TORONTO – BIG DATA UNIVERSITY:

DATA MINING ALGORITHMS

POLONG LIN & SAEED AGHABOZORGI

#bringthepizzaback
#BDUmeetup
@BigDataU
@panago_pizza

RYERSON
Wifi Password: **EGGY1**
Username: **ibm.workshop**
Password: **RUguest739**

Like us: https://www.facebook.com/bigdatauniversity

TORONTO – BIG DATA UNIVERSITY:

DATA MINING ALGORITHMS

POLONG LIN & SAEED AGHABOZORGI

#bringthepizzaback
#BDUmeetup
@BigDataU
@panago_pizza

Wifi: **UofT**
Username: **scs.guest**
Password: **guestwifi**

bit.ly/BDUalgorithms

Like us: https://www.facebook.com/bigdatauniversity

TORONTO – BIG DATA UNIVERSITY:

DATA MINING ALGORITHMS

POLONG LIN & SAEED AGHABOZORGI

#bringthepizzaback
#BDUmeetup
@BigDataU
@panago_pizza

bit.ly/BDUalgorithms

Like us: https://www.facebook.com/bigdatauniversity

AGENDA

- 6:00pm - 6:15pm - Event Introduction
- 6:15pm - 7:00pm - Algorithms & Demos
- 8:00pm - Networking

THANK YOU TO OUR SPONSORS!

WHAT IS BIG DATA UNIVERSITY (BDU)?

BigDataUniversity.com

- A **community** initiative led by IBM
- **@yourpace, @yourplace** online courses about data
- Developed by **industry experts**
- **Free** courses by the community with hands-on labs
- Certificate of completion and **badges**
- Looking for **contributors**!

QUICK TOUR OF BIG DATA UNIVERSITY

BigDataUniversity.com

Chinese:

BigDataUniversity.com.cn

Portuguese:

BigDataUniversity.com.br

LAUNCH OF DATA SCIENCE WORKSHOP SERIES

Tuesdays 6pm - 8:30pm: DS for Beginners
Thursdays 6pm - 8:30pm: DS for Intermediate/Advanced

Meetups will be **live streamed** and **recorded**

Data Scientist Workbench

Prepare data. Analyze data. Get answers.

www.datascientistworkbench.com

DATA SCIENCE METHODOLOGY

Business Understanding

Analytic Approach

Data Requirements

Data Collection

Data Understanding

Prediction
Interpretation
Justification
Testing

Feedback

Deployment

Evaluation

Modelling

Data Preparation

MAJOR ANALYTICAL APPROACHES & ALGORITHMS

- Associations
  - E.g. frequent co-occurring Items
  - Algorithms: a priori association rules
- Classification
  - E.g. prediction of item class
  - Algorithms: decision trees (ID3, C4.5, C5.0), CART, SVM, NN, Naïve Bayes, CHAID
- Estimation/Prediction
  - Predicting a continuous value
  - Algorithms: regression, SVM (support vector machines), KNN (K-nearest neighbours)
- Clustering
  - E.g. Finding cluster of patients
  - Algorithms: k-means, Hierarchical Clustering

MAJOR ANALYTICAL APPROACHES & ALGORITHMS

- Sequence mining
  - E.g. Click-stream
  - Algorithms: Markov Model, HMM
- Dimension Reduction
  - PCA
- Visualization
  - To facilitate human discovery, understanding
- Summarization
  - Describing a group
- Deviation Detection
  - Finding changes
- Link/Graph Analysis
  - Finding relationship

CLASSIFICATION

Business Understanding

Classification

Analytic Approach

Data Requirements

Data Collection

Data Understanding

Prediction
Interpretation
Justification
Testing

Feedback

Deployment

Evaluation

Classification Trees

Modeling

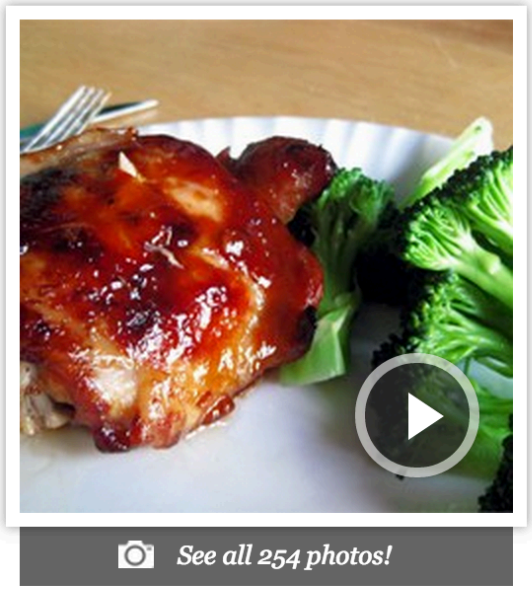
Data Preparation

BUSINESS UNDERSTANDING

Friend and I are looking at some food…

Friend: “How do you know this is Japanese food?”
Me: “Uh… because it has teriyaki sauce?”
Friend: “But how do you *know* that teriyaki = Japanese?”

"BUT HOW DO YOU *KNOW* THIS IS JAPANESE FOOD?"



READY IN 1½ hrs

Baked Teriyaki Chicken

Read Reviews (3963)

Pin it 31K+ | Like 9.1k | Tweet 45 | g+1 167

Recipe by Marian Collins

"A much requested chicken recipe! Easy to double for a large group. Delicious!"

+ Recipe Box | + Shopping List | + Menu | Email | Print

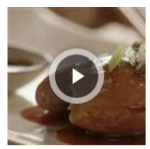
Ingredients Edit and Save

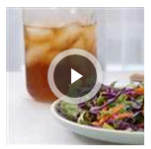
Original recipe makes 6 servings Change Servings

- [ ] 1 tablespoon cornstarch
- [ ] 1 tablespoon cold water
- [ ] 1/2 cup white sugar
- [ ] 1/2 cup soy sauce
- [ ] 1/4 cup cider vinegar
- [ ] 1 clove garlic, minced
- [ ] 1/2 teaspoon ground ginger
- [ ] 1/4 teaspoon ground black pepper
- [ ] 12 skinless chicken thighs

Check All | Add to Shopping List

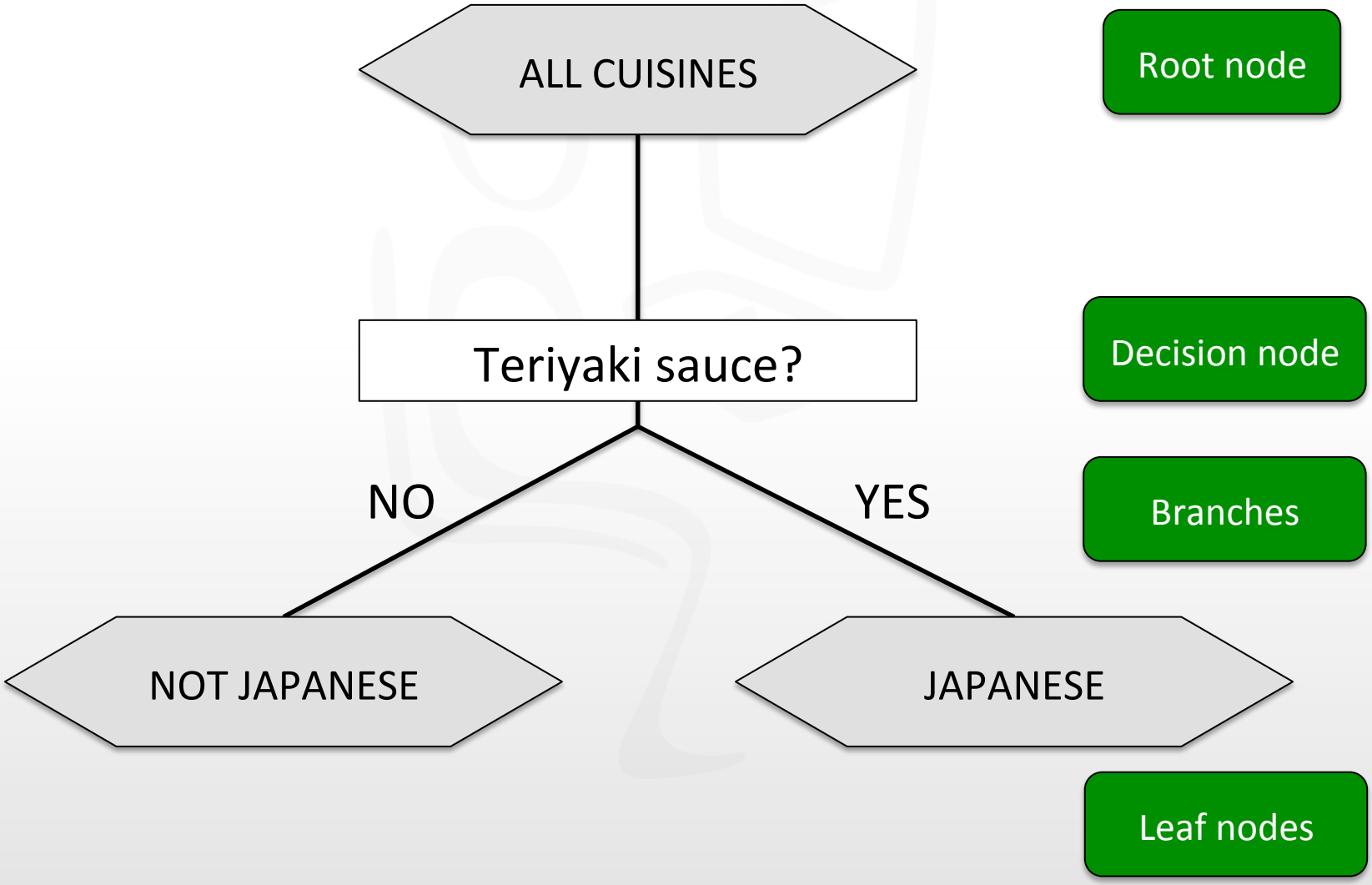
Watch video tips and tricks


Baked Teriyaki Chicken


Crispy and Tender Baked Chicken Thighs

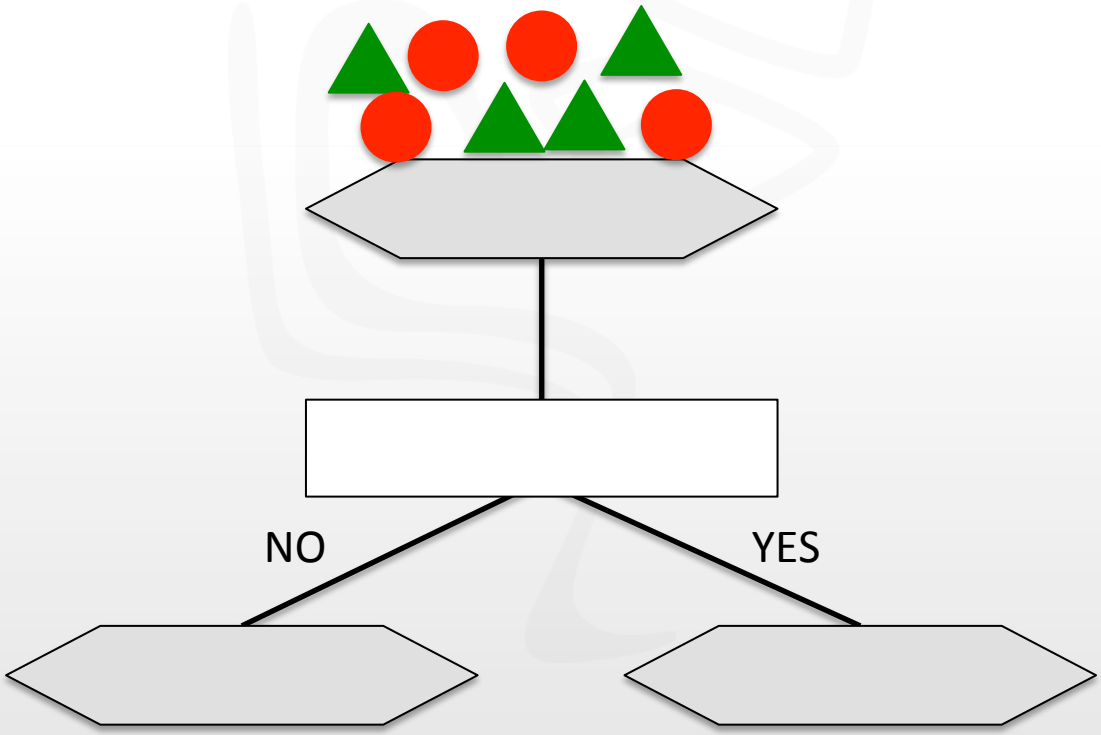
allrecipes.com

CLASSIFICATION - DECISION TREES



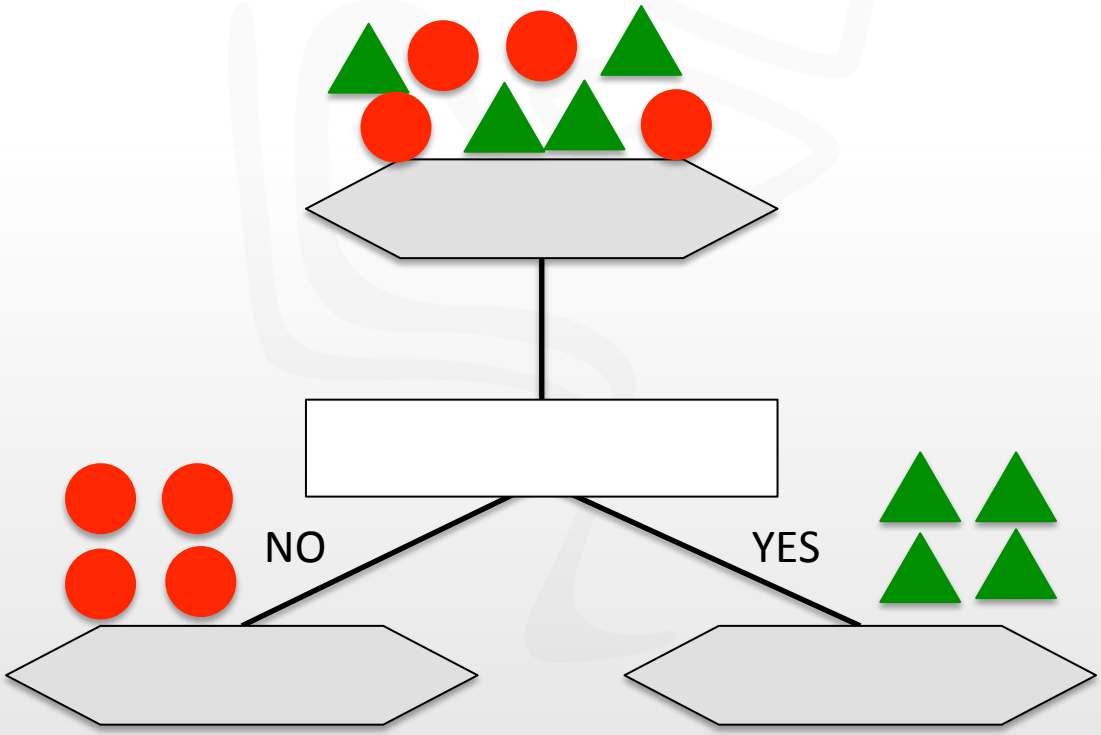
UNDERSTANDING DECISION TREES

- DTs are built using **recursive partitioning** to classify the data
- The algorithm chooses the most predictive feature to split the data on
- “predictiveness” is based on decrease in entropy (gain in information) or “impurity”



