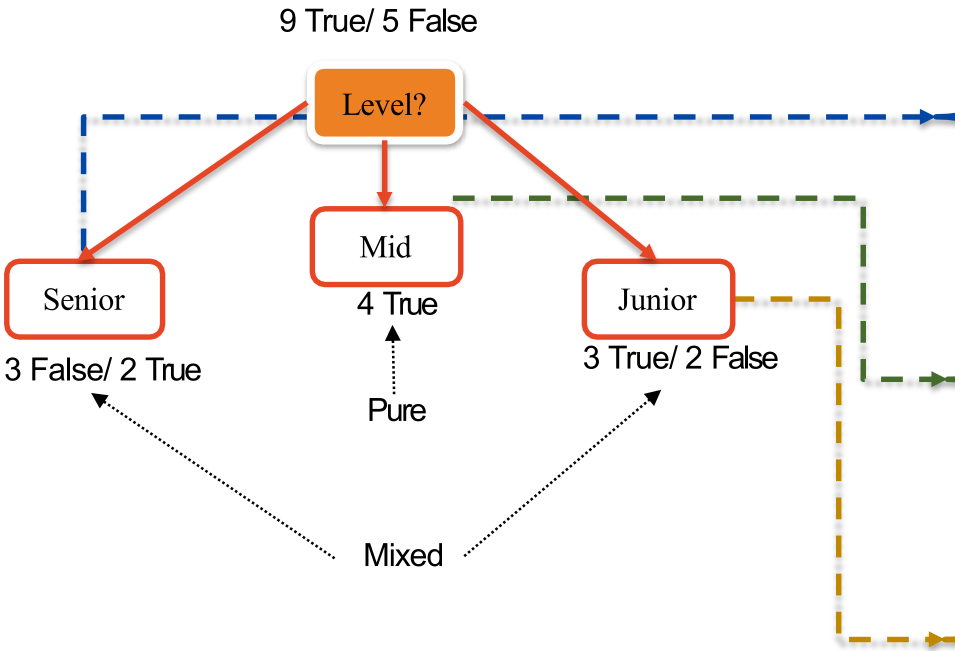
Training examples: 9 True/ 5 False

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
A15	**Senior**	**R**	**No**	**No**	**?**

