GENERAL ASSEMBLY

DATA SCIENCE

11 WEEK PART TIME COURSE

Week 6 - Bonus Round
Saturday 30th January 2016

AGENDA

1. Review
2. Dimensionality Reduction / LDA
3. Text Analysis
4. Amazon Web Services
5. Projects
6. Other Questions
7. Drinks

DATA SCIENCE PART TIME COURSE

REVIEW

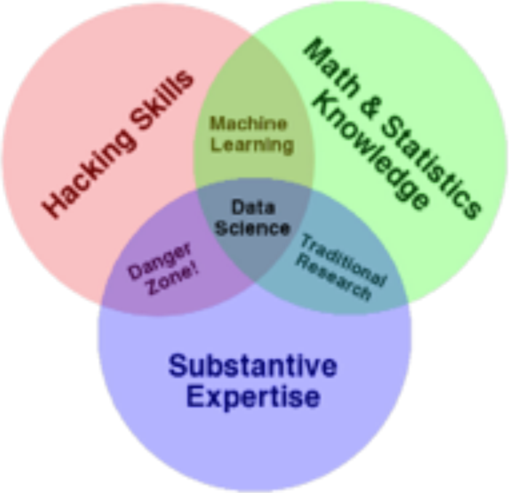
DATA SCIENCE PART TIME COURSE

REVIEW - WEEK 1

CONTENT PHILOSOPHY

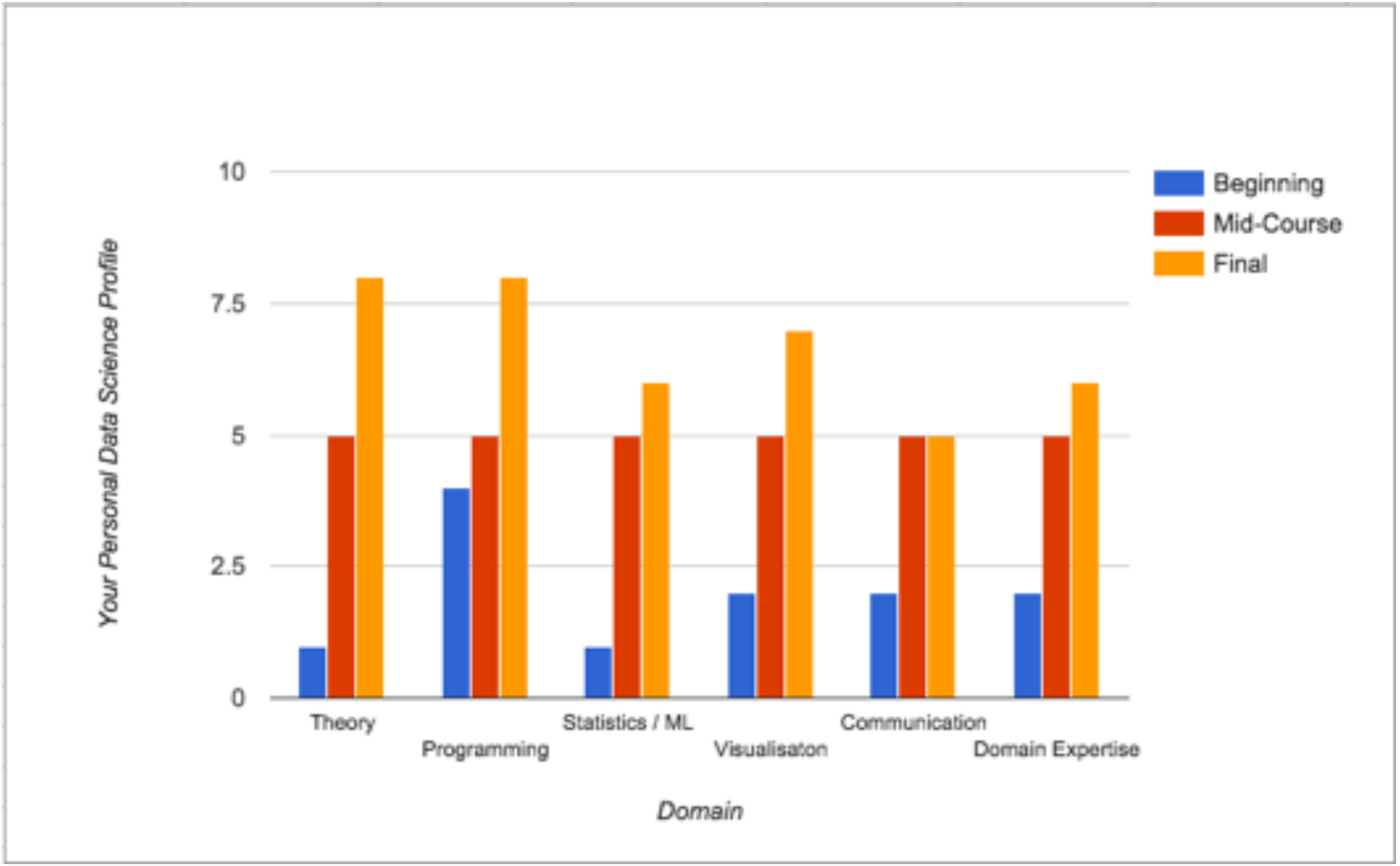
- Multidisciplinary Investigations
- Models and Methods for Data
- Computing with Data
- Pedagogy
- Tool Evaluation
- Theory

Data Science: An Action Plan for Expanding the Technical Areas of the Field of Statistics
William S. Cleveland

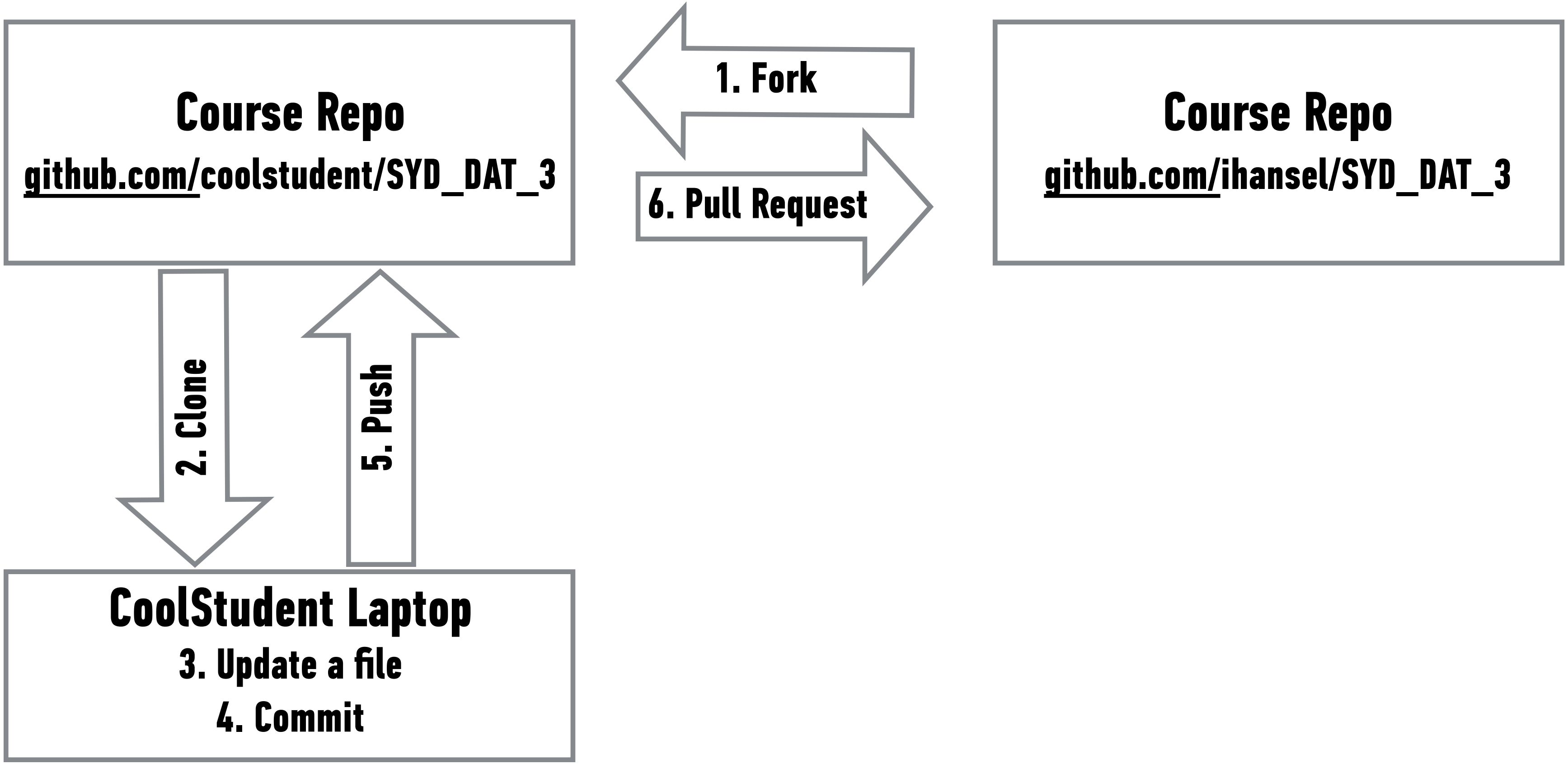


Drew Conway's
Data Science Venn Diagram

CONTENT PHILOSOPHY



HOW DO WE ACTUALLY COLLABORATE WITH GIT?