UNDERSTANDING DECISION TREES

- DTs are built using **recursive partitioning** to classify the data
- The algorithm chooses the most predictive feature to split the data on
- “predictiveness” is based on decrease in entropy (gain in information) or “impurity”



CHARACTERISTICS OF DECISION TREES

Pros	Cons
Easy to interpret	Easy to overfit or underfit the model
Can handle numeric or categorical features	Cannot model interactions between features
Can handle missing data	Large trees can be difficult to interpret
Uses only the most important features	
Can be used on very large or small data	

A tree stops growing at a node when…

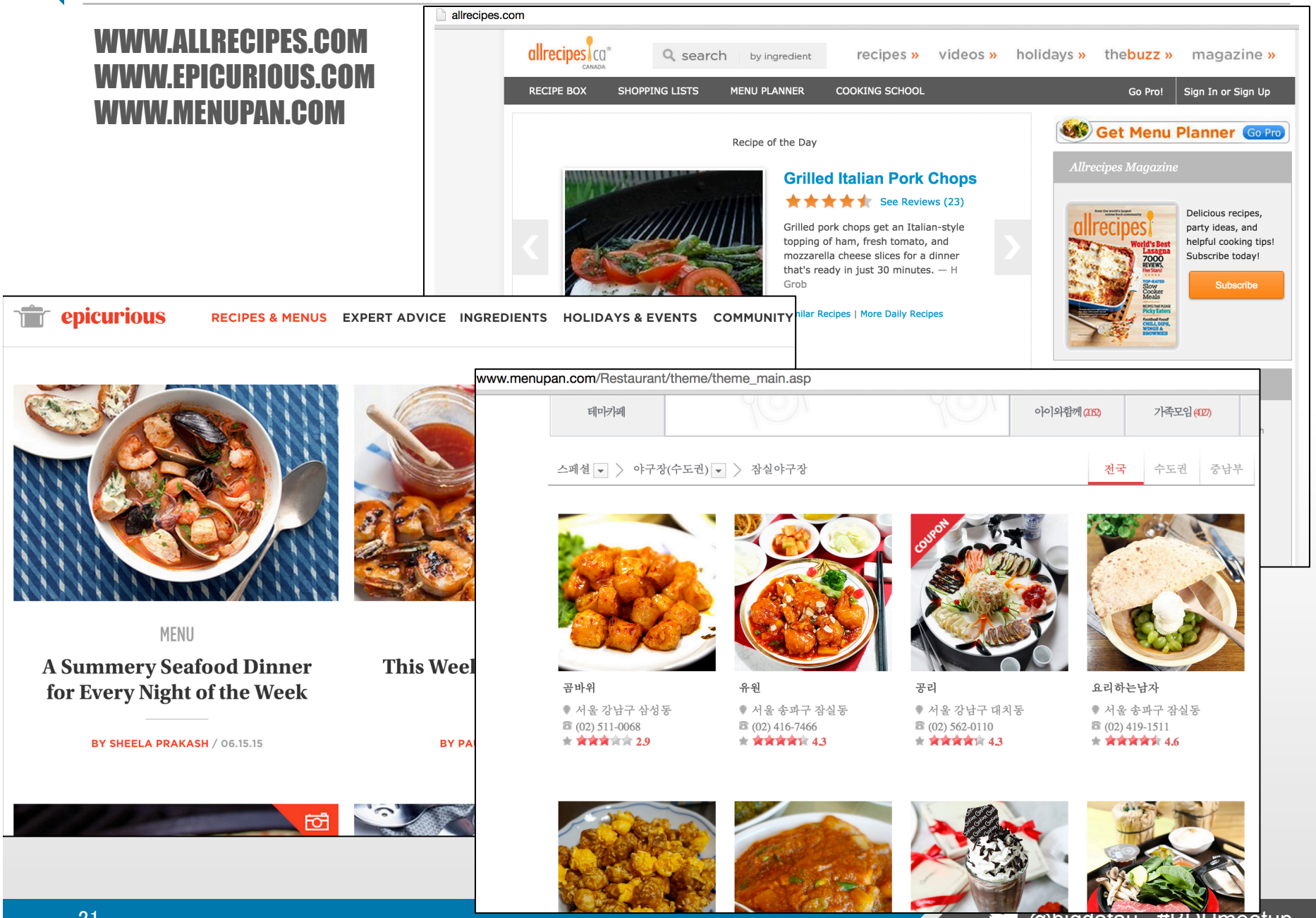
- pure or nearly pure
- no remaining variables on which to further subset the data
- the tree has grown to a preselected size limit

UNDERSTANDING THE ALGORITHMS IN DECISION TREES

- How the `rpart` package in R uses recursive partitioning
  - http://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf
- `scikit-learn` in Python
  - http://scikit-learn.org/stable/modules/tree.html
- "Machine learning – decision trees" by Professor Nando de Freitas
  - https://www.youtube.com/watch?v=-dCtJjlEEgM
- Datacamp's Kaggle R tutorial on Titanic survivorship
  - https://www.datacamp.com/courses/kaggle-tutorial-on-machine-learing-the-sinking-of-the-titanic

WEB-SCRAPE BY YONG-YEOL AHN

WWW.ALLRECIPES.COM
WWW.EPICURIOUS.COM
WWW.MENUPAN.COM



allr_recipes.txt

epic_recipes.txt

map.txt

menu_recipes.txt

README.md

Goals:

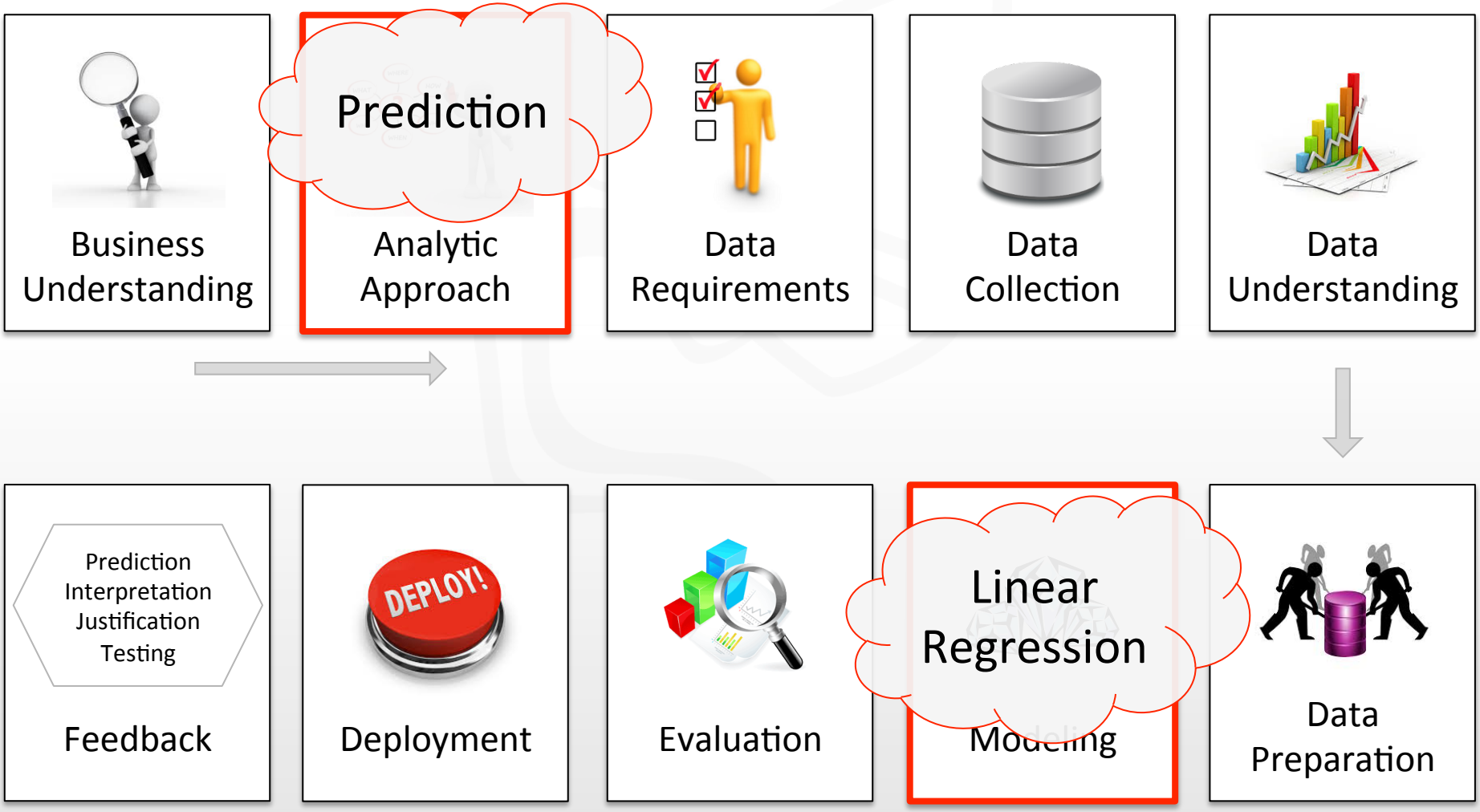
- Combine three .txt files
- Convert to dataframe
- One row per recipe
- One column per ingredient
- Row values = “Yes” or “No”

allr_recipes.txt

```
Canada  egg     yeast   wheat   milk    lard
Canada  pork    carrot  pea     onion   potato
Canada  maple_syrup
Canada  wheat   yeast   almond  honey   oat     date    vegetable_oil   whole_grain_wheat_flour
Canada  butter  lovage  clam    wheat   onion   thyme   potato  yeast   black_pepper    parsley
ginger  bay     celery  cinnamon        milk    mustard
Canada  cane_molasses   butter  wheat   raisin  ginger  lard    egg     milk    cream
Canada  asparagus       olive_oil       pepper  garlic  tomato
Canada  butter  cilantro        tea     jasmine vegetable       brown_rice
Canada  vegetable_oil   wheat   egg     milk
Canada  lemon_juice     onion   soy_sauce       black_pepper    ginger  white_wine      garlic
vegetable_oil   chicken
Canada  butter  cane_molasses   wheat   vanilla egg     milk
Canada  butter  olive_oil       pepper  rice    mushroom        onion   chicken_broth   thyme
garlic  basil   porcini
Canada  tomato  onion   black_pepper    garlic  mozzarella_cheese       tuna    rice
Canada  wheat   lard
Canada  butter  pepper  lemon   beef    fish    parsley white_wine      basil   wheat   cream
Canada  butter  cane_molasses   wheat   vanilla cocoa   egg     milk
Canada  vegetable_oil   potato
Canada  tomato  olive_oil       pork    green_bell_pepper       onion   mushroom        thyme
white_wine      basil   garlic  oregano mozzarella_cheese       rosemary
Canada  cane_molasses   wheat   yeast   raisin  oat     cinnamon        lard
Canada  tomato  olive_oil       mushroom        black_pepper    garlic  oregano basil   rosemary
Canada  butter  coffee  wheat   vanilla cocoa   egg     banana
Canada  butter  cane_molasses   wheat   yeast   milk    whole_grain_wheat_flour
Canada  butter  onion   potato  chicken_broth   black_pepper    squash  celery  carrot
Canada  onion   mushroom        egg_noodle      black_pepper    celery  bell_pepper
mozzarella_cheese       chicken cheddar_cheese  broccoli
Canada  butter  pepper  wheat   onion   potato  asparagus       chicken_broth   black_pepper
```

DEMO

PREDICTION


Business Understanding
Prediction
Analytic Approach
Data Requirements
Data Collection
Data Understanding
Data Preparation
Linear Regression
Modeling
Evaluation
DEPLOY!
Deployment
Prediction
Interpretation
Justification
Testing
Feedback

ESTIMATION/PREDICTION

Prediction example

- Can we predict the Co2 emission of car without test it?
- The Co2 Emission of a car is calculated based on the engine size, class, model, make, Cylinder, consumption of that car. Prediction is used to predict its expected CO2 emission.

What is a prediction?

Prediction is similar to classification but models Continuous/Numerical/Orderd valued

How does it work?

