ISYE6501x Mid Term Quiz 1 Revision

Course Structure

Knowledge Building

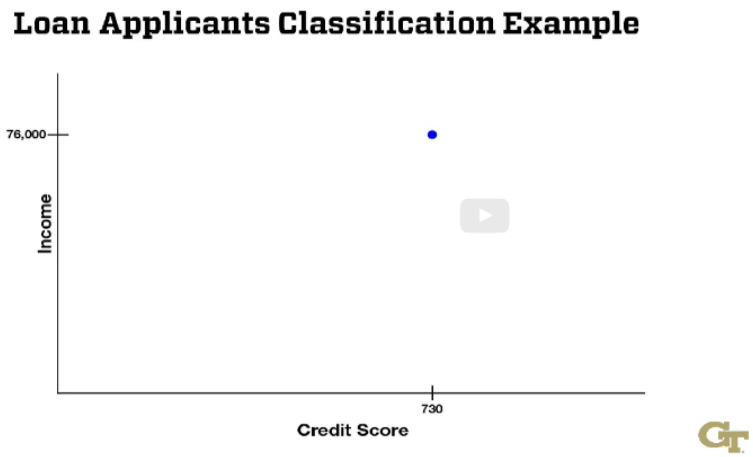
Module 1: Introduction
Module 2: Classification
Module 3: Validation
Module 4: Clustering
Module 5: Basic Data Preparation
Module 6: Change Detection
Module 7: Exponential Smoothing
Module 8: Basic Regression
Module 9: Advanced Data Preparation
Module 10: Advanced Regression
Module 11: Variable Selection
Module 12: Design of Experiments
Module 13: Probability-Based Models
Module 14: Missing Data
Module 15: Optimization
Module 16: Advanced Models