Course Repo

github.com/coolstudent/SYD_DAT_3

1. Fork

6. Pull Request

Course Repo

github.com/ihansel/SYD_DAT_3

2. Clone

5. Push

CoolStudent Laptop

3. Update a file

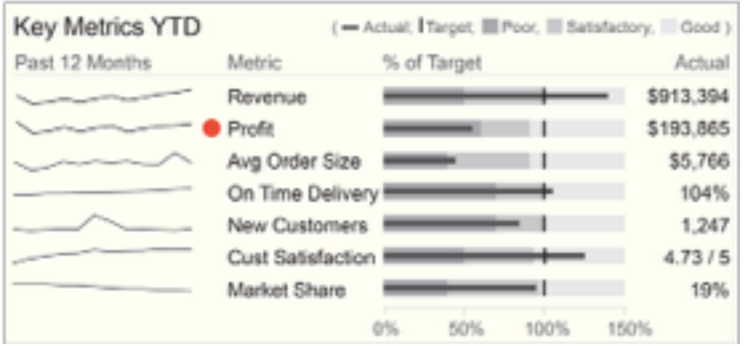
4. Commit

REVIEW - WEEK 2

WHY VISUALISE DATA?

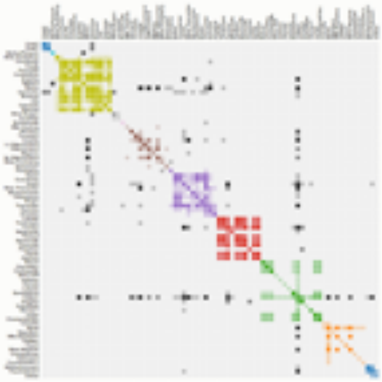
Reporting

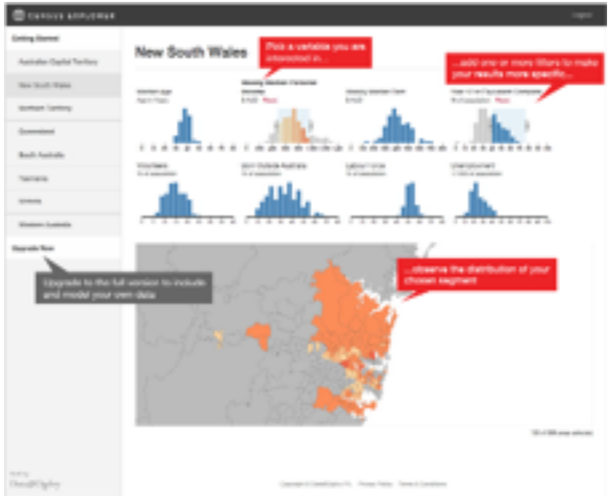
- Dashboards and Business Intelligence
- Know the questions you want answers to
- Can detect changes from the norm
- Good for taking a 30,000 foot view of the problem



Exploring

- Exploratory Data Analysis
- Combines multiple data sources for single view of a problem
- Technical analysis of data
- Combined with modelling allows for the discovery of new problems and solutions





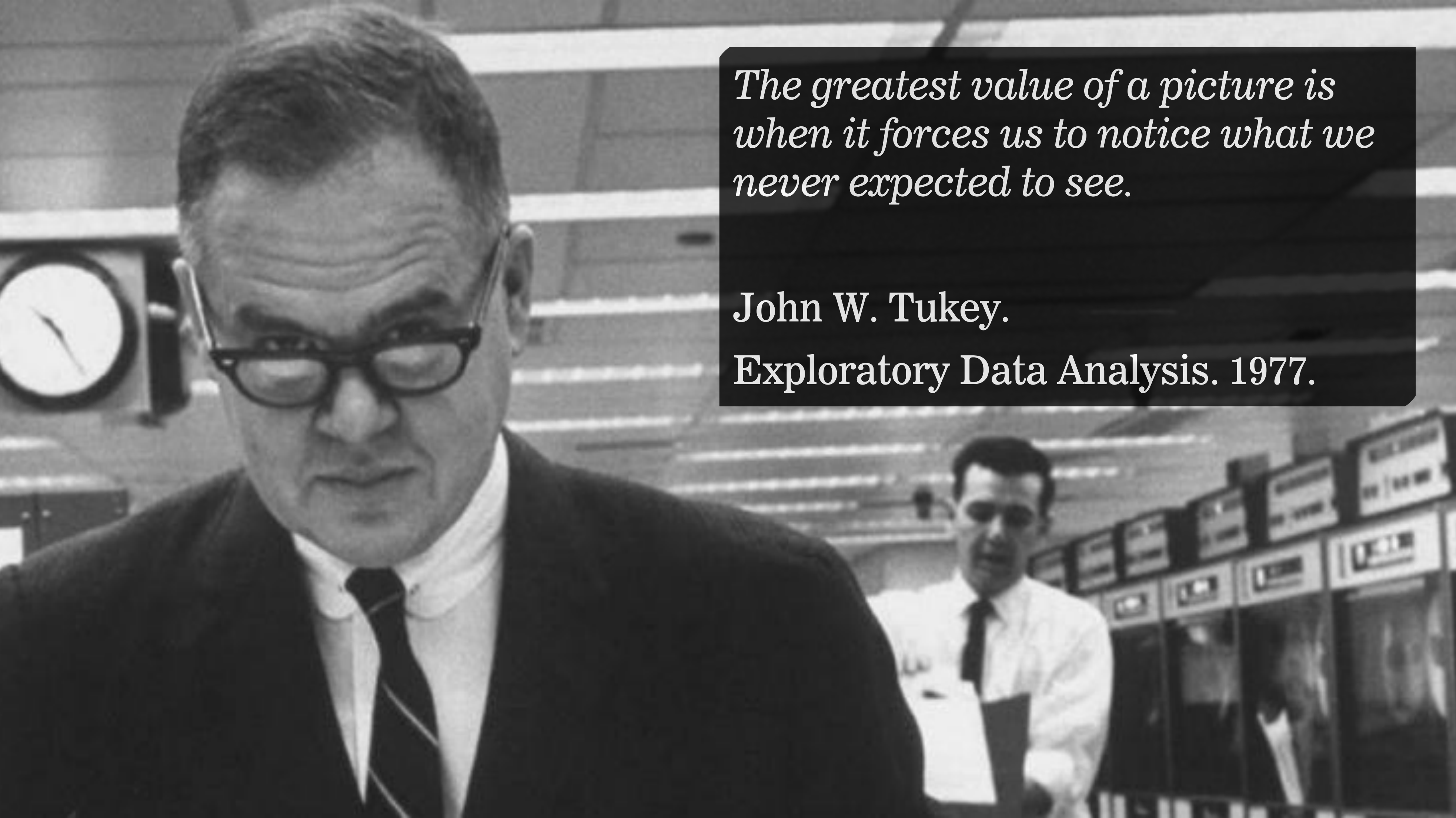
New South Wales

Pick a variable you are interested in...

...add one or more filters to make your results more specific...

Upgrade to the full version to include and model your own data

...observe the distribution of your chosen segment



The greatest value of a picture is when it forces us to notice what we never expected to see.

John W. Tukey.

Exploratory Data Analysis. 1977.

We want to predict some value, let’s call it **y**, based on some observed data we have, let’s call that **x**.