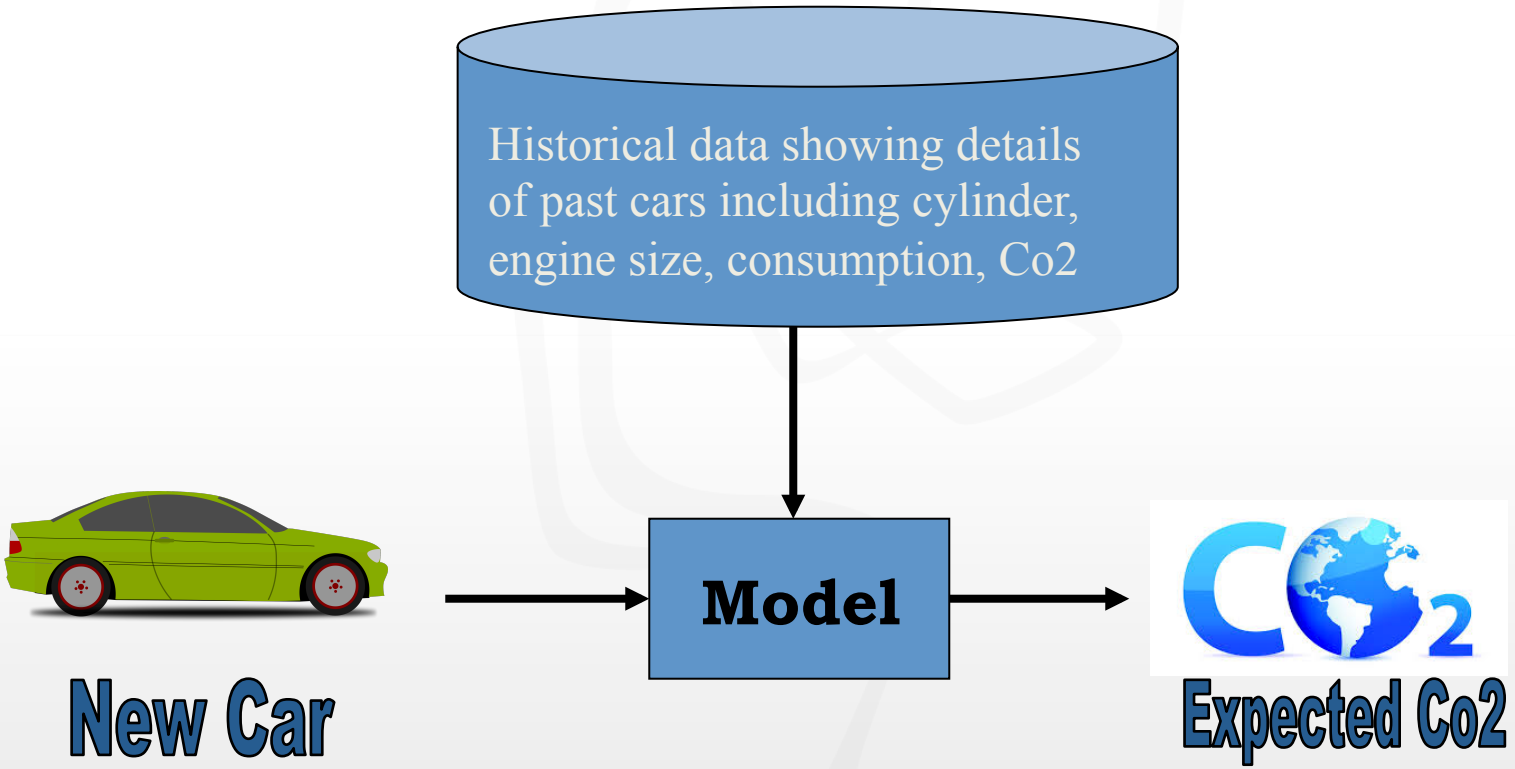
It has following steps:

1. Split your data into training and test set
2. Construct a model using training set
3. Evaluate your model using test set
4. Use model to predict unknown value

```
cdf=df[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_COMB','CO2EMISSIONS']]
cdf.head()
```

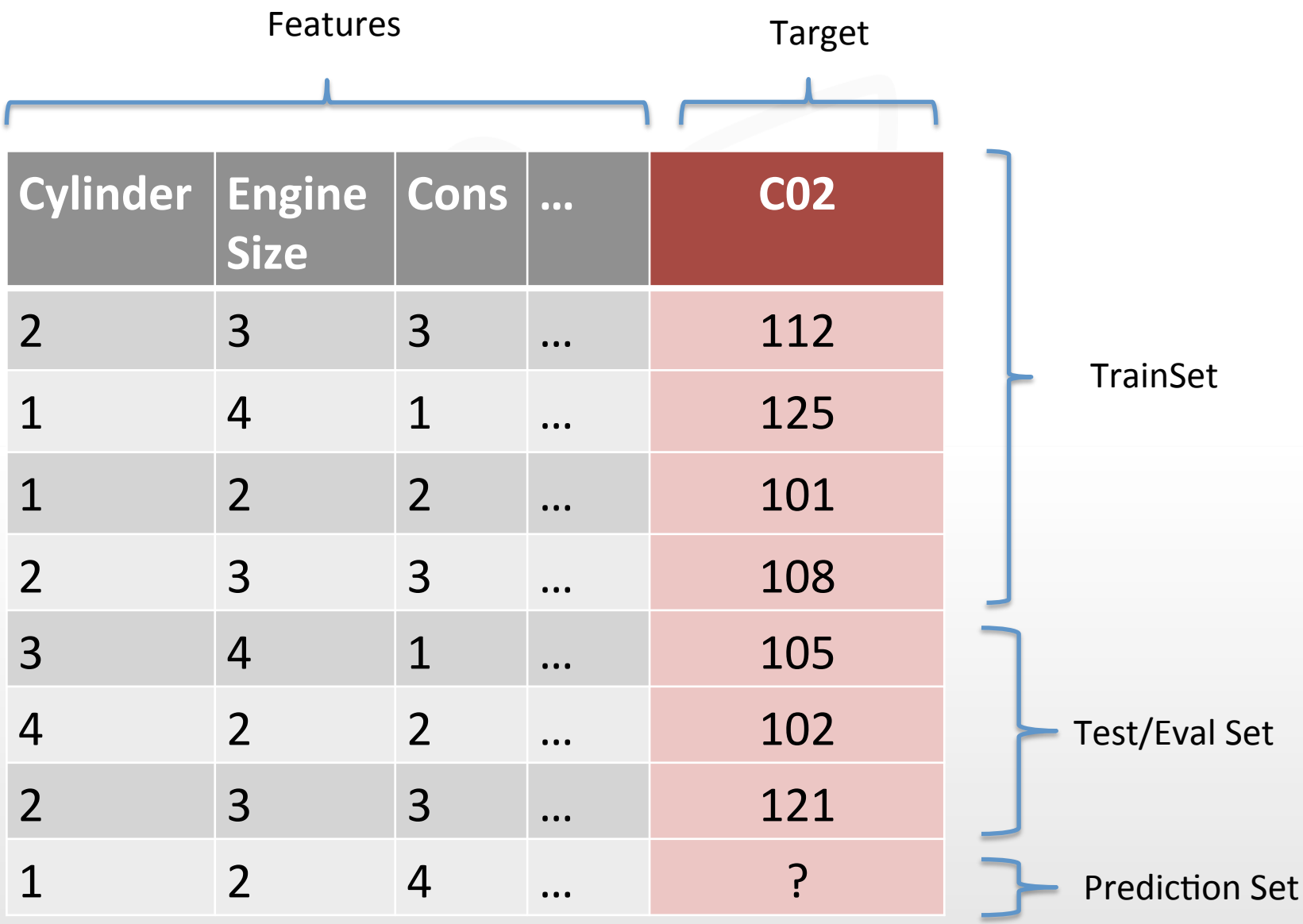
	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244

ESTIMATION/PREDICTION



Predict the expected Co2 Emission for the new car

DATASET



Creating train and test dataset

```
msk = np.random.rand(len(df)) < 0.8
train = cdf[msk]
test = cdf[~msk]
```

PREDICTION

- Algorithms:
  - Regression
    - Simple regression
    - Multiple regression
    - Linear regression
    - Non-linear regression
  - *k*-nearest neighbor methods
  - Neural Networks
  - Support Vector Regression

- **What is Regression Analysis?**
  - Simple Regression Analysis (one independent)
  - Multiple Regression Analysis (Multiple independent)

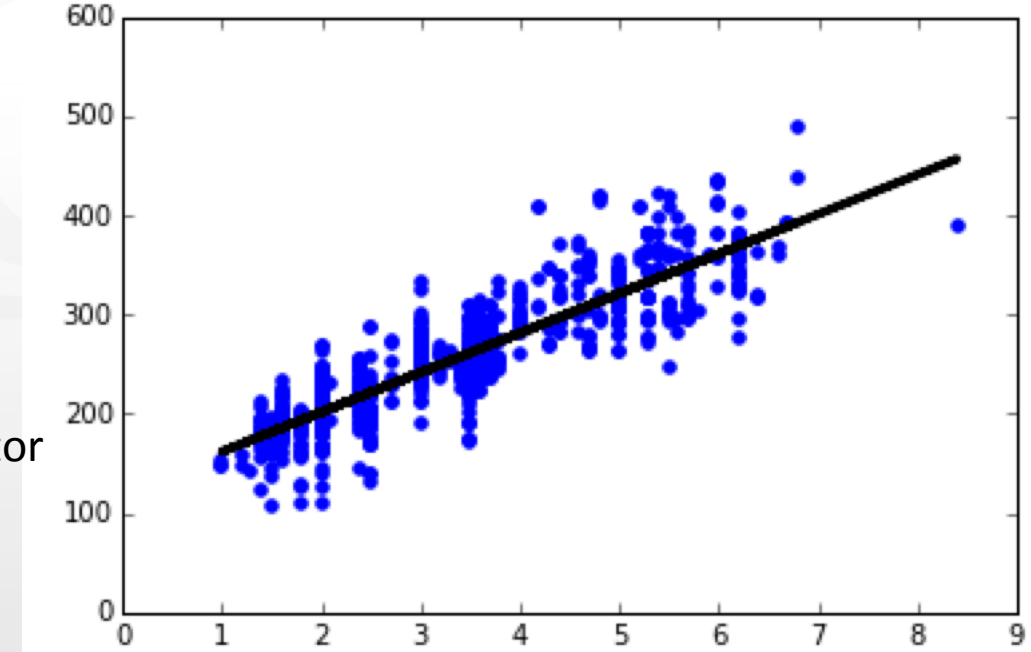
- **What are the applications:**
  - Marketing: sales forecasting
  - Psychology: satisfaction analysis

- Simple Linear Models:
  - target value is expected to be a linear combination of the input variables (straight line)

$$Y = \beta_0 + \beta_1 x_1$$

response variable

a single predictor



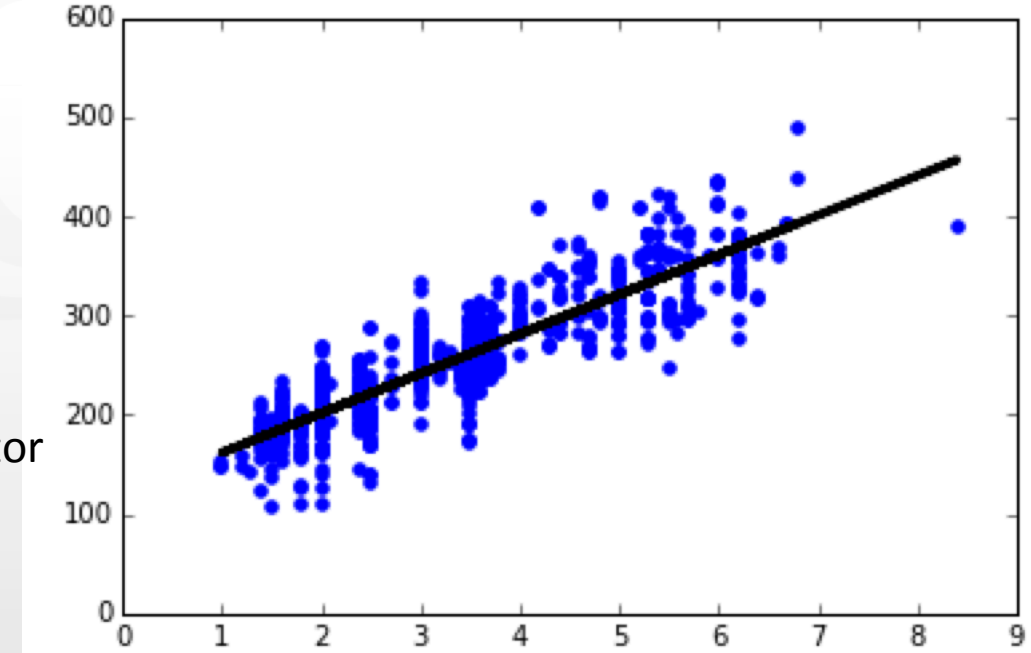
How does it work?