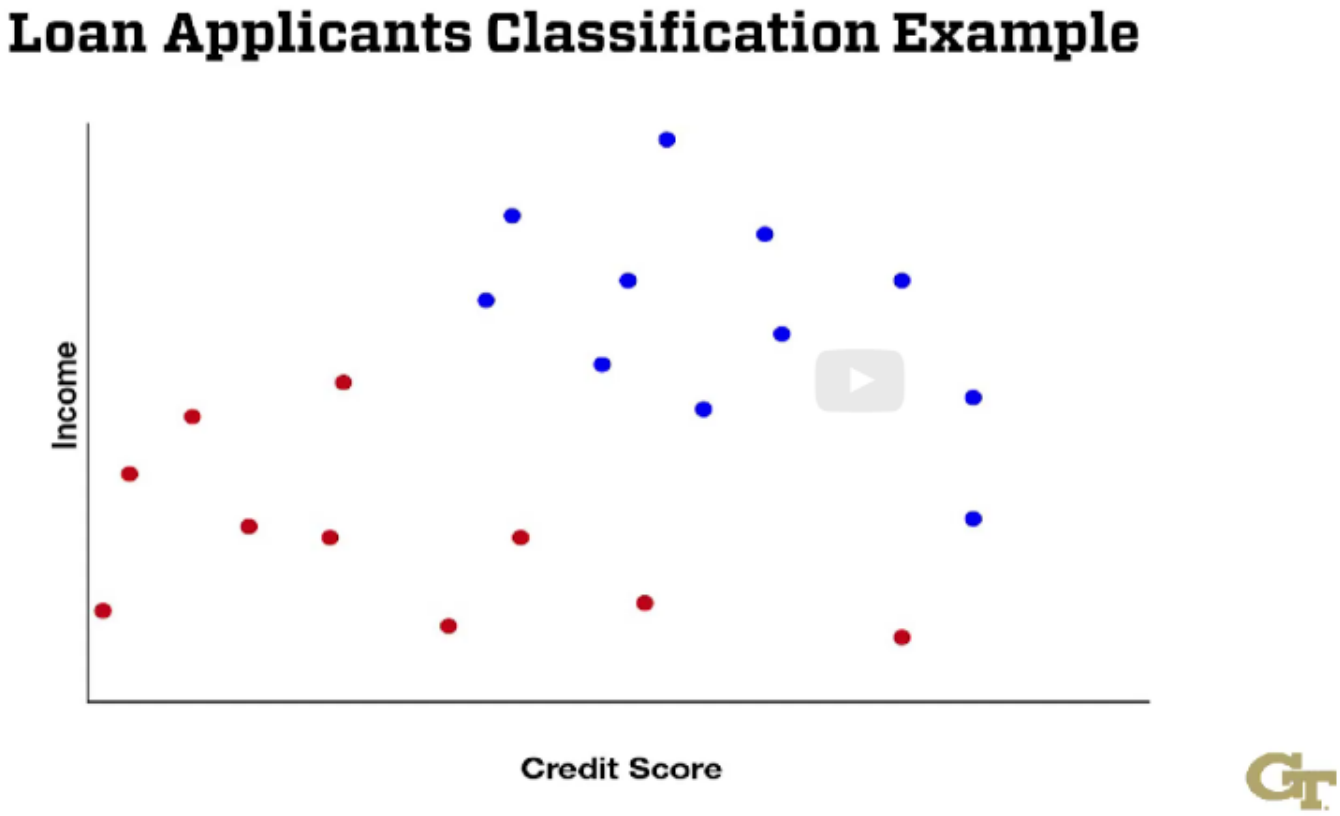
Module 1: Introduction

Nothing much here. Just course introductions.

Module 2: Classification

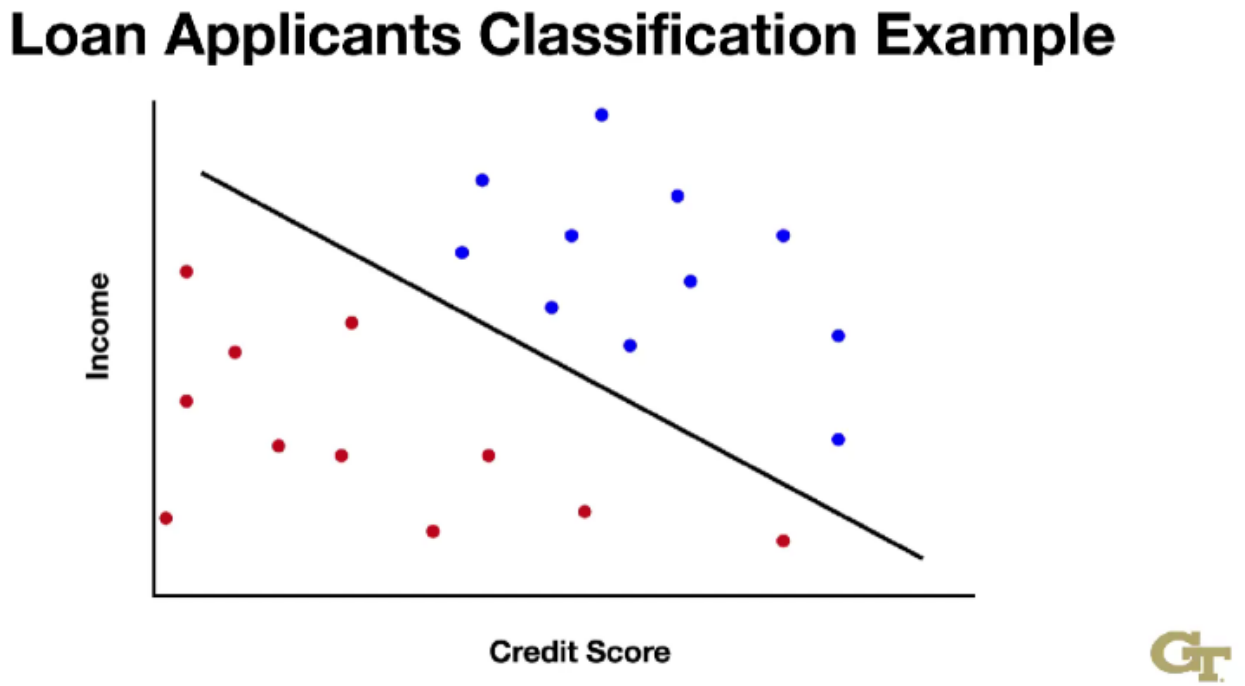
Lesson 2.1: Introduction to Classification