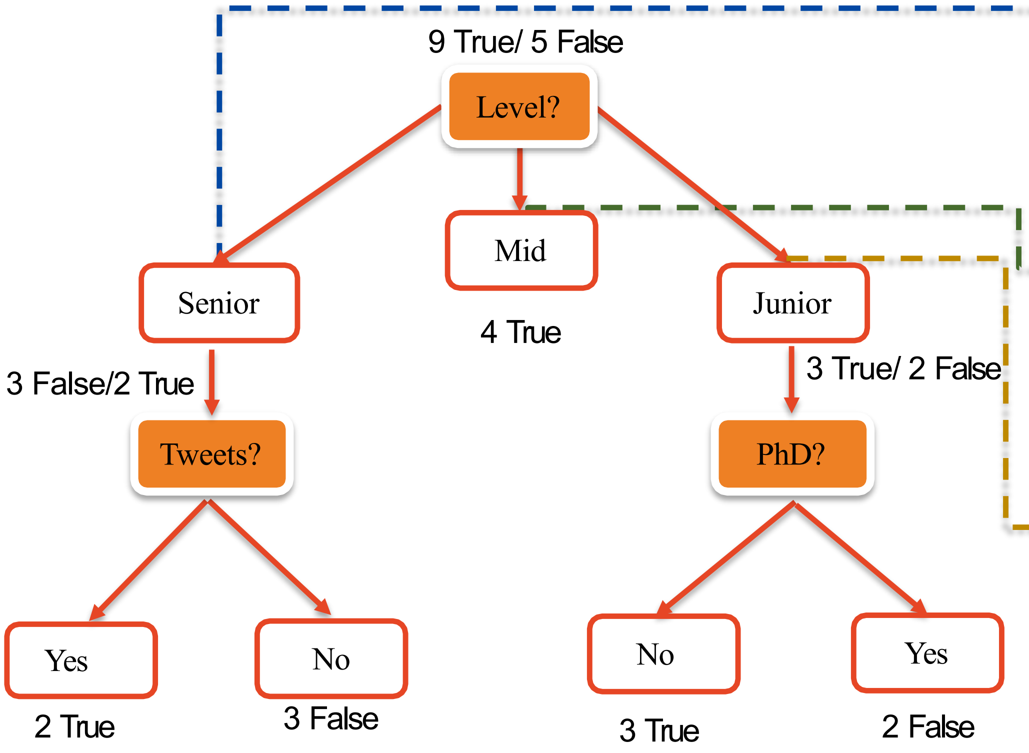
New data: A15

Decision Tree



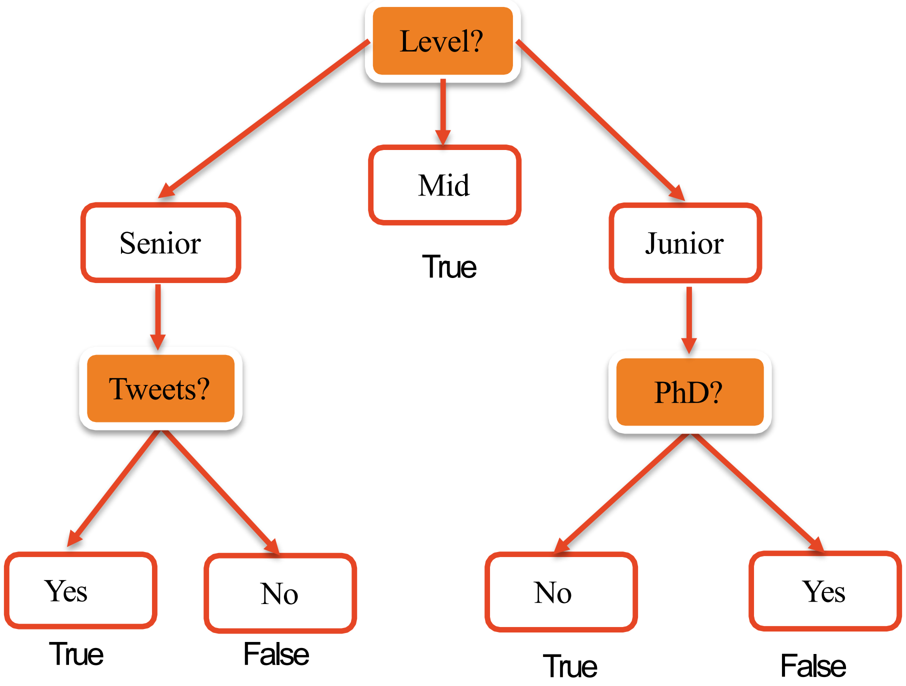
Level	Lang	Tweets	PhD	Interviewed well
Senior	Java	No	No	False
Senior	Java	No	Yes	False
Senior	Python	No	No	False
Senior	R	Yes	No	True
Senior	Python	Yes	Yes	True
Mid	Java	No	No	True
Mid	R	Yes	Yes	True
Mid	Python	No	Yes	True
Mid	Java	Yes	No	True
Junior	Python	No	No	True
Junior	R	Yes	No	True
Junior	Python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

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

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:

$$H(Y) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- Where $p_i = p(Y = y_i)$, and m is the number of classes.
- Interpretation:
  - Higher entropy => higher uncertainty.
  - Lower entropy => lower uncertainty.
- **Example:** We have input X and want to predict Y:
  - $H(Y) = -(\underbrace{0.5 * \log_2(0.5)}_{P(Y = Yes)} + \underbrace{0.5 * \log_2(0.5)}_{P(Y = No)}) = 1$

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.

$$H(Y|X) = \sum_i p(X = v_i) * H(Y|X = v_i)$$

- From data, we calculate $p(X = v_i)$:
  - $p(X = \text{Math}) = {}^4/_8 = 0.5$
  - $p(X = \text{History}) = {}^2/_8 = 0.25$
  - $p(X = \text{CS}) = {}^2/_8 = 0.25$

v_i	$p(X = v_i)$	$H(Y\|X = v_i)$
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

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

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

X	Y
Math	Yes
Math	No
Math	No
Math	Yes

X	Y
History	No
History	No

X	Y
CS	Yes
CS	Yes

– From the data, we obtain:

- $H(Y|X=\text{Math}) = -(2/_4 * \log_2(2/_4) + 2/_4 * \log_2(2/_4)) = 1$
- $H(Y|X=\text{History}) = -(0/_2 * \log_2(0/_2) +* 2/_2 \log_2(2/_2)) = 0$
- $H(Y|X=\text{CS}) = -(2/_2 * \log_2(2/_2) + 0/_2 * \log_2(0/_2)) = 0$