- Regression computes β_i from data to minimize squared error to '**fit**' the data

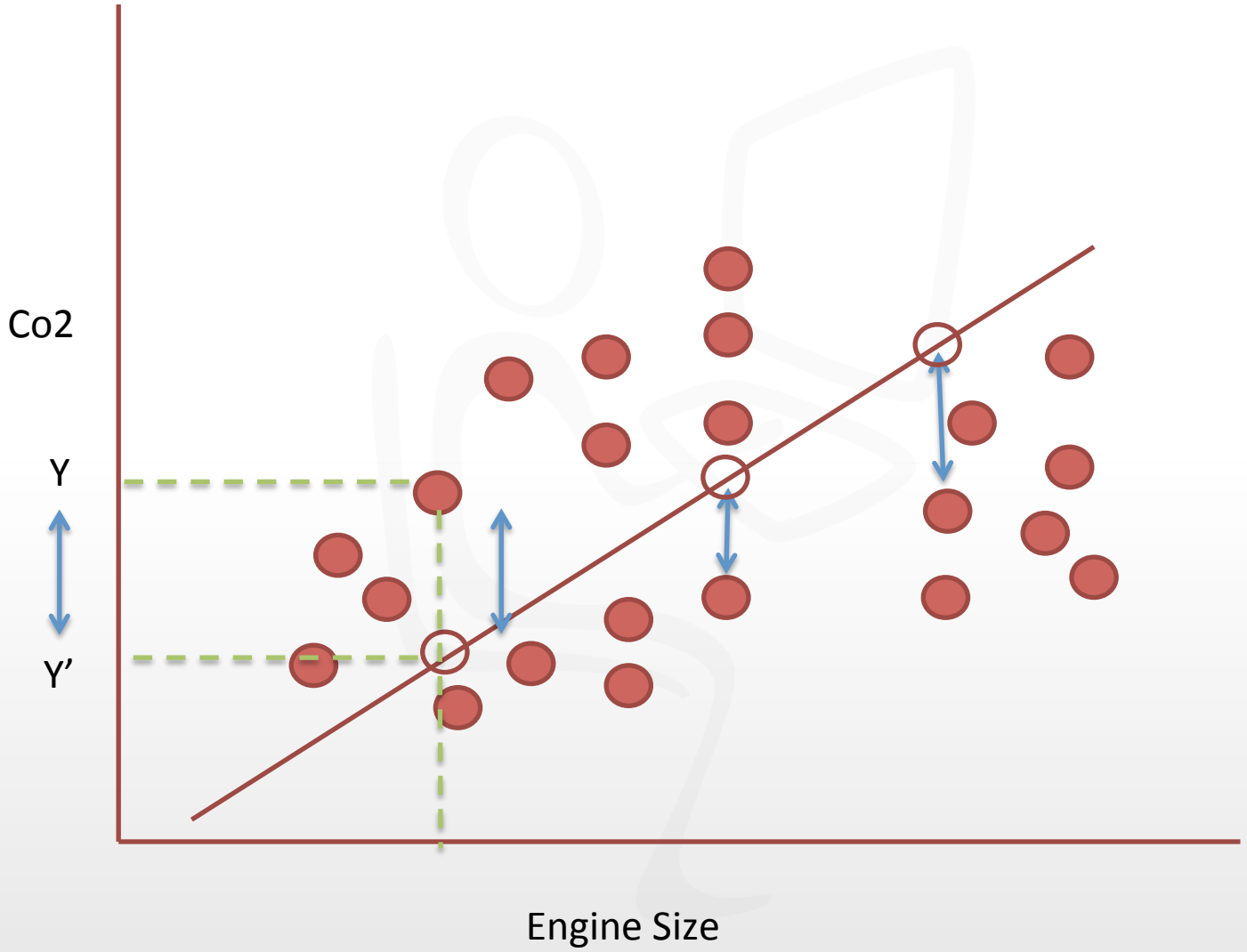
$$Y = \beta_0 + \beta_1 x$$

response variable

a single predictor



CLASSIFICATION: LINEAR REGRESSION



- Linear regression with Least Squares

- LinearRegression fits a linear model with coefficients B = (β_1, β_2,..) to minimize the residual sum of squares between the observed responses (Y) in the dataset, and the responses predicted by the linear approximation (Y').
- Mathematically it solves a problem of the form:

$$\text{Min} \; || X\beta - y||_2^2$$

LINEAR REGRESSION

- **How to find B = (β_1, β_2,..) to minimize the residual sum of squares ?**
  - Method of least squares:
    - estimates the best-fitting straight line

$$Y = \beta_0 + \beta_1 x$$

$$\beta_1 = \frac{\sum_{i=1}^{s} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{s} (x_i - \bar{x})^2}, \qquad \beta_0 = \bar{y} - \beta_1 \bar{x}$$

```
from sklearn import linear_model
regr = linear_model.LinearRegression()
train_x=np.asanyarray(train[['ENGINESIZE']])
train_y=np.asanyarray(train[['CO2EMISSIONS']])
regr.fit (train_x, train_y)
# The coefficients
print 'Coefficients: ', regr.coef_
print 'Intercept: ',regr.intercept_
```

```
Coefficients:  [[ 38.68021623]]
Intercept:  [ 126.61208813]
```

MULTIPLE LINEAR REGRESSION

- $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$

- Y $\Rightarrow$ dependent/target/predicted/response variable
- $x_1, x_2, .. \Rightarrow$ independent/input/feature/*predictor* variable
- $\beta_1, \beta_2, .. \Rightarrow$ Coefficients of dependent variables
- $\beta_0 = \quad \Rightarrow$ intercept
- $\beta_1 > 0 \Rightarrow$ Positive Association
- $\beta_1 < 0 \Rightarrow$ Negative Association
- $\beta_1 = 0 \Rightarrow$ No Association

- Problem? To find the the best fit

MULTIPLE LINEAR REGRESSION

- $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$
  - –Solvable by extension of least square method
  - –Many nonlinear functions can be transformed into the above

NONLINEAR REGRESSION

What is nonlinear models ?

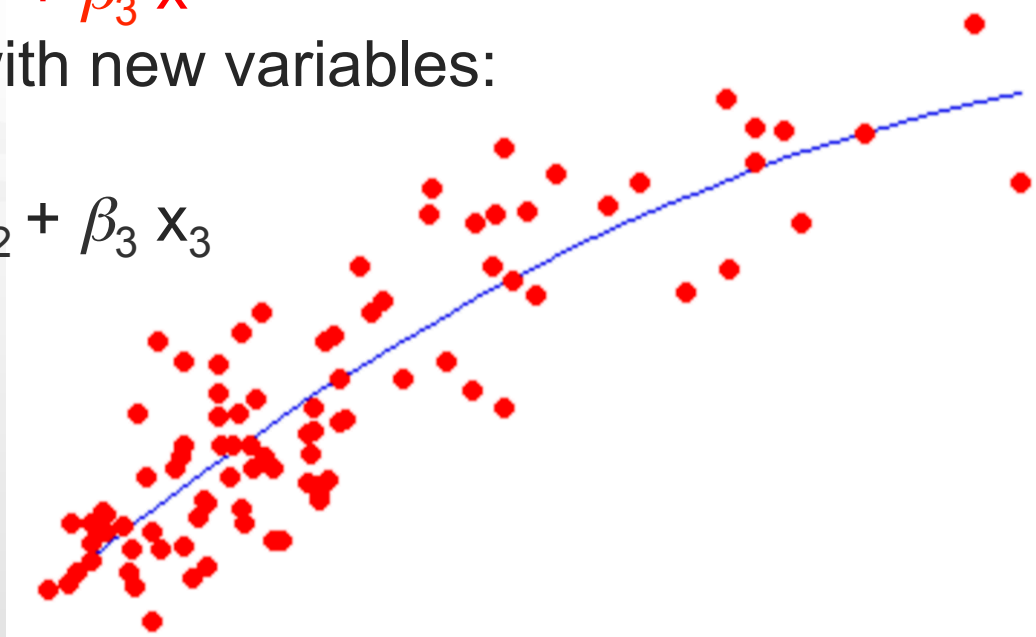
- Some nonlinear models can be modeled by a polynomial function
- A polynomial regression model can be transformed into linear regression model.
- For example,

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

convertible to linear with new variables:

$$x_2 = x^2,\ x_3 = x^3$$

$$y = \beta_0 + \beta_1 x + \beta_2 x_2 + \beta_3 x_3$$



ACCURACY

- Measure predictor accuracy:
  - measure how far off the predicted value (Y') is from the actual known value (Y)
- **Loss function**:
  - measures the error between y_i and the predicted value y_i'
  - Absolute error: $| y_i - y_i' |$
  - Squared error: $(y_i - y_i')^2$

ACCURACY

- Test error (generalization error): the average loss over the test set
  - Mean absolute error: $\frac{\sum_{i=1}^{d} |y_i - y_i'|}{d}$
  - Mean squared error:
  - Relative absolute error: (Residual sum of square) $\frac{\sum_{i=1}^{d} (y_i - y_i')^2}{d}$
  - Relative squared error: $\frac{\sum_{i=1}^{d} |y_i - y_i'|}{\sum_{i=1}^{d} |y_i - \bar{y}|}$
  - Relative squared error: $\frac{\sum_{i=1}^{d} (y_i - y_i')^2}{\sum_{i=1}^{d} (y_i - \bar{y})^2}$
  - R^2=1-Relative squared error

```
test_x=np.asanyarray(test[['ENGINESIZE']])
test_y=np.asanyarray(test[['CO2EMISSIONS']])
test_y_=regr.predict(test_x)


print("Residual sum of squares: %.2f"
      % np.mean((test_y_ - test_y) ** 2))
# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % regr.score(test_x, test_y))
# Plot outputs
plt.scatter(test_x, test_y,  color='blue')
plt.plot(test_x, test_y_, color='black', linewidth=3)
plt.xlabel("Engine size")
plt.ylabel("Emission")
plt.show()
```

```
Residual sum of squares: 874.93
Variance score: 0.81
```

EVALUATION METHOD

- Holdout method
  - Given data is randomly partitioned into two independent sets
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation
  - Random sampling: a variation of holdout
    - Repeat holdout k times, accuracy = avg. of the accuracies obtained
- Cross-validation (*k*-fold, where k = 10 is most popular)
  - Randomly partition the data into *k mutually exclusive* subsets, each approximately equal size
  - At *i*-th iteration, use D_i as test set and others as training set
  - Leave-one-out: k folds where k = # of tuples, for small sized data
  - Stratified cross-validation: folds are stratified so that class dist. in each fold is approx. the same as that in the initial data