We will use statistical learning to estimate a function that approximates **y** based on the input, **x**.

y is also called; label, dependent variable, target

x is also called; predictor, independent variable, features

SUPERVISED LEARNING – REGRESSION & CLASSIFICATION

If the y variable is numeric then we have a regression problem - we are trying to predict a continuous number

If the y variable is a category (for example trying to predict a type of flower) the we have a classification problem - we are trying to classify what group that y belongs to.

UNSUPERVISED LEARNING

We want to find some underlying structure or patterns in the data but in this case we don’t have any labeled data.

So for example, if we have a large group of customers but would like to separate them into groups (or clusters) to better target them.

$$y = X\beta + \epsilon$$

- y = target variable
- X = input variable
- β = coefficients
- ϵ = error term

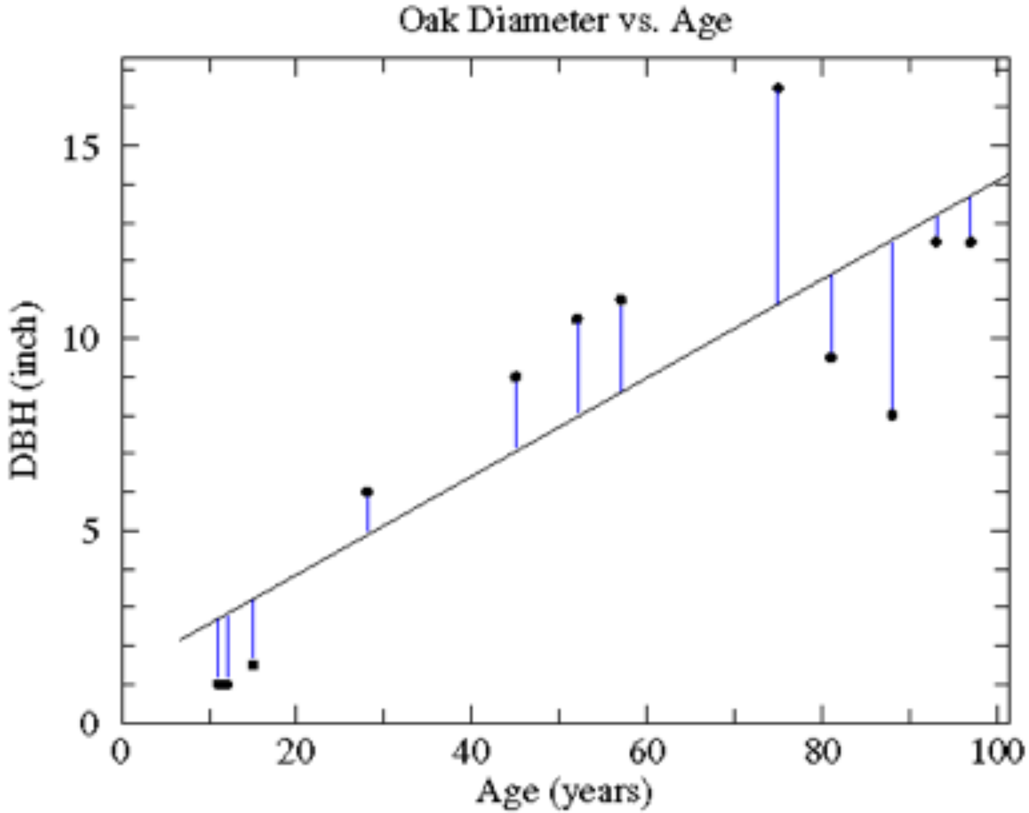
Note, one of our input variables can be 1 so we have an intercept parameter

ORDINARY LEAST SQUARES

$$\boxed{SS_{res}} = \boxed{\sum_{i=1}^{n}} \boxed{\left(\boxed{\boxed{y_i} - \boxed{f(x_i)}}\right)^2}$$

Basically, what we are trying to do is minimise the Residual Sum of Squares. This is the Sum of the squared difference between our observed value and the value from the model

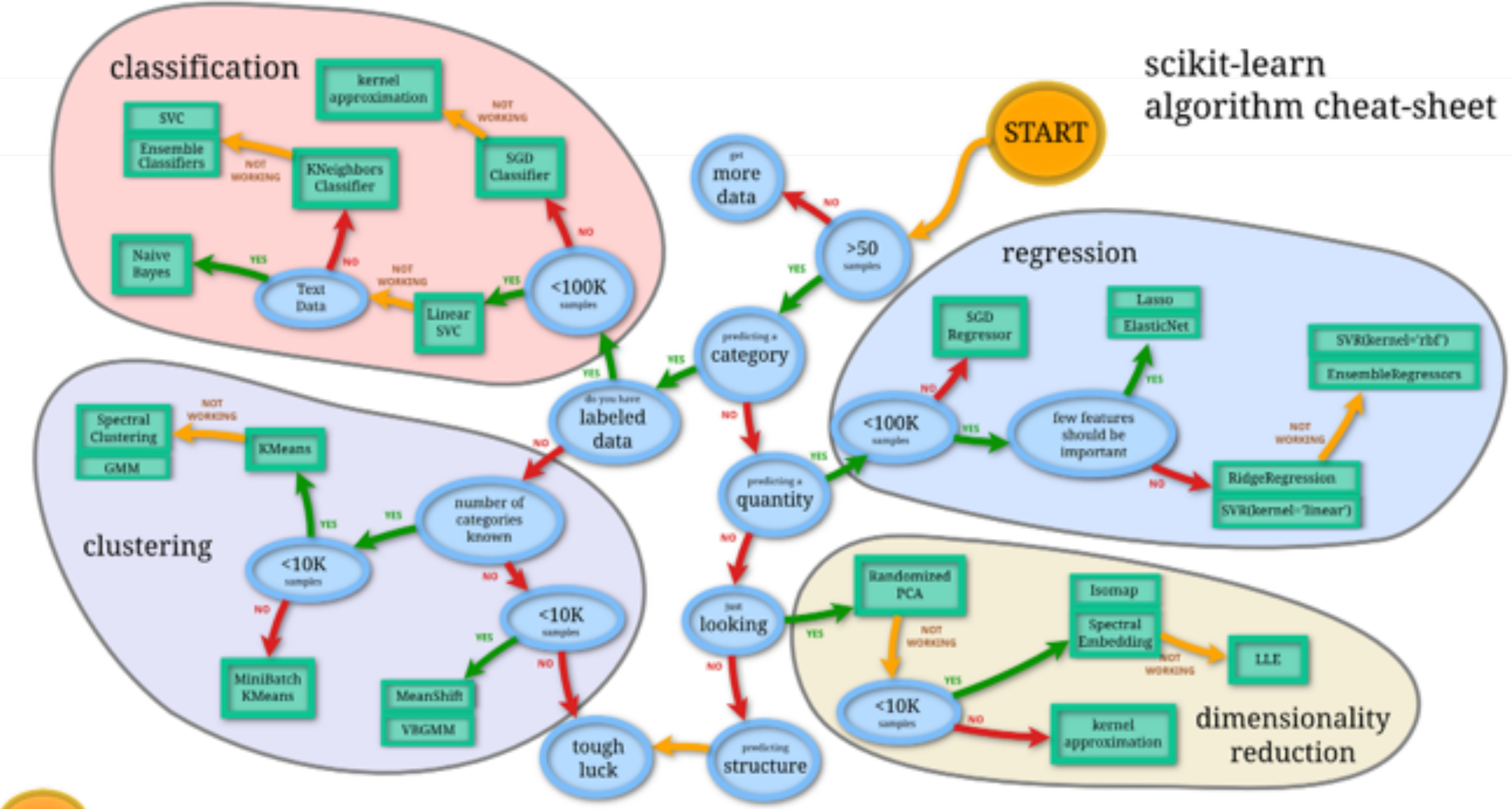
ORDINARY LEAST SQUARES



DATA SCIENCE PART TIME COURSE

REVIEW - WEEK 3

scikit-learn algorithm cheat-sheet



Back

We want to build a classifier that correctly identifies which class our target variable y belongs to given our input variable x.

Why not use the linear regression model?

$$y = X\beta + \epsilon$$

LOGISTIC REGRESSION

- If we only have a binary response variable (0 or 1) it might make sense… BUT we can have our estimated value of $y > 1$ or $y < 0$ … which doesn’t make sense.
- What of the case where we have more than one class? Linear regression cannot easily handle these cases.
- We want a classification method that can handle these cases and give us results we can easily interpret.

Q: What’s wrong with training error?