Conditional Entropy: H(Y | X)

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

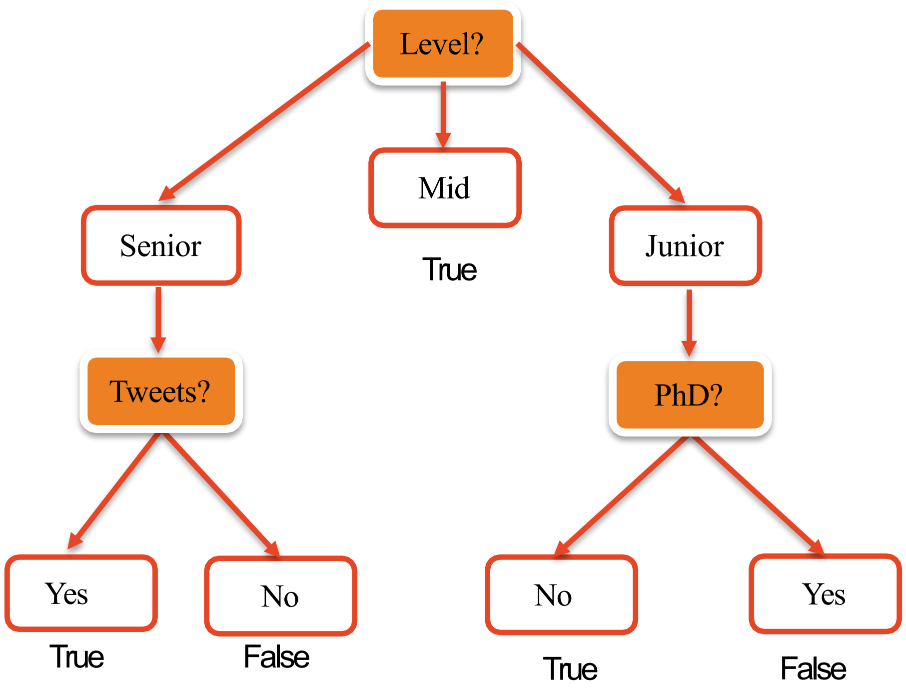
$$H(Y|X) = \sum_i p(X = v_i) * H(Y|X = v_i)$$
$$= 0.5 * 1 + 0.25 * 0 + 0.25 * 0$$
$$= 0.5$$

Information Gain (IG)

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

- From the example:
  - $H(Y) = 1$
  - $H(Y|X) = 0.5$
- Thus:
  - $IG(Y|X) = 1 - 0.5 = 0.5$

Is the previous decision tree good?



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

Calculation

- $H(\text{Interviewed well}) = H(9{,}5) = -\left({}^{9}/_{14}\log_2({}^{9}/_{14}) + {}^{5}/_{14}\log_2({}^{5}/_{14})\right) = 0.94$
- $H(\text{Interviewed well} \mid \text{Level}) = \sum_i p(\text{Level} = v_i) * H(\text{Interviewed well} \mid \text{Level} = v_i)$