Blue - Loan Repaid
Red - Defaulted

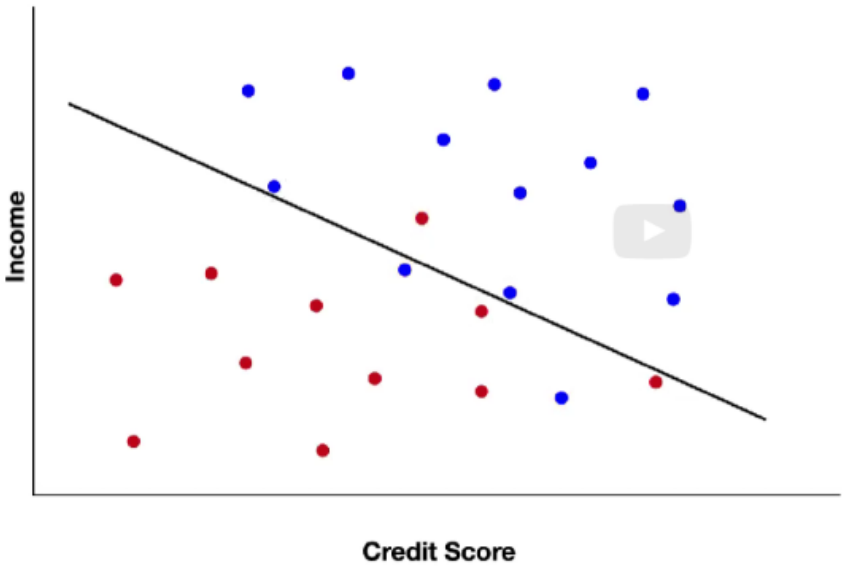
Lesson 2.2 (M): Choosing a Classifier



We can see where the new applicant's data is relative to the line, and classify it accordingly.

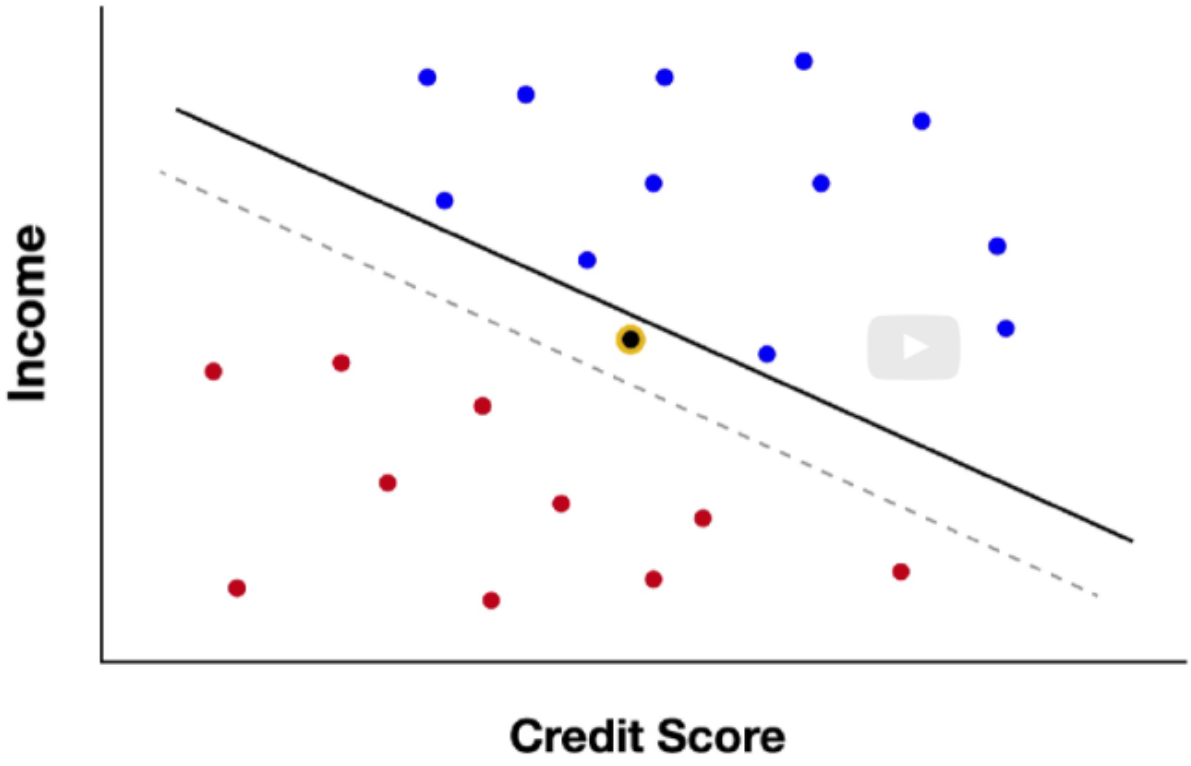
A Soft Classifier is used when you cannot perfectly separate the points.