Thought experiment:

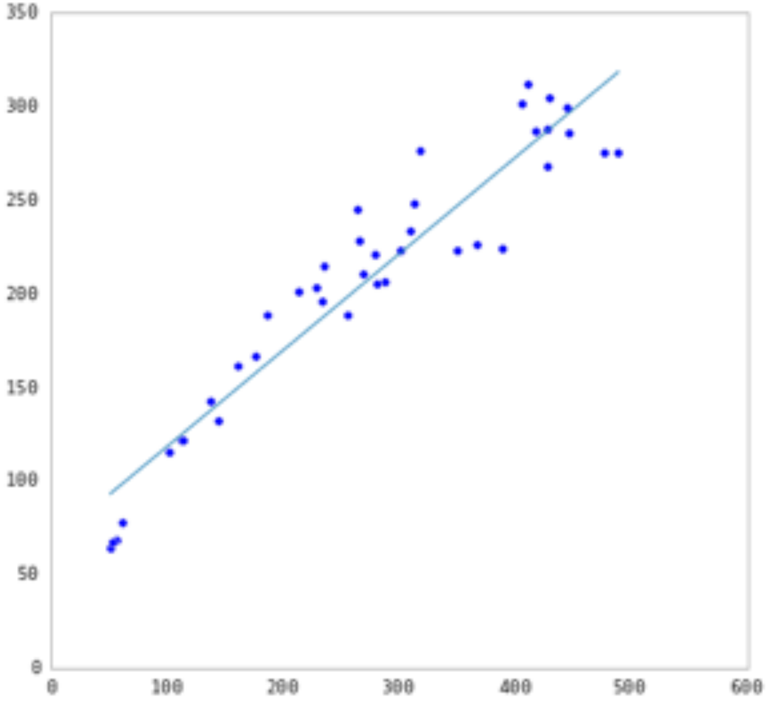
Suppose we train our model using the entire dataset.

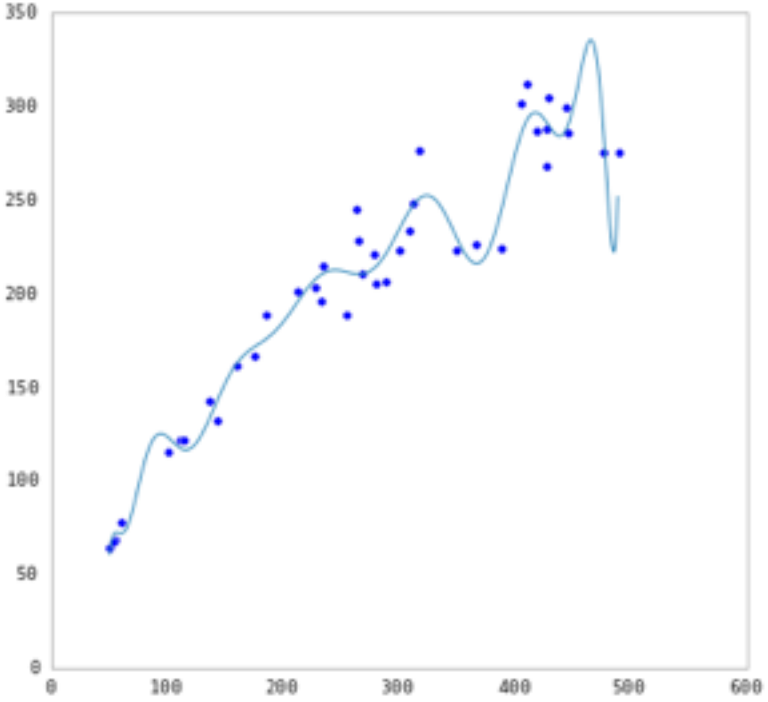
Q: How low can we push the training error?

- We can make the model arbitrarily complex (effectively “memorizing” the entire training set).

A: Down to zero!

TRAINING ERROR

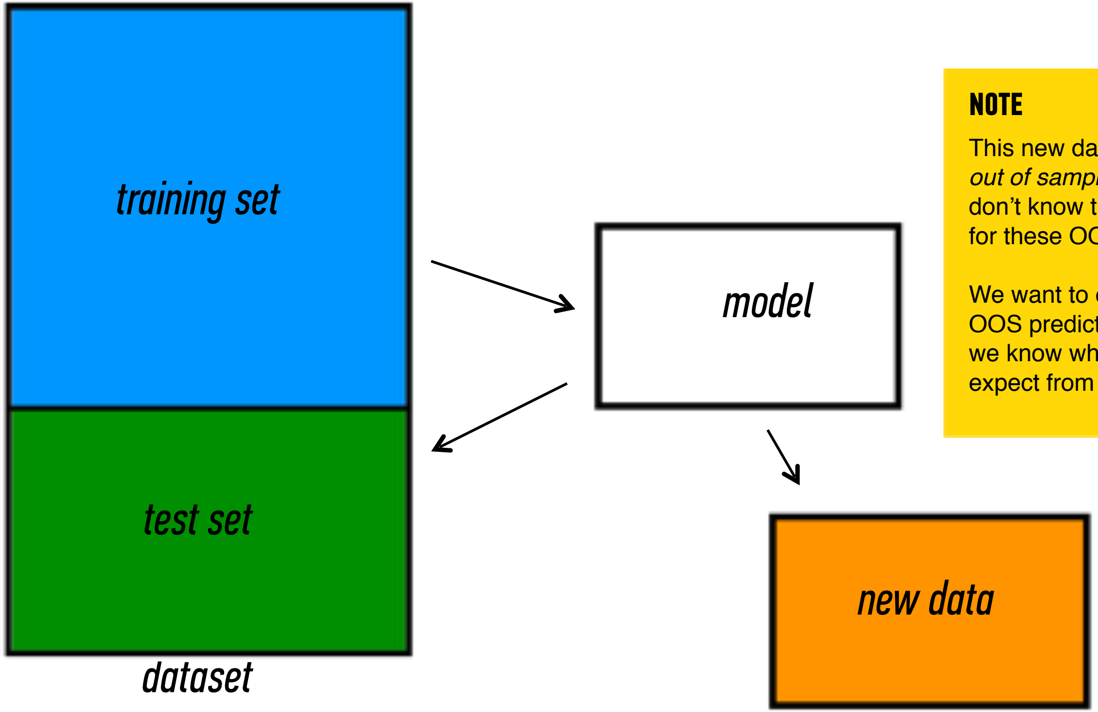




TEST SET APPROACH

Q: How can we make a model that generalizes well?

1) split dataset
2) train model
3) test model
4) parameter tuning
5) choose best model
6) train on all data
7) make predictions on new data



NOTE

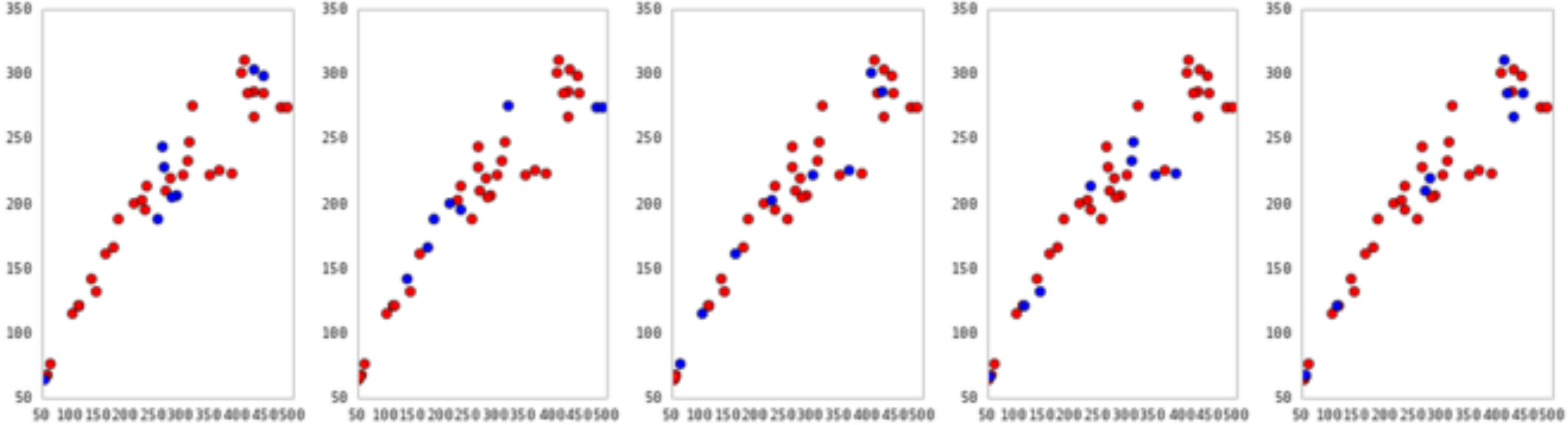
This new data is called *out of sample* data. We don't know the labels for these OOS records!