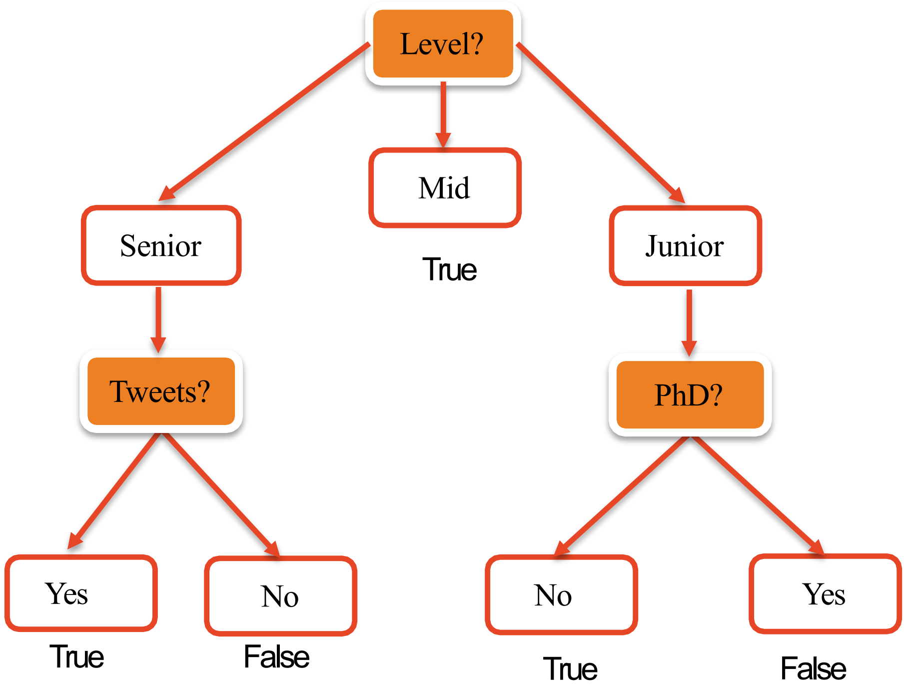
v_i	$p(\text{Level} = v_i)$	$H(\text{Interviewed well} \mid \text{Level} = v_i)$
Senior	${}^{5}/_{14} = 0.36$	$H(2{,}3) = -\left({}^{2}/_{5} * \log_2\left({}^{2}/_{5}\right) + {}^{3}/_{5} * \log_2\left({}^{3}/_{5}\right)\right) = 0.97$
Mid	${}^{4}/_{14} = 0.29$	$H(4{,}0) = -\left({}^{4}/_{4} * \log_2({}^{4}/_{4}) + {}^{0}/_{4} * \log_2({}^{0}/_{4})\right) = 0$
Junior	${}^{5}/_{14} = 0.36$	$H(3{,}2) = -\left({}^{3}/_{5} * \log_2\left({}^{3}/_{5}\right) + {}^{2}/_{5} * \log_2\left({}^{2}/_{5}\right)\right) = 0.97$

- Then:
  - $H(\text{Interviewed well} \mid \text{Level}) = 0.36 * 0.97 + 0.29 * 0 + 0.3 * 0.97 = 0.7$
- Thus:
  - $IG(\text{Interviewed well} \mid \text{Level}) = H(\text{Interviewed well}) - H(\text{Interviewed well} \mid \text{Level}) = 0.94 - 0.7 = 0.24$

Calculation

- Similarly:
  - $IG(\text{Interviewed well} \mid \text{Tweets})$
    $= H(\text{Interviewed well}) - H(\text{Interviewed well} \mid \text{Tweets}) = 0.15$
  - $IG(\text{Interviewed well} \mid \text{PhD})$
    $= H(\text{Interviewed well}) - H(\text{Interviewed well} \mid \text{PhD}) = 0.048$
  - $IG(\text{Interviewed well} \mid \text{Lang})$
    $= H(\text{Interviewed well}) - H(\text{Interviewed well} \mid \text{Lang}) = 0.029$
- Level has the highest information gain, therefore it was good to choose that attribute.

Is the previous decision tree good?



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

PhD attribute – subset of 5 records with Junior level

5

	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

- $H(\text{Interviewed well}) = H(3,2) = -({}^{3}/_{5}\log_2({}^{3}/_{5}) + {}^{2}/_{5}\log_2({}^{2}/_{5})) = \underline{0.97}$
- $H(\text{Interviewed well} \mid \text{PhD}) = \sum_i p(\text{PhD} = v_i) * H(\text{Interviewed well} \mid \text{PhD} = v_i)$

v_i	$p(\text{PhD} = v_i)$	$H(\text{Interviewed well} \mid \text{PhD} = v_i)$
Yes	${}^{2}/_{5} = 0.4$	$H(0,2) = 0$
No	${}^{3}/_{5} = 0.6$	$H(3,0) = 0$

- Then: $H(\text{Interviewed well} \mid \text{PhD}) = 0$

Calculation

- Then, the Information gain for each attribute:
  - $IG(\text{Interviewed well} \mid \text{PhD})$
    $= H(\text{Interviewed well}) - H(\text{Interviewed well} \mid \text{PhD}) = 0.97$
  - $IG(\text{Interviewed well} \mid \text{Tweets})$
    $= H(\text{Interviewed well}) - H(\text{Interviewed well} \mid \text{Tweets}) = 0.02$
  - $IG(\text{Interviewed well} \mid \text{Lang})$
    $= H(\text{Interviewed well}) - H(\text{Interviewed well} \mid \text{Lang}) = 0.02$
- PhD has the highest information gain, therefore it was good to choose that attribute next for the Junior Level.