Loan Applicants Classification Example



In most cases we are unable to perfectly separate the two classes. So we pick a (soft) classifier that minimizes the number of incorrectly classified points.

We have to weigh the cost of actual mistakes and near mistakes.



Let's say that the cost of making a bad loan is twice as high as the cost of turning away a good loan, we should shift the line so it is closer to the blue points than to the red points.

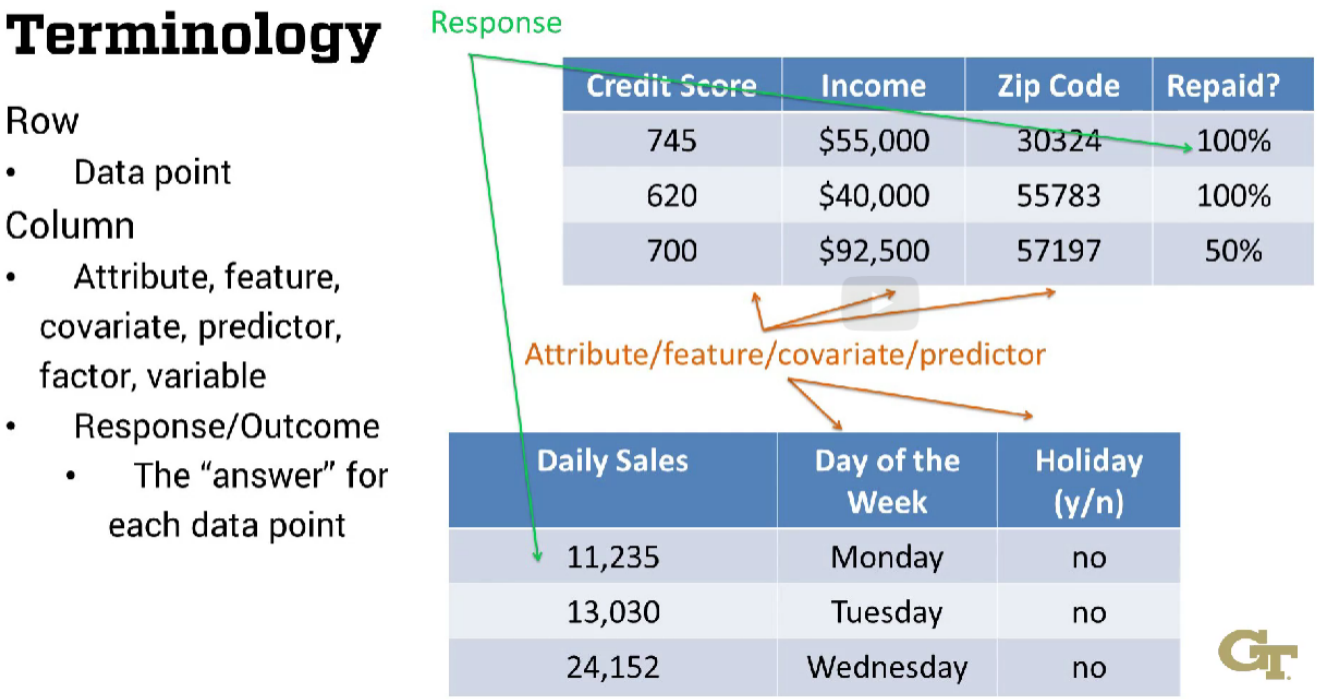
Given that realistically it is impossible to separate with no mistakes, we might be more willing to accept one type of mistake than another.