We want to estimate OOS prediction error so we know what to expect from our model.

Steps for K-fold cross-validation:

1) Randomly split the dataset into K equal partitions.
2) Use partition 1 as test set & union of other partitions as training set.
3) Calculate test set error.
4) Repeat steps 2-3 using a different partition as the test set at each iteration.
5) Take the average test set error as the estimate of OOS accuracy.

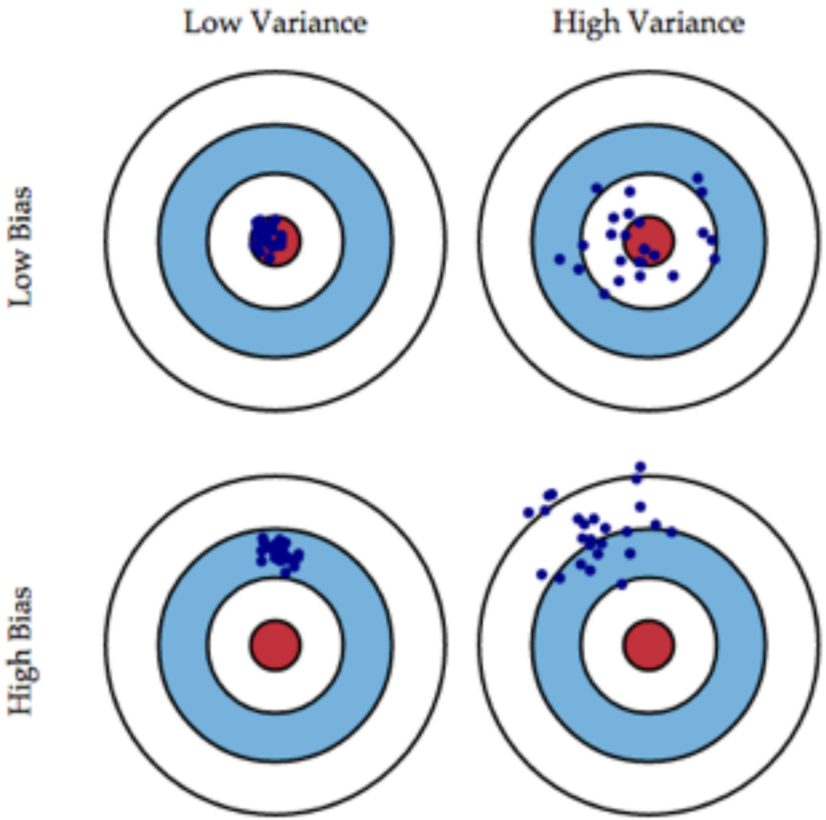
CROSS VALIDATION



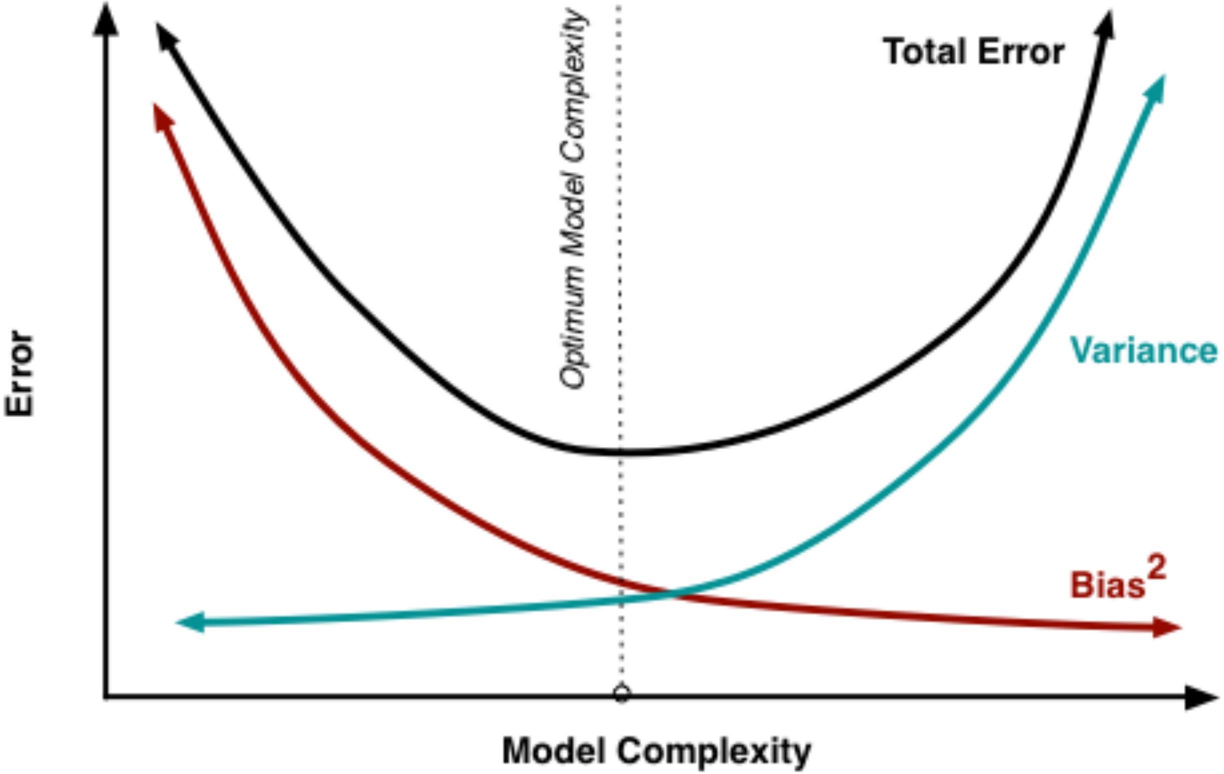
5-fold cross-validation: red = training folds, blue = test fold

Source: http://nbviewer.ipython.org/github/fonnesbeck/Bios366/blob/master/notebooks/Section6_3-Model-Selection-and-Validation.ipynb

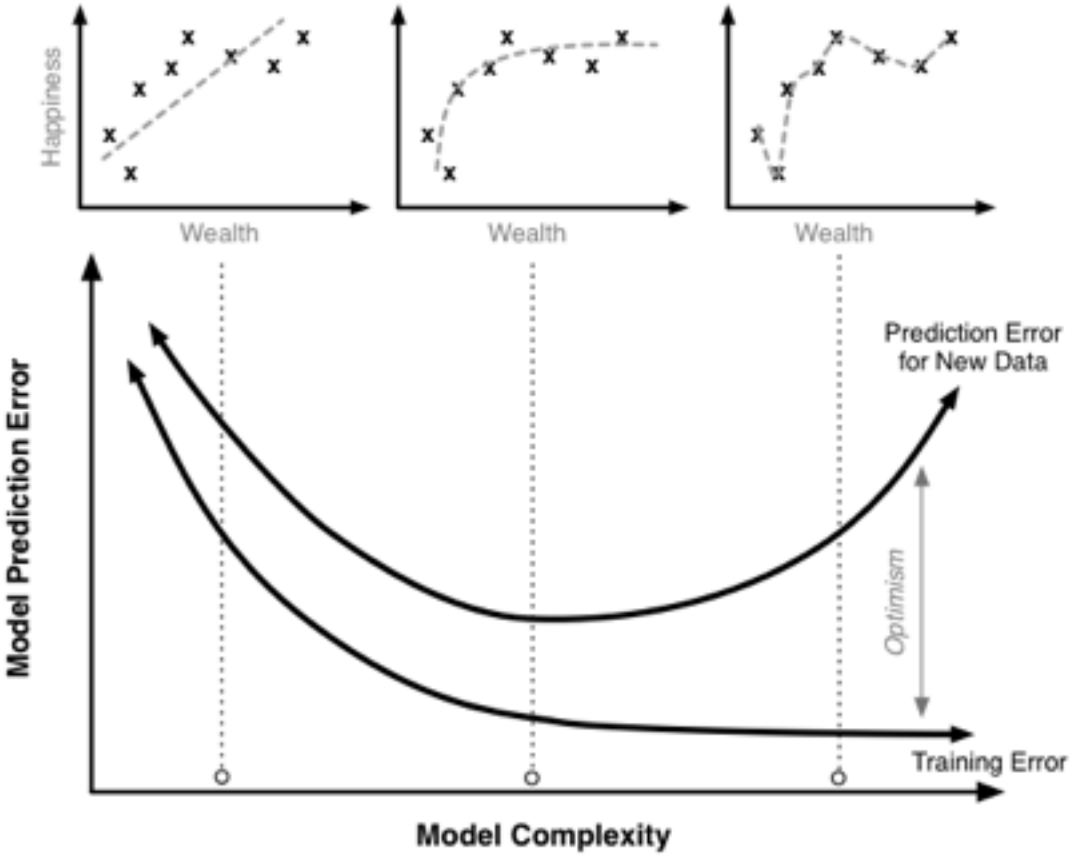
BIAS – VARIANCE TRADEOFF



BIAS – VARIANCE TRADEOFF



BIAS – VARIANCE TRADEOFF



DATA SCIENCE PART TIME COURSE

REVIEW - WEEK 4

Produse
INTERNATIONALE
CURRY PASTE
TABASCO
SUSHI RICE

We could fit a separate linear regression model for every combination of our features.