Train a decision tree classifier in scikit-learn

```
from sklearn.tree import DecisionTreeClassifier

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

Some important parameters:

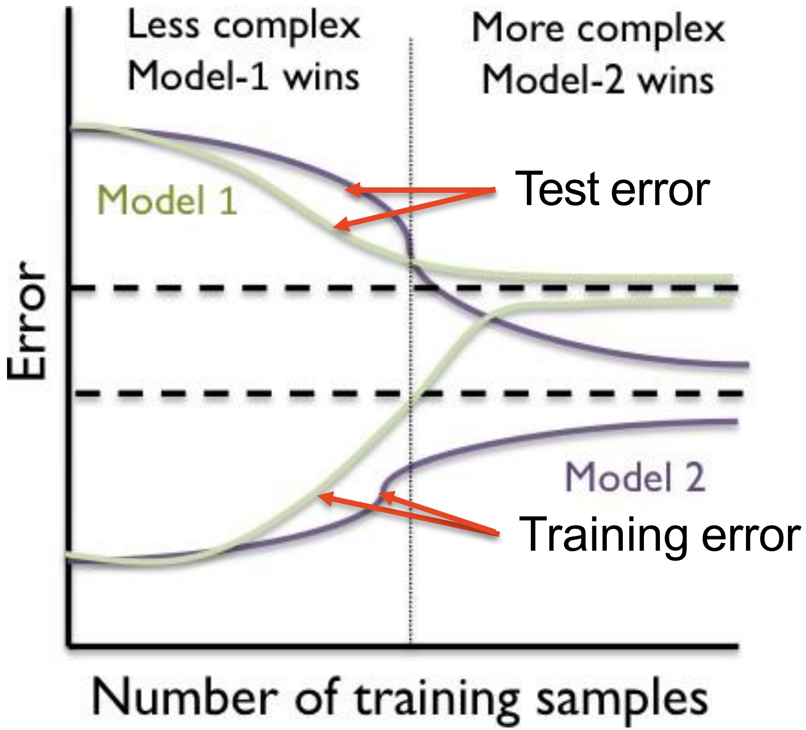
- **max_depth:** the maximum depth of the tree.
- **criterion:**
  - **gini**: choose splits that minimise misclassification.
  - **entropy**: choose splits that minimise total uncertainty.
- **splitter:**
  - **best**: choose the optimal threshold for each feature.
  - **random**: choose the best random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

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.

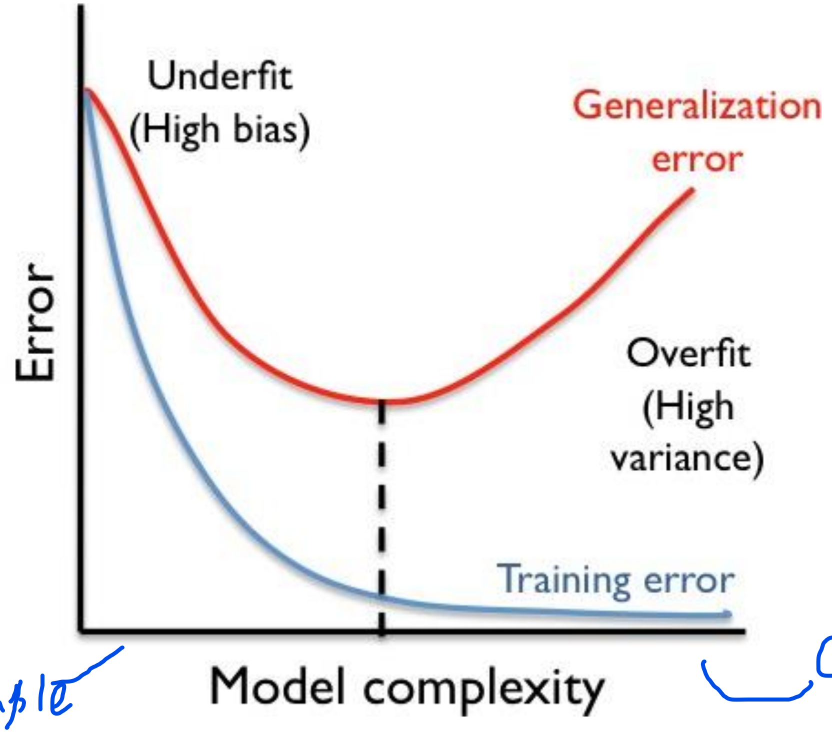
Model choice depends on amount of data available



- 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



simple ⟵ Complex model

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

- The dashed line on right shows point where we switch from 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.

Data drift (non-stationary data)

What is it?

- 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.

BUILDING A GOOD SOLUTION

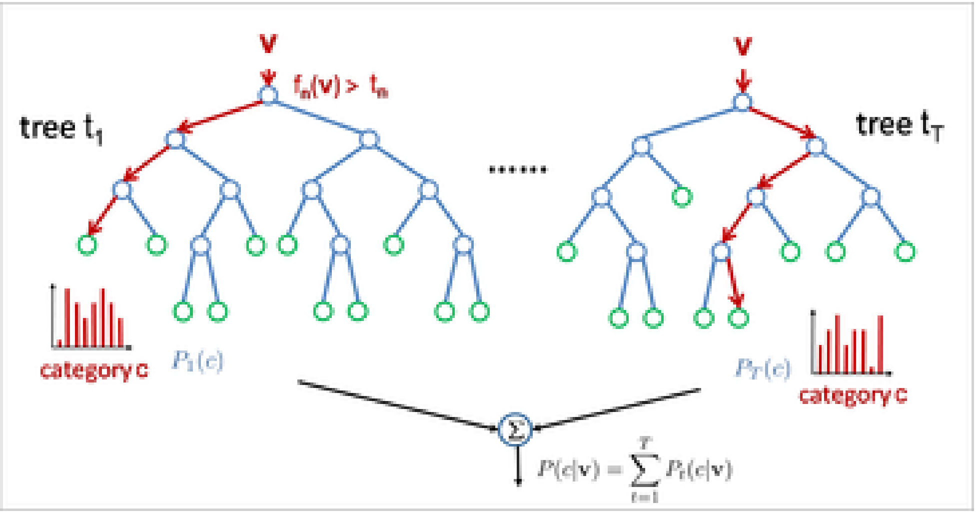