DEMO

DIMENSION REDUCTION

Business Understanding

Dimension Reduction

Analytic Approach

Data Requirements

Data Collection

Data Understanding

Prediction
Interpretation
Justification
Testing

Feedback

Deployment

Evaluation

PCA

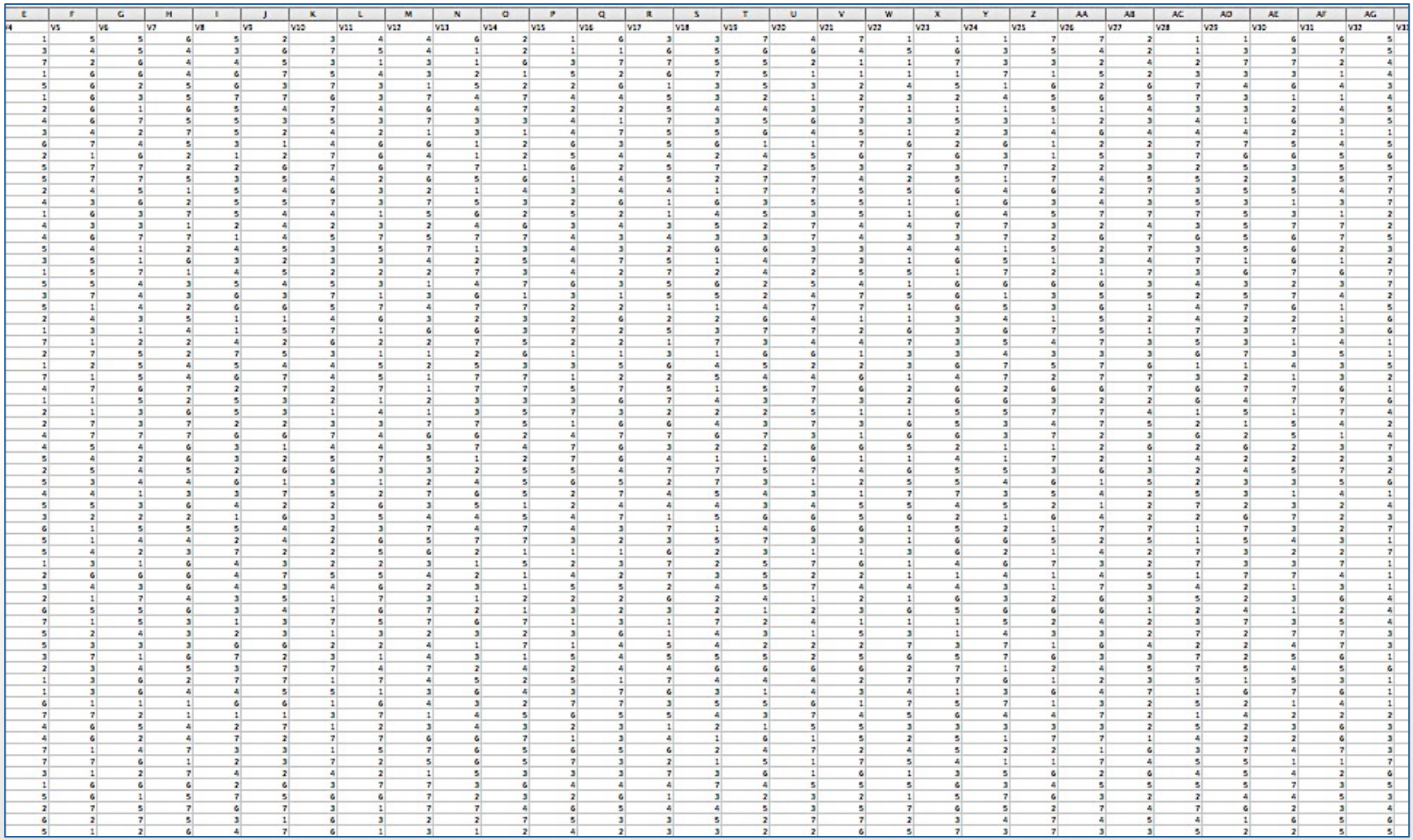
Modeling

Data Preparation

DIMENSION REDUCTION PROBLEM

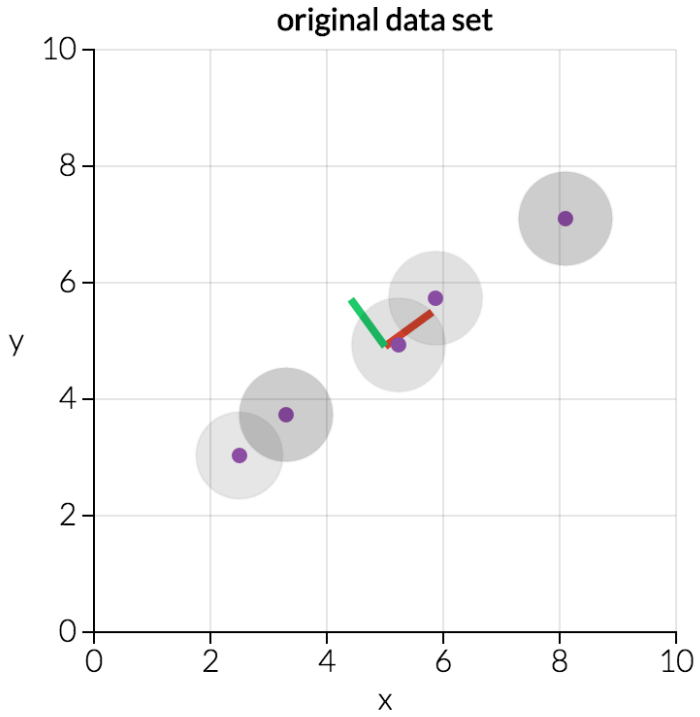
Lots of Variables

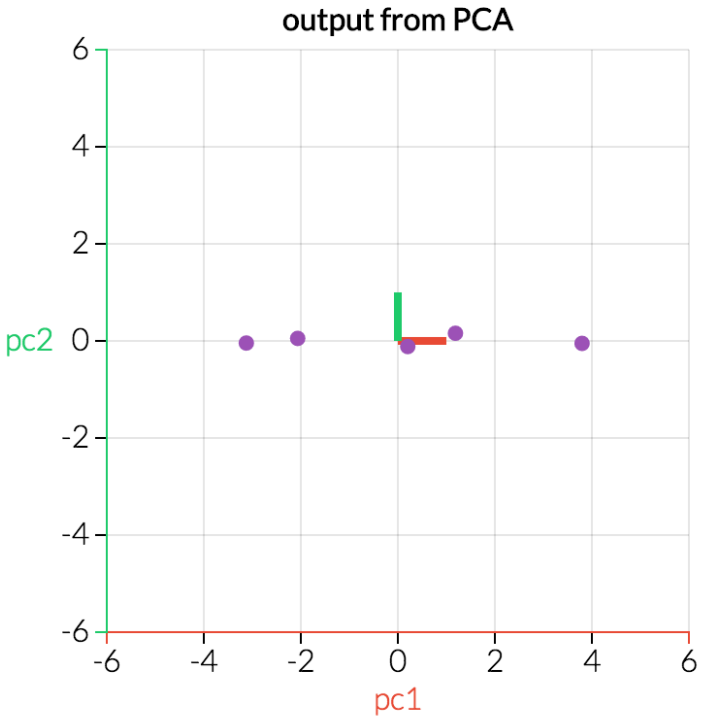
Lots of Observations



Created by: Konstantin Tskhay

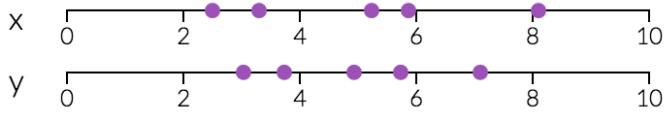
2 DIMENSIONS → 1 DIMENSION

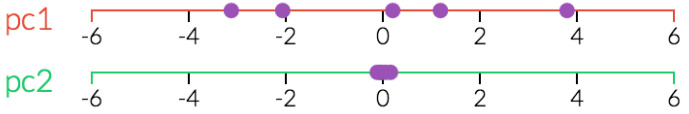




PCA is useful for eliminating dimensions. Below, we've plotted the data along a pair of lines: one composed of the x-values and another of the y-values.

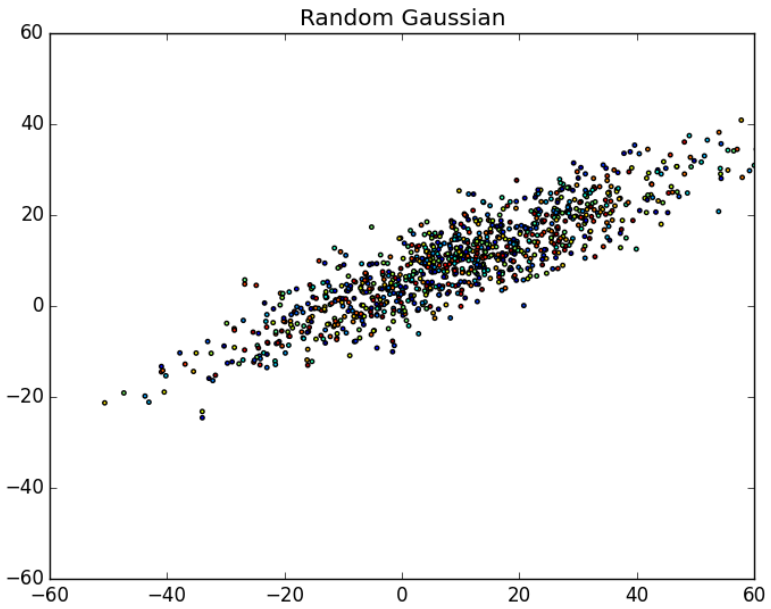
If we're going to only see the data along one dimension, though, it might be better to make that dimension the principal component with most variation. We don't lose much by dropping PC2 since it contributes the least to the variation in the data set.

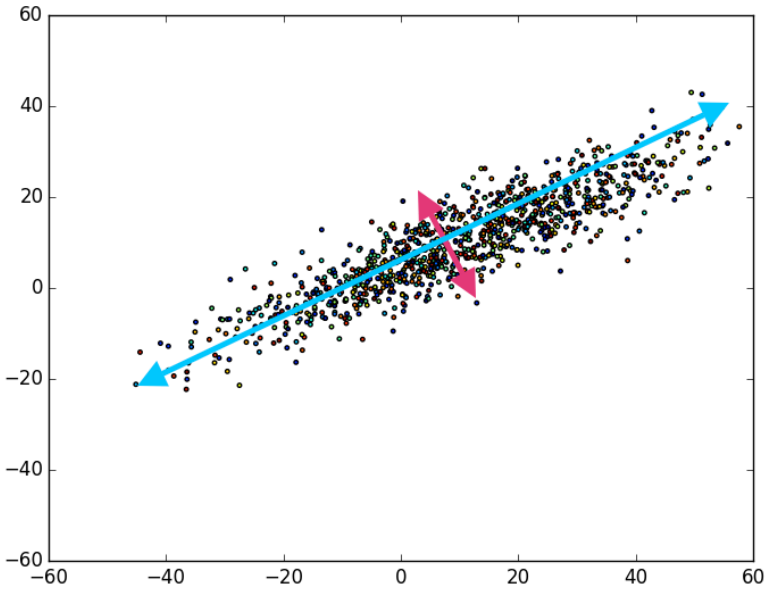




http://setosa.io/ev/principal-component-analysis/

2 DIMENSIONS → 1 DIMENSION





http://setosa.io/ev/principal-component-analysis/

PRINCIPAL COMPONENT ANALYSIS (PCA)

- *A method for reducing your data into the fewest "principal components" that simultaneously maximize the amount of variance explained*

- **Goal: use d number of components to explain most of the variance from n features**

DOING PCA

- Correlation captures the covariation between two vectors
- Correlations between multiple variables can be combined into a matrix of correlations
- "Eigenvectors" and "eigenvalues" are calculated for the matrix
- **Eigenvectors** are the Principal Components
- **Eigenvalues** are the amount of variance the Principal Components explain

Created by: Konstantin Tskhay

HOW MANY COMPONENTS?

- **Kaiser-Guttman Rule**
  - The number of factors extracted equals the number of factors with eigenvalues greater than 1
- **Percentage of Common Variance**
  - The number of factors retained should have a **cumulative variance** explained should be at least 50% of the variance, but most people go for 75% and the ideal is 90%
- **Scree Test**
  - The number of factors retained should be the last number before the rate of change in eigenvalues levels off

Created by: Konstantin Tskhay

CURRENT DATA

- **Question:**

  Do students' evaluations cluster together into components?

- **Data:**

  Aggregate students' evaluations across 6 dimensions:

  1. Present
  2. Explain
  3. Communicate
  4. Teach
  5. Workload
  6. Difficulty

Created by: Konstantin Tskhay

DEMO

CLUSTER ANALYSIS

Business Understanding

Cluster Analysis

Analytic Approach

Data Requirements

Data Collection

Data Understanding

Prediction
Interpretation
Justification
Testing

Feedback

Deployment

Evaluation

K-means

Modeling

Data Preparation

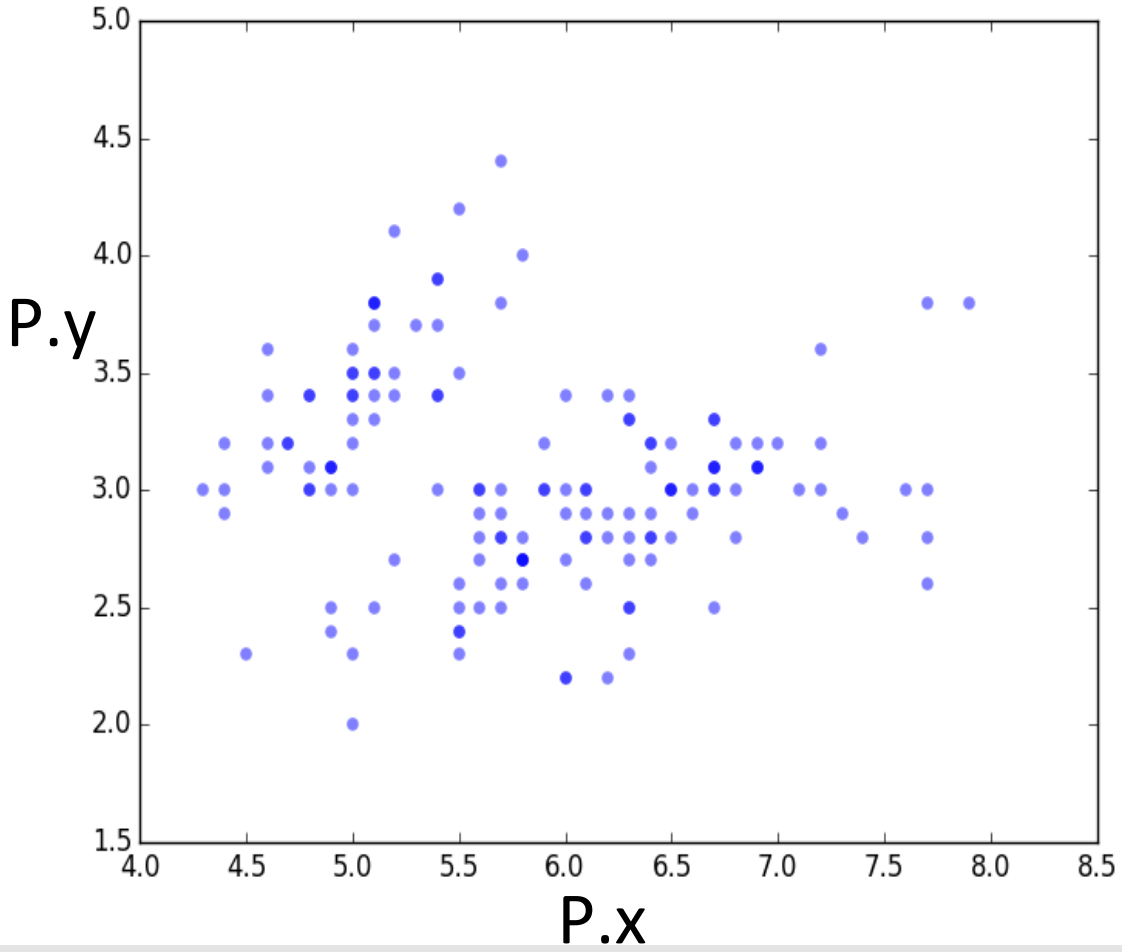
CLUSTER ANALYSIS

- What is a cluster?
  - A group of data points that are similar to other data points in the cluster, and dissimilar to data points in other clusters