But what happens when we have a large number of features?

Computation time becomes a factor and we also need to consider that as we include more features we are increasing the chance we include a variable that doesn’t add any predictive power for future data.

HOW DOES REGULARIZATION WORK?

- A tuning parameter lambda (or sometimes alpha) imposes a penalty on the size of coefficients.
- Instead of minimizing the "loss function" (mean squared error), it minimizes the "loss plus penalty".
- A tiny alpha imposes no penalty on the coefficient size, and is equivalent to a normal linear model.
- Increasing the alpha penalizes the coefficients and shrinks them toward zero.

RIDGE REGRESSION

Ridge Regression is similar to least squares, except we include a penalty term,

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \mathrm{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2,$$

the λ term is a tuning parameter. When it is zero we get least squares, as it increases the term, $\lambda\sum_{j=1}^{p}\beta_j^2$ (the shrinkage penalty) has more of an impact and the coefficients will *approach* zero.

LASSO REGRESSION

Lasso Regression is similar to Ridge Regression, except we have the absolute value of beta in our penalty term,

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \mathrm{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|.$$

the λ term is a tuning parameter. When it is zero we get least squares, as it increases the term, $\lambda\sum_{j=1}^{p}|\beta_j|$ (the shrinkage penalty) has more of an

impact and the coefficients will **equal** zero.

WHAT IS A CLUSTER?

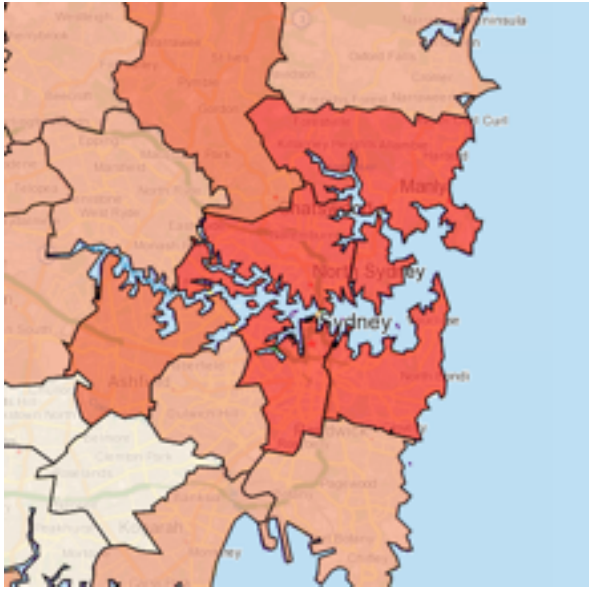
Recall unsupervised learning is when we are trying to find interesting patterns or groups in our data. We don’t have a variable we are trying to predict (a Y value).

Clustering aims to discover subgroups in our data where the points are similar to each other. So we have a collection of groups and all points belonging to the same group are similar. Points in different groups are different to each other.

We have to decide what variables we will construct the groups on. What makes them different (or similar)?

WHY WOULD WE CLUSTER DATA?

Marketing teams might want to group customers into like groups as a way of summarising the data

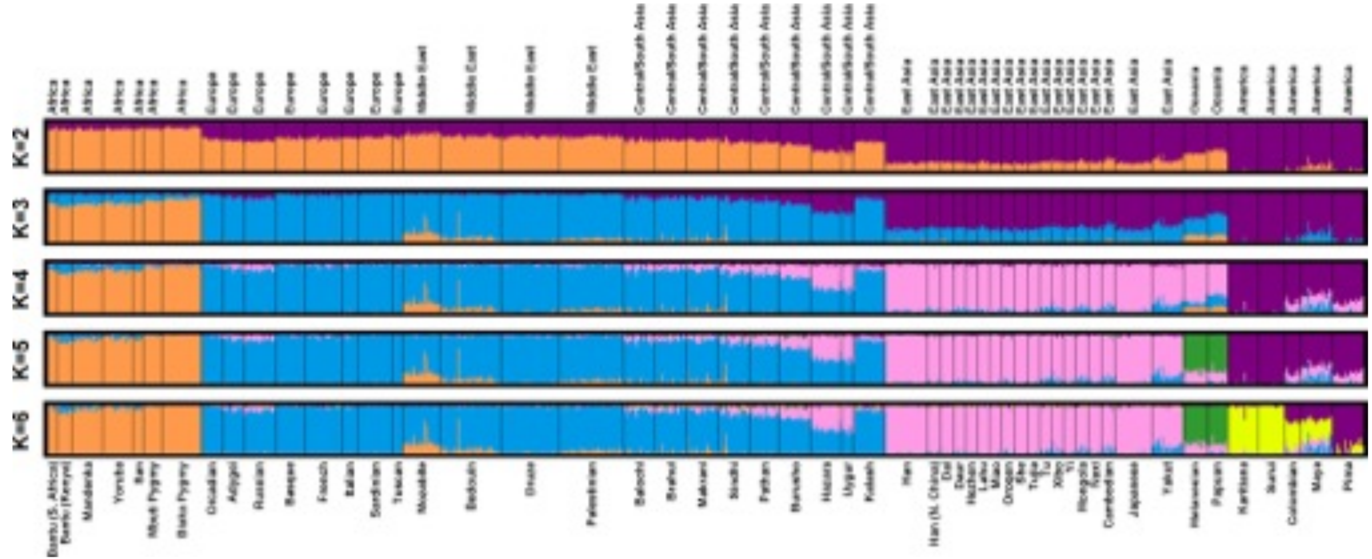


WHY WOULD WE CLUSTER DATA?

Financial groups may want to group transactions into like groups as a way to find unusual payments

WHY WOULD WE CLUSTER DATA?

Genetics data can be clustered to identify ancestry

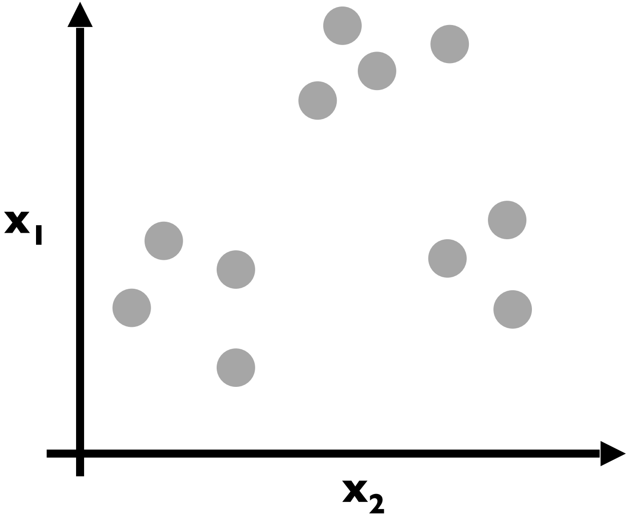


1) Choose k initial centroids

2) For each point:
   - find distance to each centroid
   - assign point to nearest centroid

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



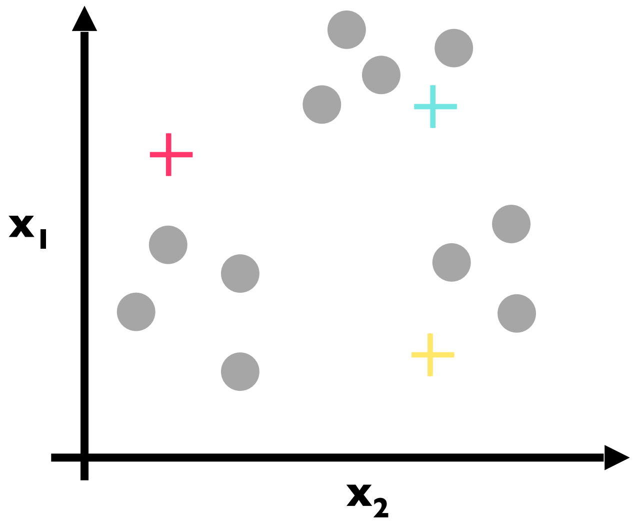
1) Choose k initial centroids

2) For each point:

- find distance to each centroid
- assign point to nearest centroid

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



1) Choose k initial centroids

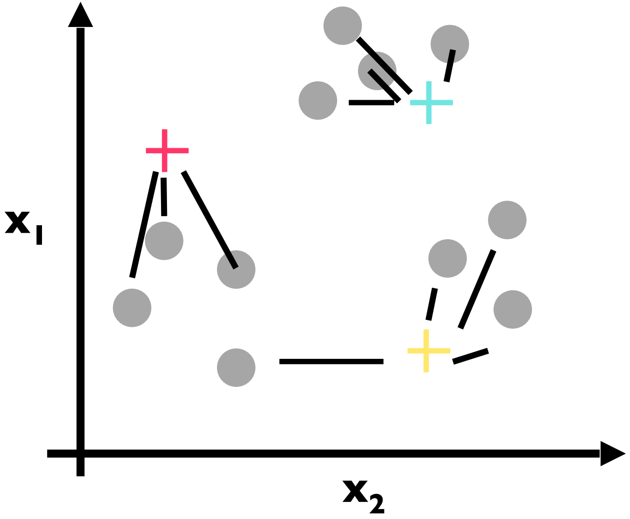
2) For each point:

   - find distance to each centroid

   - assign point to nearest centroid

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



1) Choose k initial centroids

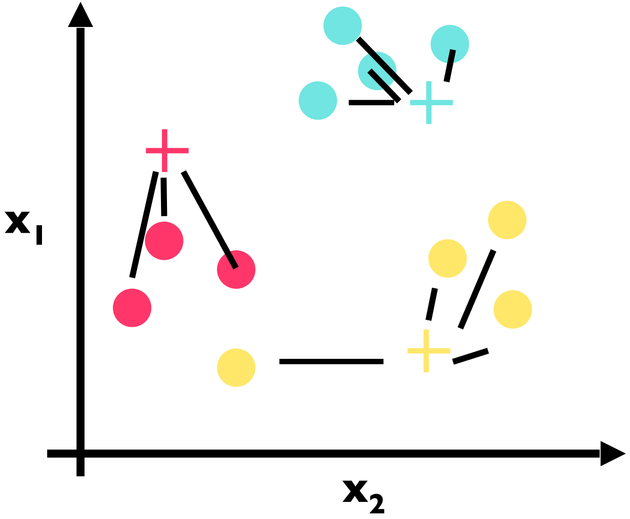
2) For each point:

   - find distance to each centroid

   - **assign point to nearest centroid**

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



1) Choose k initial centroids

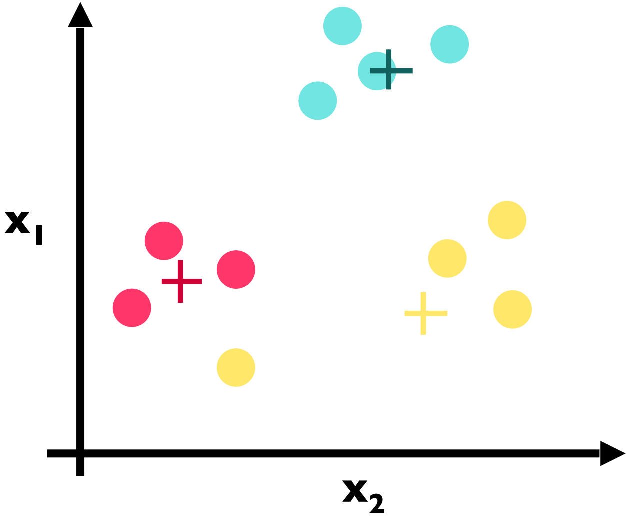
2) For each point:

   - find distance to each centroid

   - assign point to nearest centroid

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



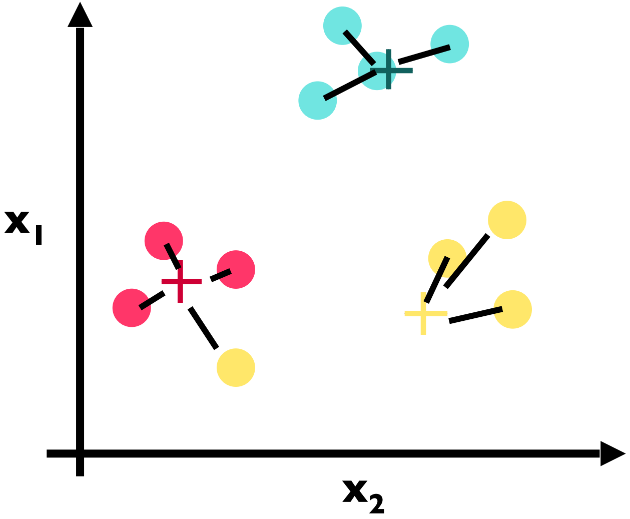
1) Choose k initial centroids

2) For each point:

- find distance to each centroid
- assign point to nearest centroid

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



1) Choose k initial centroids

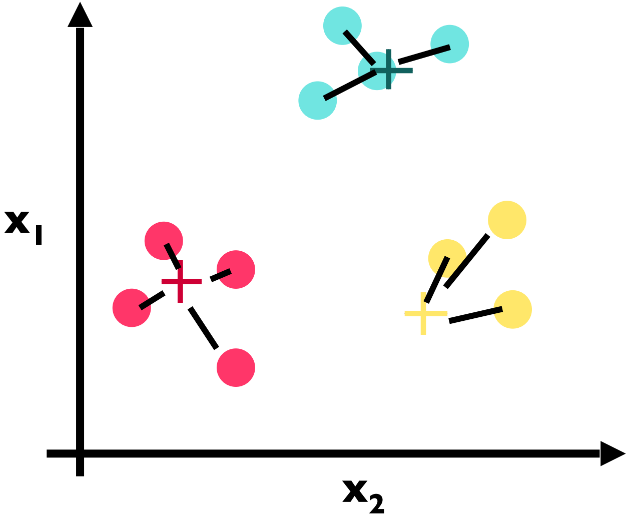
2) For each point:

    - find distance to each centroid

    - assign point to nearest centroid

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



1) Choose k initial centroids

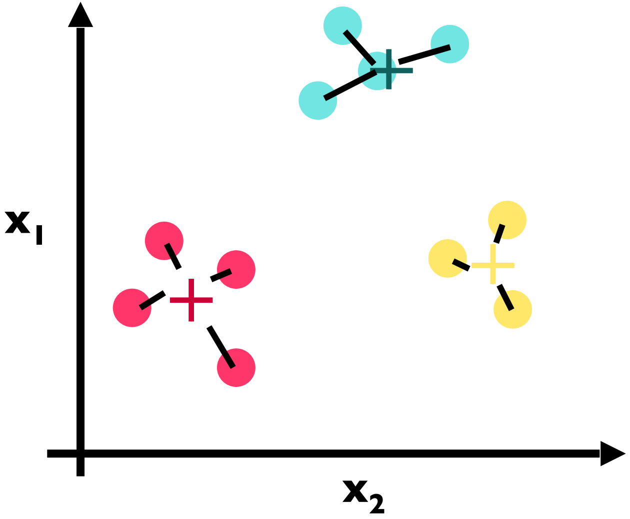
2) For each point:

  - find distance to each centroid

  - assign point to nearest centroid

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



1) Choose k initial centroids

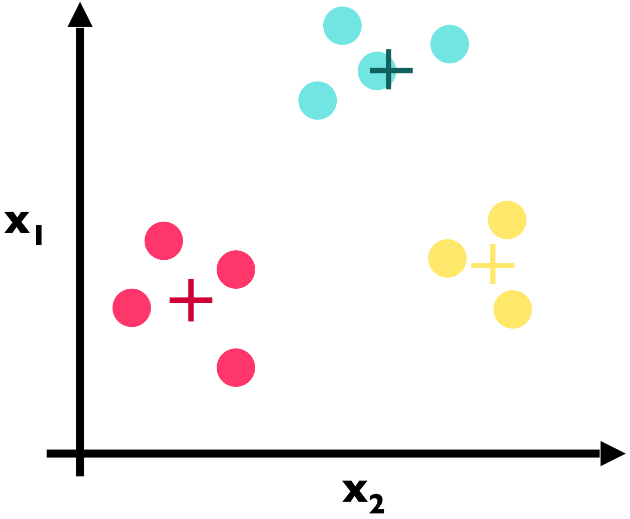
2) For each point:

  - find distance to each centroid

  - assign point to nearest centroid

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



DATA SCIENCE PART TIME COURSE

REVIEW - WEEK 5

RECOMMENDATION ENGINES

Recommendation engines aims to match users to things (movies, songs, items, events, etc) they might enjoy but have not yet tried.