Build a simple model first, evaluate, iterate

- 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.

COMMUNICATING RESULTS

Telling a story

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

A2

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?

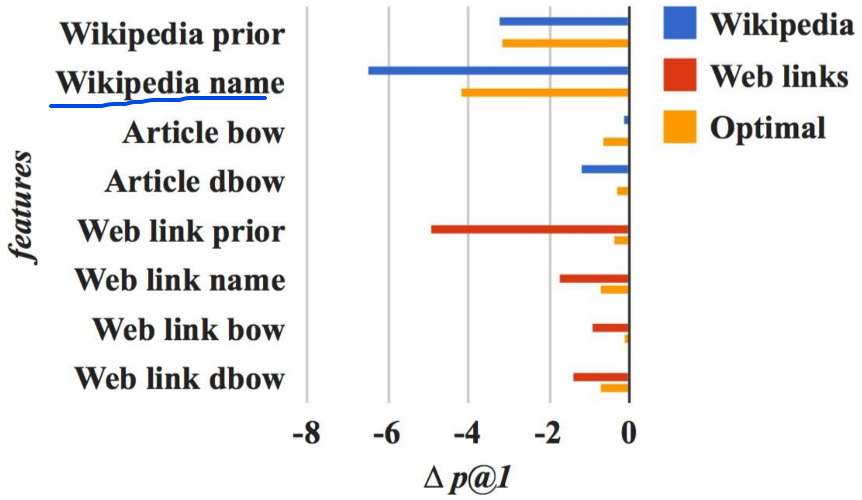
Reporting accuracy and reliability

- 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



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

- 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.

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.
- Things to keep in mind:
  - Efficiency.
  - Reliability of code.
  - Monitoring drift.

REVIEW

W10 review: Decision tree

Objective

Learn techniques for supervised machine learning, with tools in Python.

Lecture

- Decision tree
- Evaluation setup
- Build a good solution
- Communicate results

Readings

- Data Science from Scratch, Ch. 17

Exercises

- sklearn: decision tree
- sklearn: random forest

On good data science

- How to evaluate machine learning models: https://machinelearningmastery.com/how-to-evaluate-machine-learning-algorithms/
- 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