Lesson 2.3 (C): Data Definitions

Row: Data Point (A data point is all the information about one observation)

Column:

- Attribute, feature, covariate, predictor, factor, variable
- Response/Outcome (the "answer" for each data point



Structured Data

Data that can be stored in a structured way

- Quantitative: credit score, age, sales, etc
- Categorical: M/F, Hair Colour, etc

Example: The amount of money in a person's bank account

Unstructured Data

- data not easily described and stored
- example: Written text

Example: The contents of a person's Twitter feed

Time Series Data

- same data recorded over time
- often recorded in equal intervals (doesn't have to be)

- Eg: Daily sales, stock prices, child's height on each birthday, The average cost of a house in the United States every year since 1820

Lesson 2.4 (M): Support Vector Machines (SVM)

SVM is a type of Classification Models

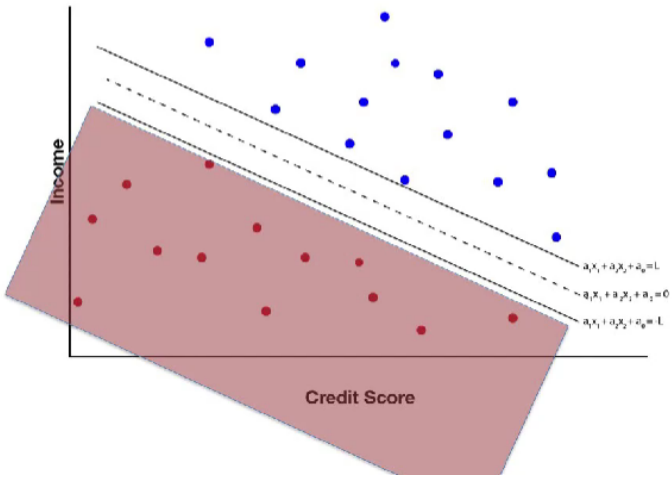
Extra reading on SVM Classifier (http://pyml.sourceforge.net/doc/howto.pdf (http://pyml.sourceforge.net/doc/howto.pdf))

Blue points:

$$a_1x_{i1} + a_2x_{i2} + \ldots + a_nx_{in} + a_0 \geq 1$$

Red points:

$$a_1x_{i1} + a_2x_{i2} + \ldots + a_nx_{in} + a_0 \leq -1$$



m = number of data points

n = number of attributes

x_{ij} = jth attribute of ith data point

x_{i1} = credit score of person i

x_{i2} = income of person i

y_i = response for data point i

$$y_i = \begin{cases} 1, \text{ if data point i is blue} \\ -1, \text{ if data point i is red} \end{cases}$$

Line

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n + a_0 = 0$$

$$\sum_{j=1}^{n} a_jx_j + a_0 = 0$$

We want to find values of a_0, a_1 up to a_n that classify the points correctly and have the maximum gap or margin between the parallel lines.

Since we defined y_i to be 1 for blue points and negative 1 for red points, we can combine these two expressions to get the following:

All points:

$$(a_1x_{i1} + a_2x_{i2} + \ldots + a_nx_{in} + a_0)y_i \geq 1$$

The above inequality will hold true in the case of a correct classification, ie. when a data point is on the correct side of the line.

We need to **maximise** the margin of separation (distance) between both parallel lines in the classifier, which means the following:

$$\text{Distance between solid lines} = \frac{2}{\sqrt{\sum_j (a_j)^2}} \quad \text{So, } \underset{a}{\text{Minimize}} \sum_j (a_j)^2$$

The above is basically the Euclidean (Orthogonal) Distance between the two parallel lines.

$$\underset{a_0,\ldots,a_n}{\text{Minimize}} \sum_{j=1}^{n} (a_j)^2$$

Subject to

$$(a_1 x_{i1} + a_2 x_{i2} + \ldots + a_n x_{in} + a_0) y_i \geq 1$$

for each data point i

As mentioned above, the following inequalities will hold true:

Correct side of the line:

$$\left(\sum_{j=1}^{n} a_j x_{ij} + a_0 \right) y_i - 1 \geq 0$$

Wrong side of the line:

$$\left(\sum_{j=1}^{n} a_j x_{ij} + a_0 \right) y_i - 1 < 0$$

The error for the data point *i* is as follows:

Error for data point i:

$$\max\left\{0,1-\left(\sum_{j=1}^{n} a_j x_{ij} + a_0\right) y_i\right\}$$

The total error we want to minimise can be written as the sum over all data points i of the following:

Total error:

$$\sum_{i=1}^{m} \max\left\{0,1-\left(\sum_{j=1}^{n} a_j x_{ij} + a_0\right) y_i\right\}$$

We experience a tradeoff between the **ERROR** and **MARGIN** as can be seen below:

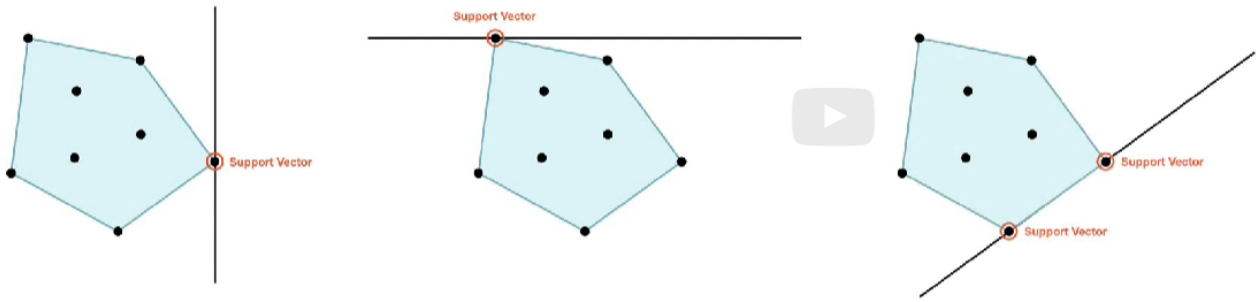
$$\underset{a_0,\ldots,a_n}{\text{Minimize}} \sum_{i=1}^{m} \max\left\{0,1-\left(\sum_{j=1}^{n} a_j x_{ij} + a_0\right) y_i\right\} + \lambda \sum_{j=1}^{n} (a_j)^2$$

We can pick a value of Lambda (durin hyperparameter tuning) and minimise the combination of error minus margin.

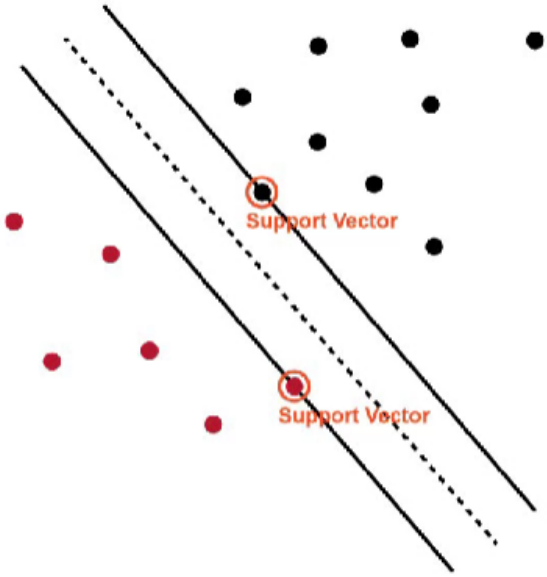
Question: In Lesson 2.4

- why did they say at 6:09 that "the margin we want to maximise is the sum of a_ij squared"? Shouldn't it be that when we want to maximise the margin, we should then minimise a_ij squared?
- "as lambda gets large, this term gets large, so the importance of a larger margin outweighs avoiding mistakes in classifying known data points" isn't the sum of a_ij just the denominator and not the actual distance between the two parallel lines?
- it seems to contradict what is said in Lesson 2.6

Lesson 2.5 (M): SVM: What the Name Means



- Point that holds up shape = support vector
  - Support vectors can support sides, top, etc.



- Support Vector Machine model
  - Determines “support vectors”
  - Automatically from data (hence, “machine”)

The **classifier** it returns is actually not one of the lines touching a support vector.