The rating is produced by analysing other user/item ratings (and sometimes item characteristics) to provide personalised recommendations to users.

eHarmony® LinkedIn P PANDORA Spotify

CONTENT-BASED FILTERING

Content-based filtering begins by mapping each item into a feature space. Both users and items are represented by vectors in this space.

Item vectors measure the degree to which the item is described by each feature, and user vectors measure a user’s preferences for each feature.

Ratings are generated by taking dot products of user & item vectors.

CONTENT-BASED FILTERING

Content-based filtering has some difficulties:

- Must map items into a feature space (manual work)
- Recommendations are limited in scope (items must be similar to each other)
- Hard to create cross-content recommendations (eg books/music films…this would require comparing elements from different feature spaces)

COLLABORATIVE FILTERING

“Customers who purchased X also purchased Y”

Someone with similar tastes to you will be able to recommend things you might like, e.g. people who watch ‘The Newsroom’ will probably enjoy ‘The Social Network’ because there is a large audience in common.

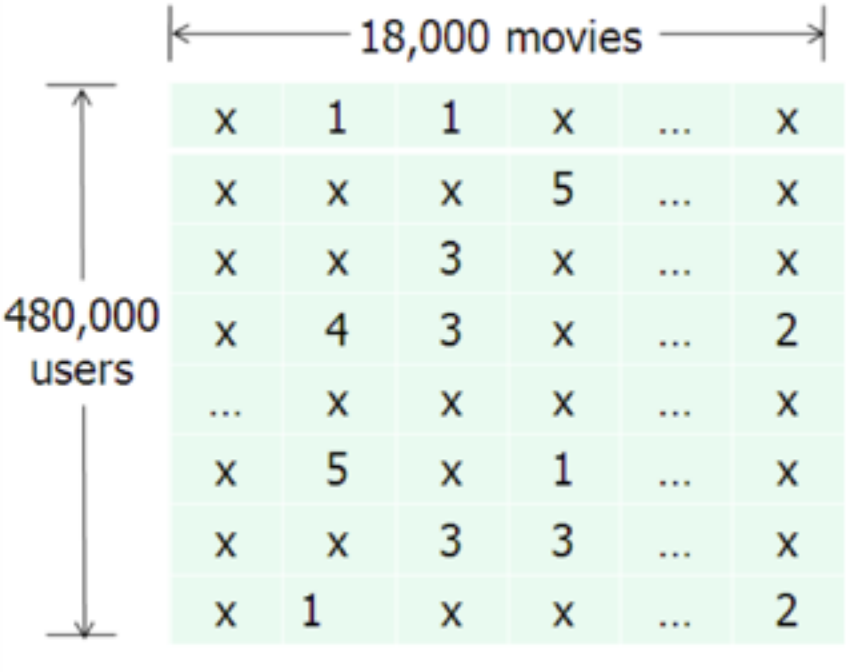
Collaborative filtering refers to a family of methods for predicting ratings where instead of thinking about users and items in terms of a feature space, we are only interested in the existing user-item ratings themselves.

In this case, our dataset is a ratings matrix whose columns correspond to items, and whose rows correspond to users.

COLLABORATIVE FILTERING

This will be the general form of the data we analyse for collaborative filtering.

The method relies on previous user-item ratings (or feedback).



18,000 movies

480,000 users

x	1	1	x	...	x
x	x	x	5	...	x
x	x	3	x	...	x
x	4	3	x	...	2
...	x	x	x	...	x
x	5	x	1	...	x
x	x	3	3	...	x
x	1	x	x	...	2

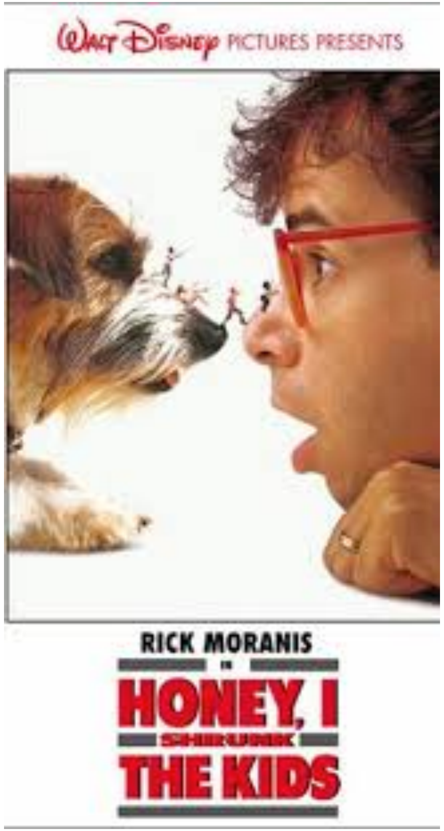
DATA SCIENCE PART TIME COURSE

DIMENSIONALITY REDUCTION / LDA

DIMENSIONALITY REDUCTION

A set of techniques for reducing the size (in terms of features,
records, and/or bytes) of the dataset under examination.

In general, the idea is to regard the dataset as a matrix and to decompose the matrix into simpler, meaningful pieces.



GOALS OF DIMENSIONALITY REDUCTION

- We’d like to analyze the data using the most meaningful basis (or coordinates) possible.

- More precisely: given an n x d matrix X (encoding n observations of a d-dimensional random variable), we want to find a k-dimensional representation of X ($k < d$) that captures the information in the original data, according to some criterion.

‘It finds a low-dimensional representation of a data set that contains as much as possible of the variation. The idea is that each of the n observations lives in p-dimensional space, but not all of these dimensions are equally interesting. PCA seeks a small number of dimensions that are as interesting as possible, where the concept of interesting is measured by the amount that the observations vary along each dimension. Each of the dimensions found by PCA is a linear combination of the p features.’

- Introduction to Statistical Learning

Slightly different to PCA (but very much related). Linear Discriminant Analysis (LDA) takes into account the class of an observation.

It aims to find a linear combination of features that separate two or more different classes.

Can be used as an alternative to Logistic Regression.

DATA SCIENCE PART TIME COURSE

TEXT ANALYSIS

WHAT IS TEXT ANALYSIS

- Text is considered to be un-structured data. This means we don’t have nice features we can use as inputs. We will have to construct them using a model or rules we know about language.
- When analysing text we need to consider what we actually want to use the text for
  - Sentiment Analysis
  - Topic Labelling
  - Classification

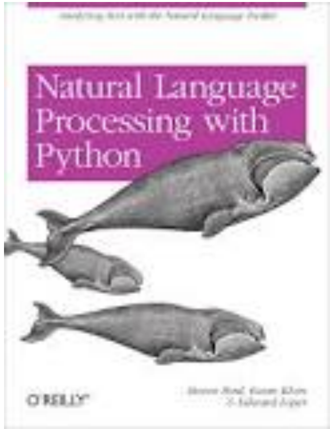
WHAT IS TEXT ANALYSIS?

- Entity Extraction
- Sentiment Analysis
- Keyword Extraction
- Concept Tagging
- Relation Extraction
- Taxonomy Classification
- Author Extraction
- Language Detection
- Text Extraction
- Microformats Parsing
- Feed Detection
- Linked Data Support

AlchemyAPI™
An IBM Company

NATURAL LANGUAGE TOOLKIT (NLTK) - Python

- NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.



MORE LEARNING

- Udacity Deep Learning Course by Google – https://www.udacity.com/course/deep-learning--ud730
- Gaussian Process Models – http://www.gaussianprocess.org/gpml/
- Great Podcast – http://www.thetalkingmachines.com/

COURSE REMAINDER

- Practical Skills (SQL, No-SQL, Cloud)
- Advanced Topics (Time Series, Graph Analysis, Neural Networks, Natural Language Processing)
- Course Review
- Project Presentations (Final 2 sessions)