CLUSTERING

Point1(2,3)
Point1(3,2)
Point1(1,1)
…

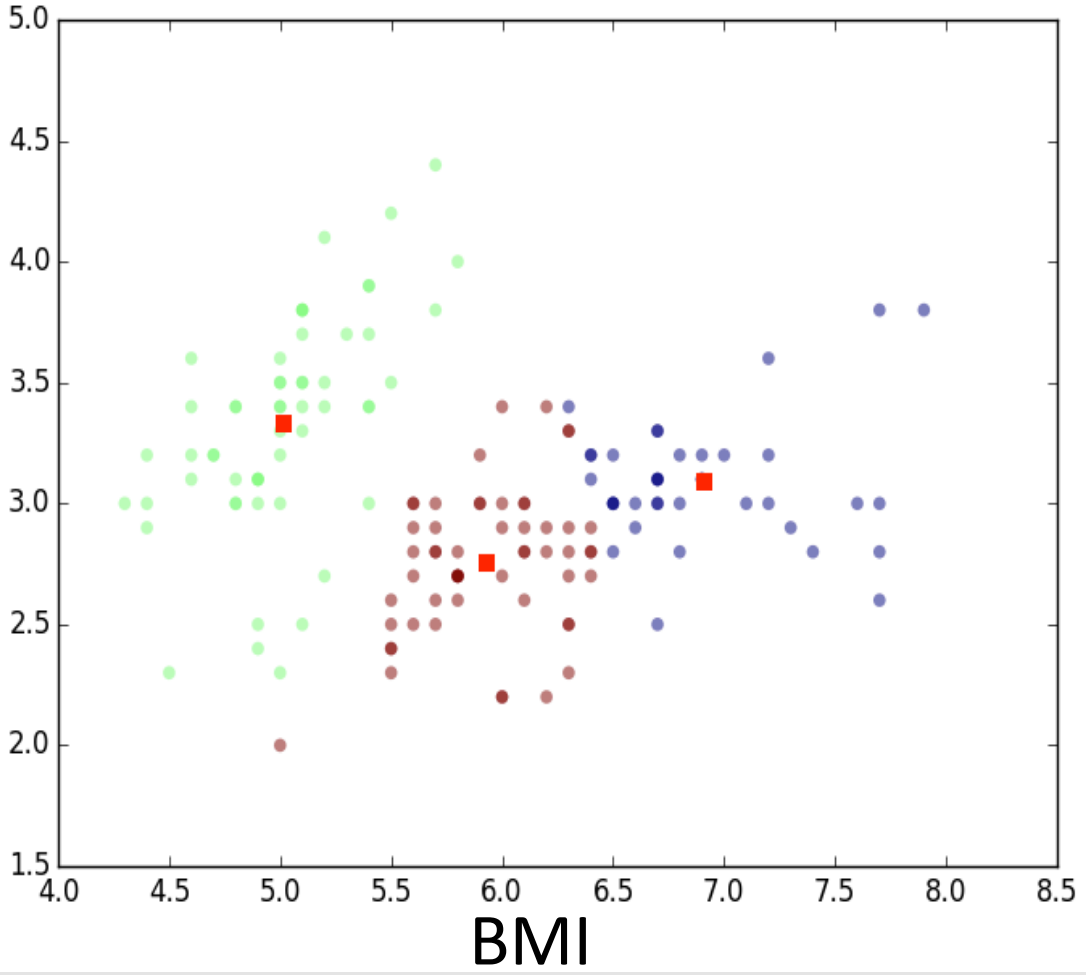
P1.x	P2.y
2	3
3	2
1	1
1	2
…	…
…	…



CLUSTERING

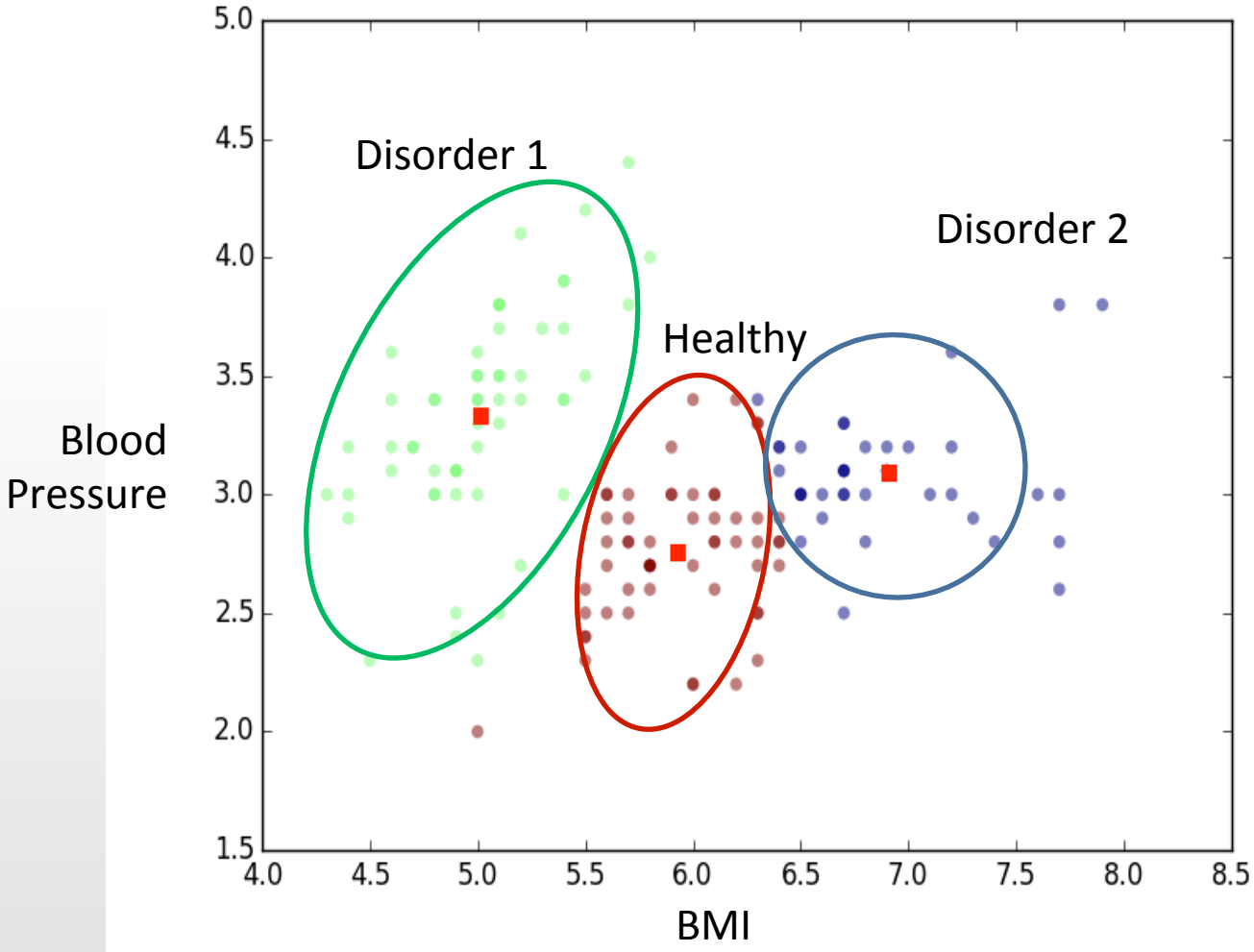
- Unsupervised learning

Blood Pressure

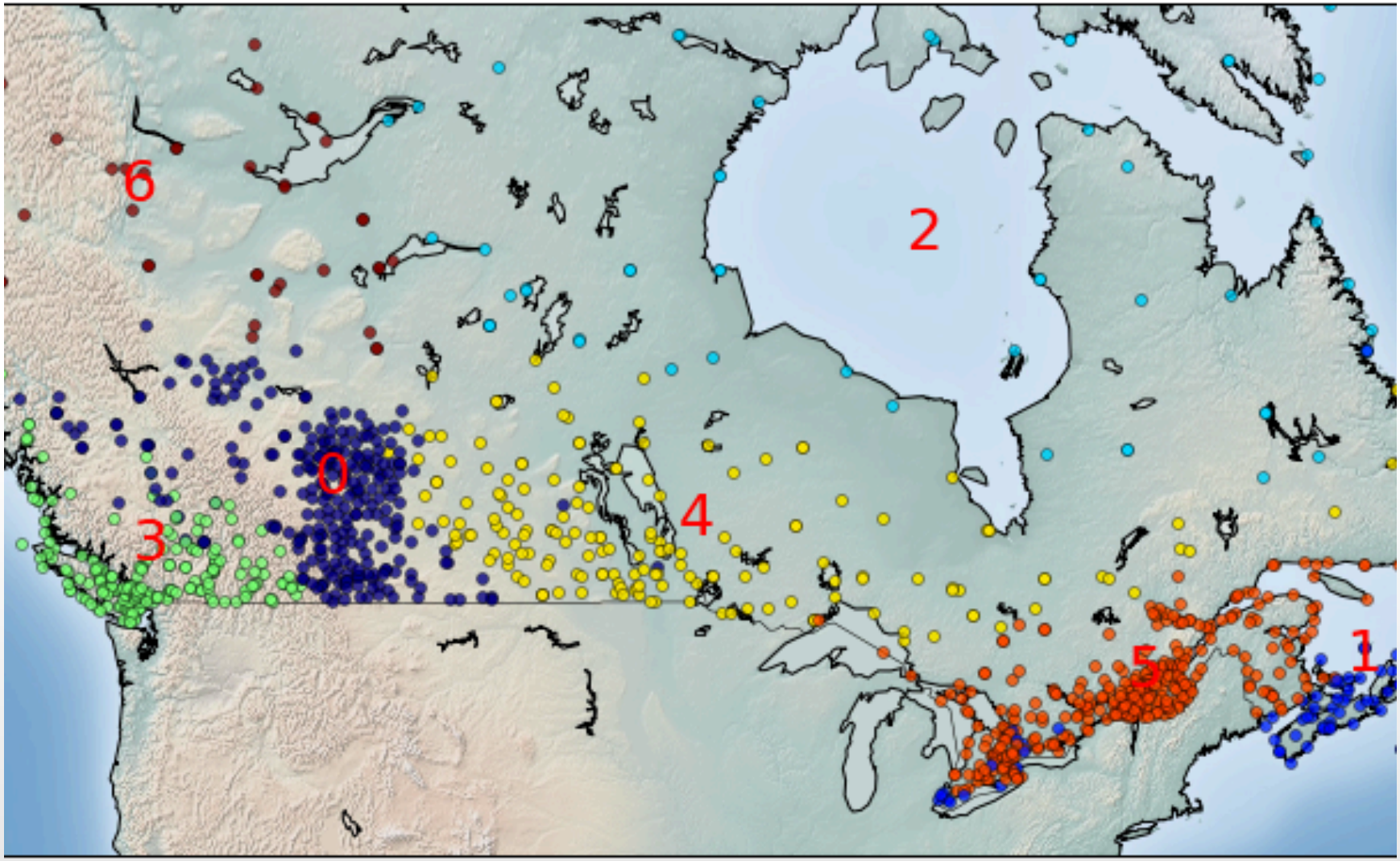


CLUSTERING

- What is the usage of Clustering?



WEATHER STATION CLUSTERING



CLUSTERING APPLICATIONS

- RETAIL/MARKETING:
  - –Finding cluster of customers based on their demographic characteristics, preferences, or their buying patterns
    - To recommend a new book, or to new customer by identifying clusters of books or clusters of customer preferences

CLUSTERING APPLICATIONS

- BANKING:
- Clustering of normal transactions to find the patterns of fraudulent credit card use
- Identifying clusters of customers
  - e.g.,Customers with both business and personal accounts; unusually high percentage of loyalty
- Determining credit card spending by customer groups

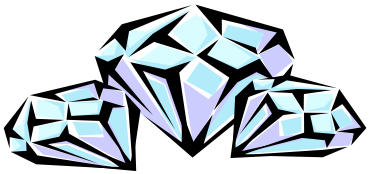
CLASSIFICATION VS CLUSTERING

BMI	Age	BP	…	Label
2	3	3	…	?
1	4	1	…	?
1	2	2	…	?
…	…	…	…	..

← Cluster A

← Cluster B

← Cluster B



K-means

CLASSIFICATION VS CLUSTERING

Labeled dataset:

BMI	Age	BP	…	Label/ Actual val
2	3	3	…	disorder1
1	4	1	…	disorder2
1	2	2	…	healthy
…		…	…	…

Decision Tree

BMI	Age	BP	…	Label
4	3	3	…	?
5	2	1	…	?
1	4	2	…	?
…	…	…	…	…

← disorder2

← healthy

← disorder1

Testing dataset:

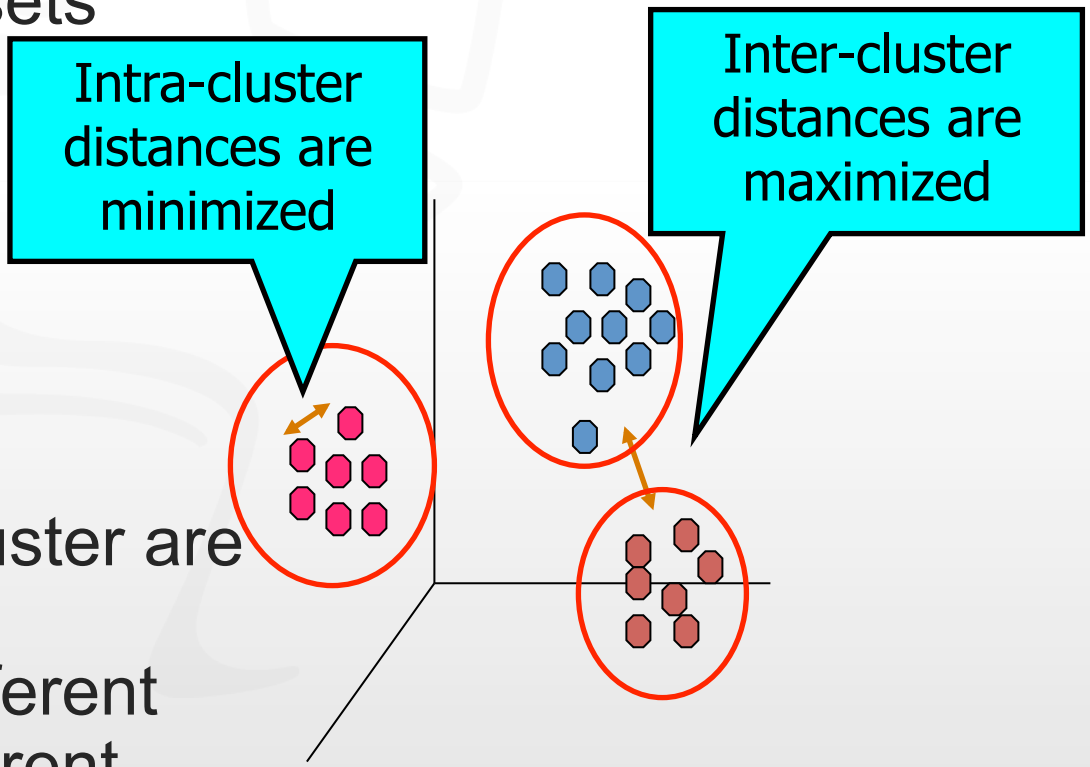
CLUSTERING ALGORITHMS

- **What kind of clustering algorithms?**
- k-means Clustering
- Fuzzy Clustering
- Hierarchical Clustering
- Density based Clustering

K-MEANS

what is k-Means?