Lesson 2.6 (M): Advanced SVM

Hard Margin

$$\underset{a_0,\ldots,a_m}{\text{Minimize}} \sum_{i=1}^{m} (a_i)^2$$

Subject to

$$(a_1x_1 + a_2x_2 + \ldots + a_mx_m + a_0)y_j \geq 1$$

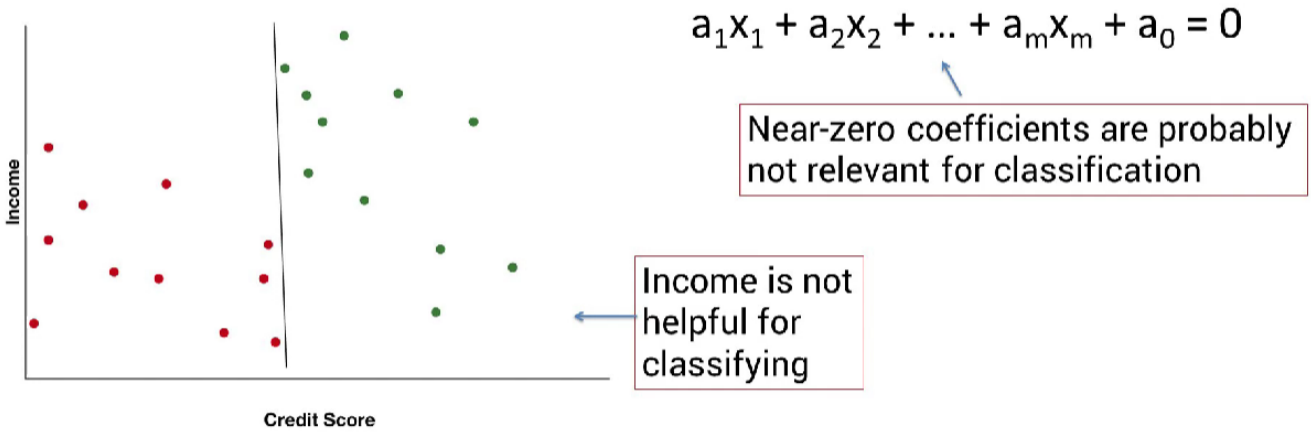
for each data point i

Soft Margin

Which trades off reducing errors and enlarging the margin

$$\underset{a_0,\ldots,a_m}{\text{Minimize}} \sum_{j=1}^{n} \max\left\{0, 1 - \left(\sum_{i=1}^{m} a_i x_{ij} + a_0\right) y_j\right\} + \lambda \sum_{i=1}^{m} (a_i)^2$$

Classification:
Support Vector Machines



Additional Notes:

- SVM can be non-linear with the use of Kernel methods.
- other methods like Logistic Regression can give probability answers

Look at the classification error expression below. For which set of data points (1-20 or 21-50) is it more important to avoid classification errors? $\sum_{j=1}^{20} 5 \times max\{0, 1 - (\sum_{i=1}^{m} a_i x_{ij} + a_0)\, y_j\}$ $+ \sum_{j=21}^{50} 200 \times max\{0, 1 - (\sum_{i=1}^{m} a_i x_{ij} + a_0)\, y_j\}$

- ○ 1-20
- ◉ 21-50 ✔

Answer
Correct:
The multiplier for classification errors is 200 for data points 21-50, much more than 5 for data points 1-20

Lesson 2.7 (C): Scaling and Standardization

Adjusting the data - Scaling

Common scaling: data between 0 and 1

Scale factor by factor

- Let $x_{\min j} \stackrel{\text{def}}{=} \min_i x_{ij}$
- Let $x_{max\ j} \stackrel{\text{def}}{=} \max_i x_{ij}$
- For each data point i:
  - $x_{ij}^{scaled} = \frac{x_{ij} - x_{\min j}}{x_{max\ j} - x_{\min j}}$

General scaling between b and a:

- $x_{ij}^{scaled\ [b,a]} = x_{ij}^{scaled\ [0,1]}(a - b) + b$

Adjusting the data - Standardizing

- Scaling to a normal distribution
  - Common scaling: mean = 0, standard deviation = 1
  - Factor j has mean $\mu_j = \frac{\sum_{i=1}^{n} x_{ij}}{n}$
  - Factor j has standard deviation σ_j
  - For each data point i:
    - $x_{ij}^{standardized} = \frac{x_{ij} - \mu_j}{\sigma_j}$

Lesson 2.8 (M): K-Nearest Neighbor Algorithm

Solving Classification Problems
k-Nearest Neighbor algorithm

Keep in mind:

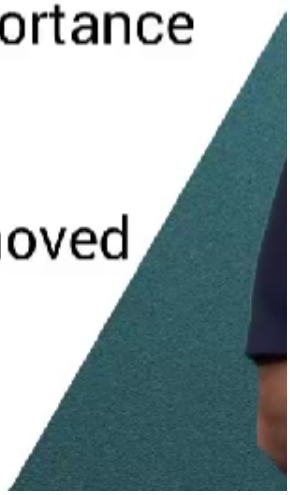
- Can use other distance metrics

  (straight-line distance is $\sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$)
- Attributes can be weighted by importance

  $\sqrt{\sum_{i=1}^{n} w_i |x_i - y_i|^2}$
- Unimportant attributes can be removed