- Partitioning Clustering
- K-Means divides the data into non-overlapping subsets (clusters) without any internal structure

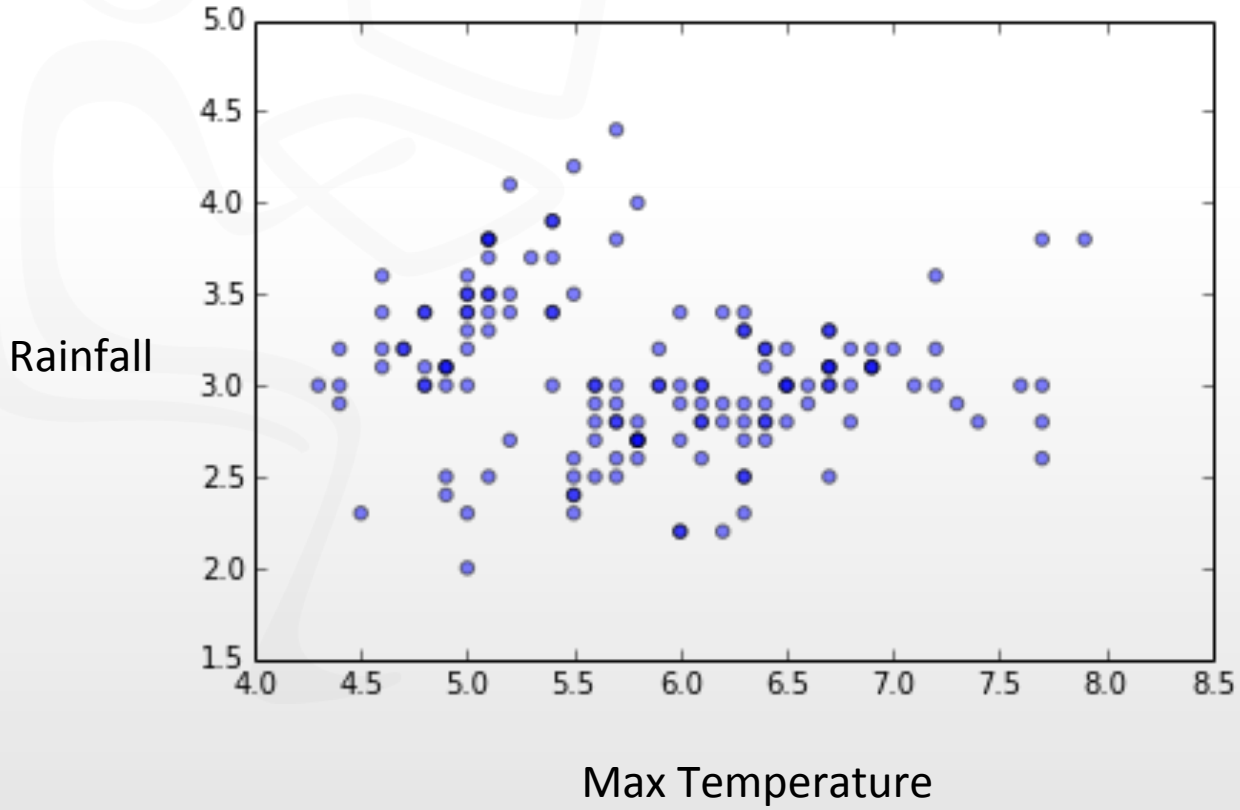


- Examples within a cluster are very similar
- Examples across different clusters are very different

K-MEANS CLUSTERING

1) Load your data

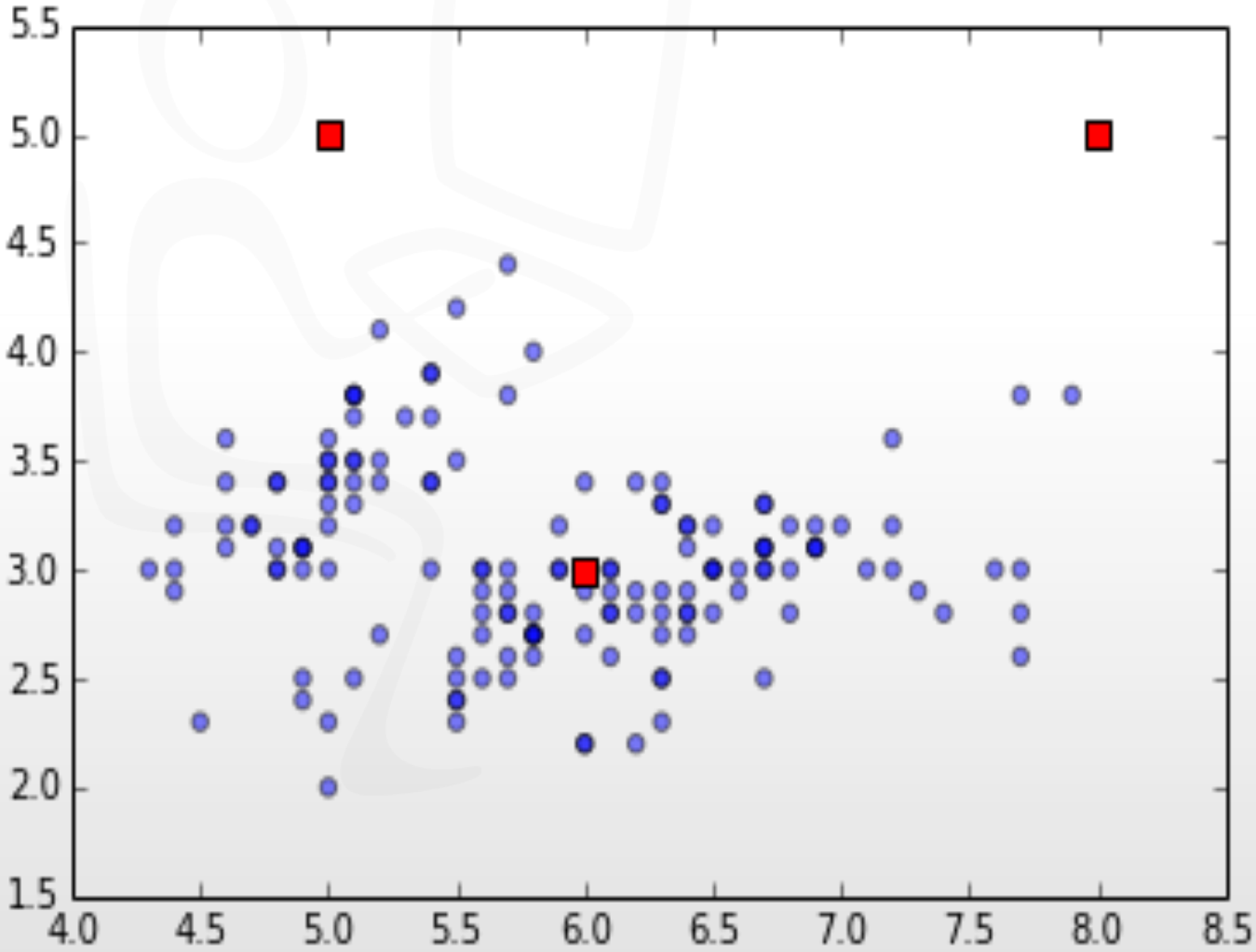
Rainfall	Max Temp
3 mm	4
2mm	6
3.5mm	2
…	..



K-MEANS CLUSTERING

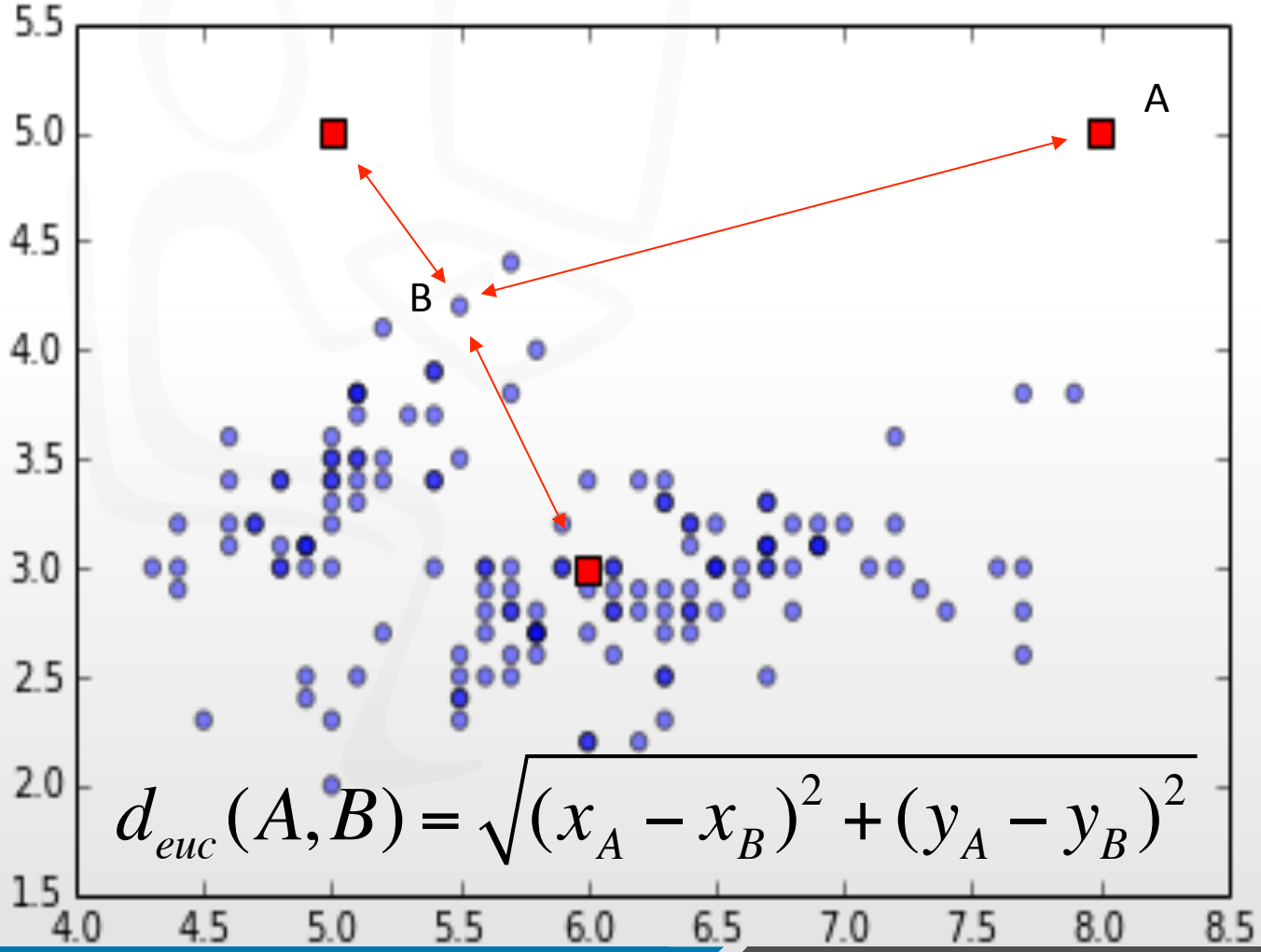
2) Initialize k=3 centroids randomly

[[8., 5.],
[5., 5.],
[6., 3.]]



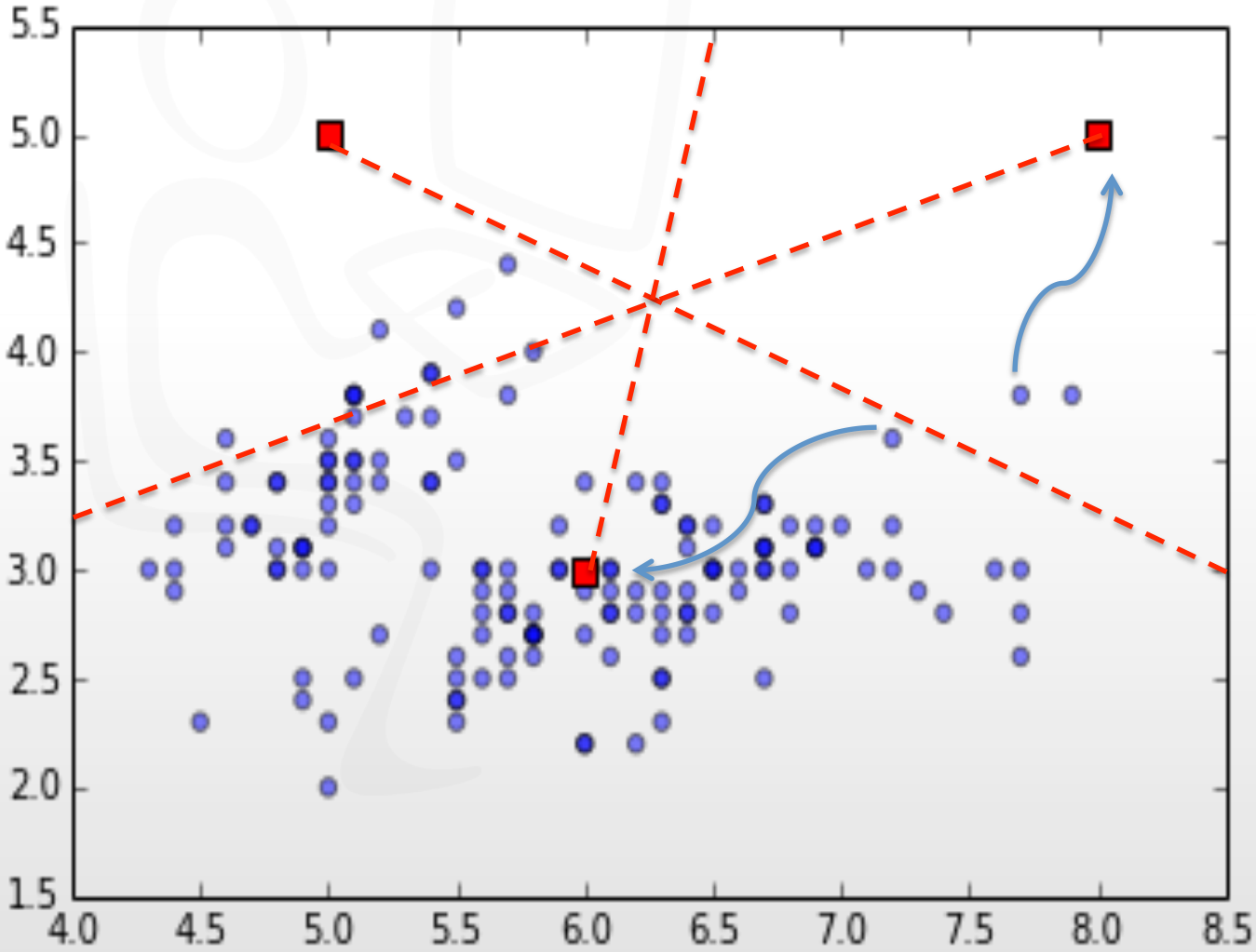
K-MEANS CLUSTERING

3) Calculate distance of all points from centroids
Similarity=1-distance



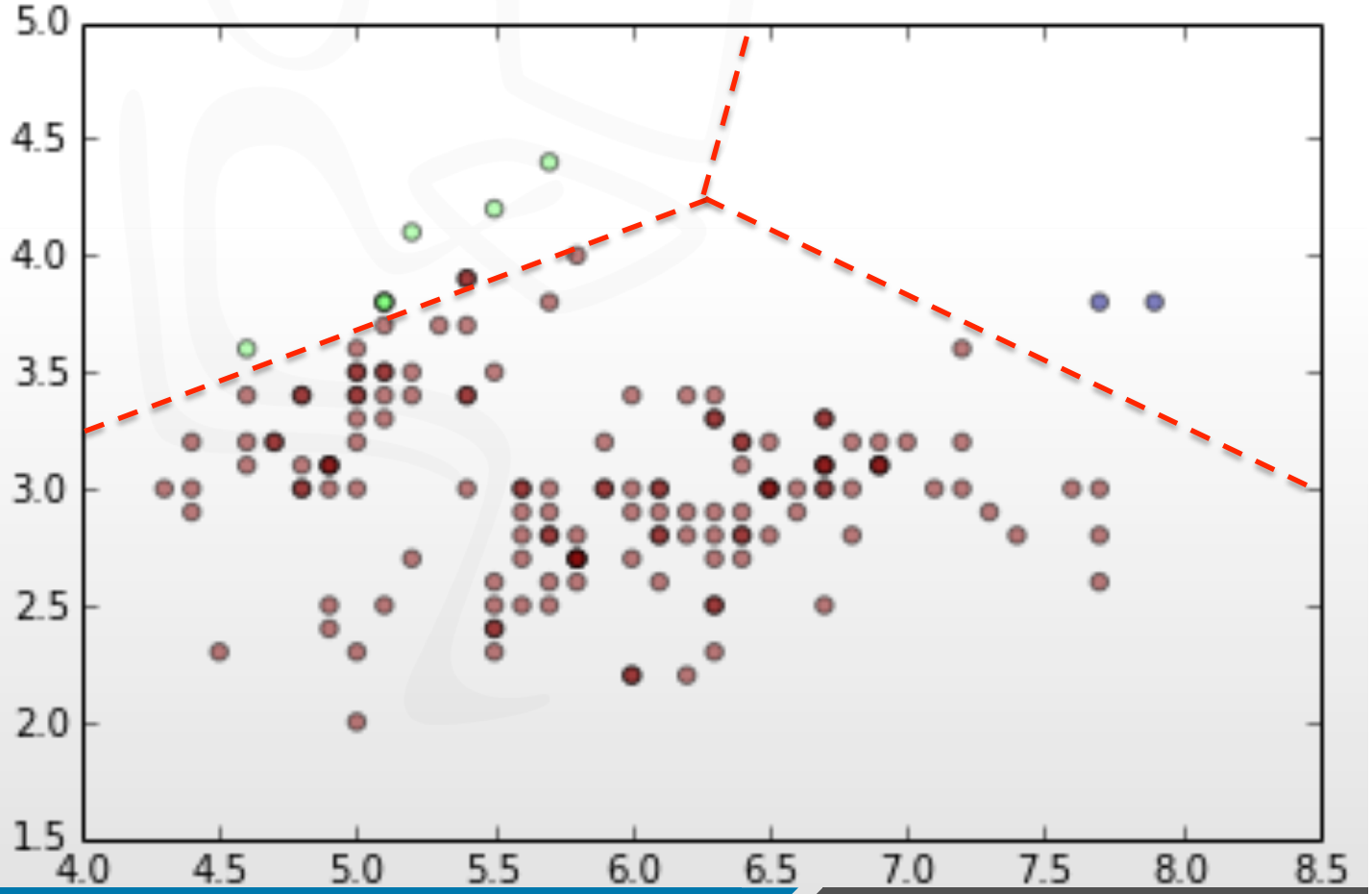
K-MEANS CLUSTERING

4) Find the closest center to each data point



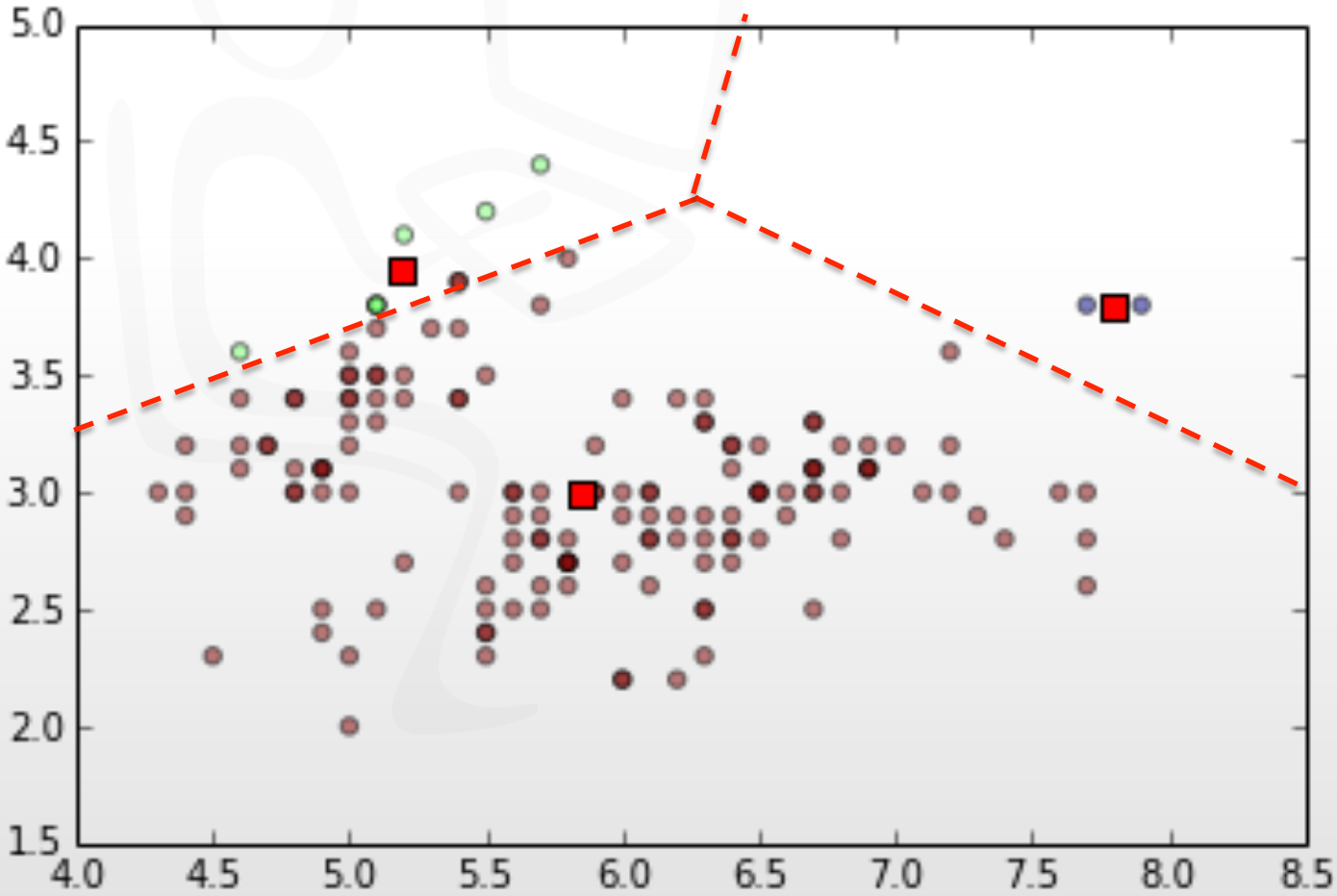
K-MEANS CLUSTERING

5) Assign each point to the closest centroid

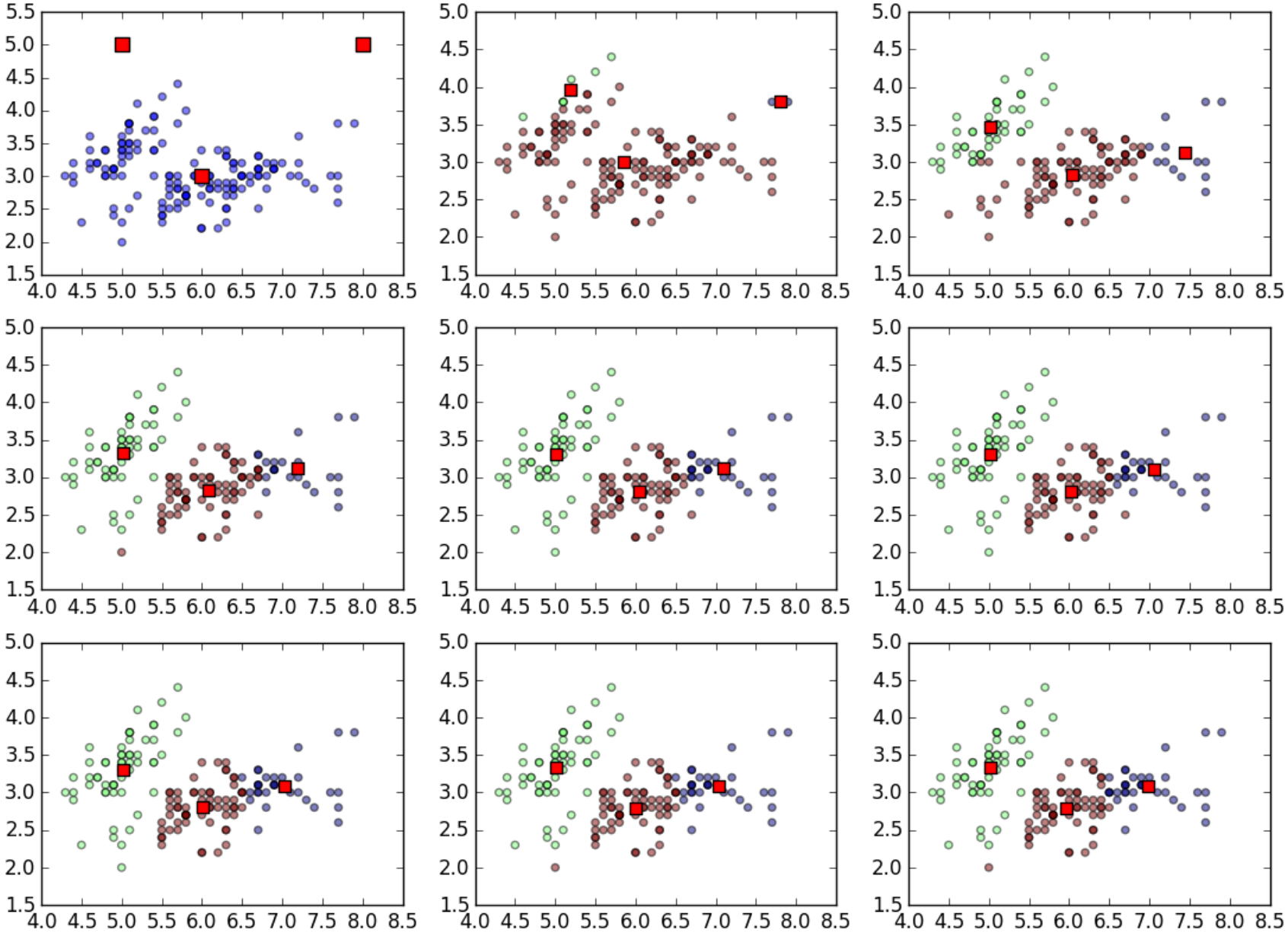


K-MEANS CLUSTERING

6) Compute the new centroids for each cluster.



7) Repeat until there are no more changes



THE MAIN CLUSTERING ALGORITHMS

- Partition based (K-means)
  - –Med and Large sized databases (*Relatively efficient*)
  - –Produces sphere-like clusters
  - –Needs number of clusters (K)

- Partition based (FCM)
  - –Produces Fuzzy clusters
  - –Long computational time

THE MAIN CLUSTERING ALGORITHMS

- Hierarchical based(Agglomerative)
  - –Produces trees of clusters
- Density based (DBScan)
  - –Produces arbitrary shaped clusters
  - –Good when dealing with spatial clusters (maps)

RESOURCES

Course: bit.ly/bduclustering

- Overview of Clustering Algorithms
  - http://homepages.inf.ed.ac.uk/rbf/BOOKS/JAIN/Clustering_Jain_Dubes.pdf
- Evaluation of clustering
  - http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html
- Cluster Analysis in R
  - http://www.stat.berkeley.edu/~spector/s133/Clus.html
- Cluster Analysis in SPSS
  - http://studysites.uk.sagepub.com/burns/website%20material/Chapter%2023%20-%20Cluster%20Analysis.pdf
- Cluster Analysis in Python (scikit-learn)
  - http://scikit-learn.org/stable/modules/clustering.html

DEMO