  ($w_i = 0$ for unimportant attributes)
- Choose a good value of k

  (see validation lesson)



Module 3: Validation (C)

Lesson 3.1 (C): Introduction to Validation

Data has two types of patterns

- Real Effect - real relationship between attributes and responses
- Random Effect - random, but looks like a real effect

Fitting matches both real and random effects

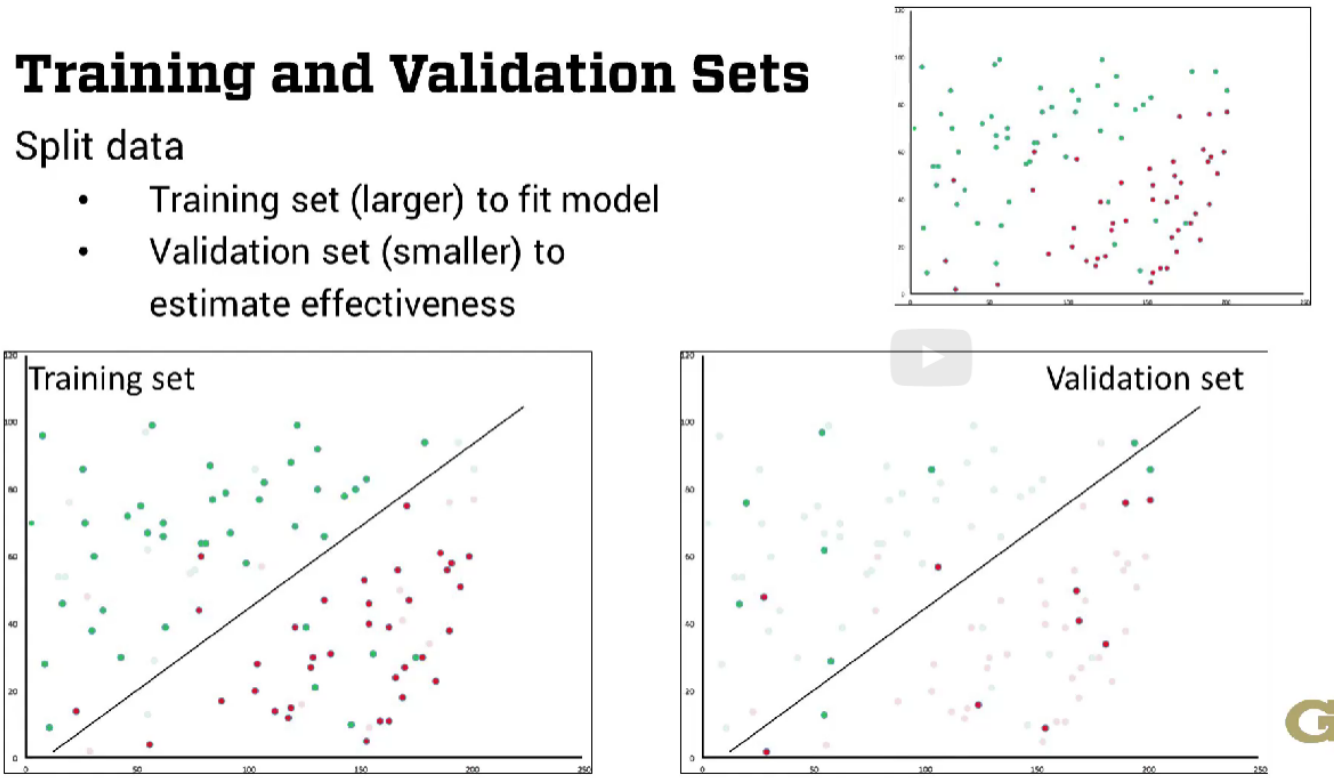
- Real Effects: same in all data sets
- Random Effects: different in all data sets

If we use the same data to fit a model as we do to estimate how good it is, what is likely to happen?

The model will appear to be better than it really is.

The model will be fit to both real and random patterns in the training data. The model's effectiveness on this training data set will include both types of patterns, but its true effectiveness on other data sets (with different random patterns) will only include the real patterns

Lesson 3.2 (C): Validation and Test Data Sets



The percentage performance on the validation data is a more accurate measure of the model's effectiveness

Training and Validation Sets

Choosing the best model?

- Example: 5 SVM models and 5 k-nearest-neighbor models

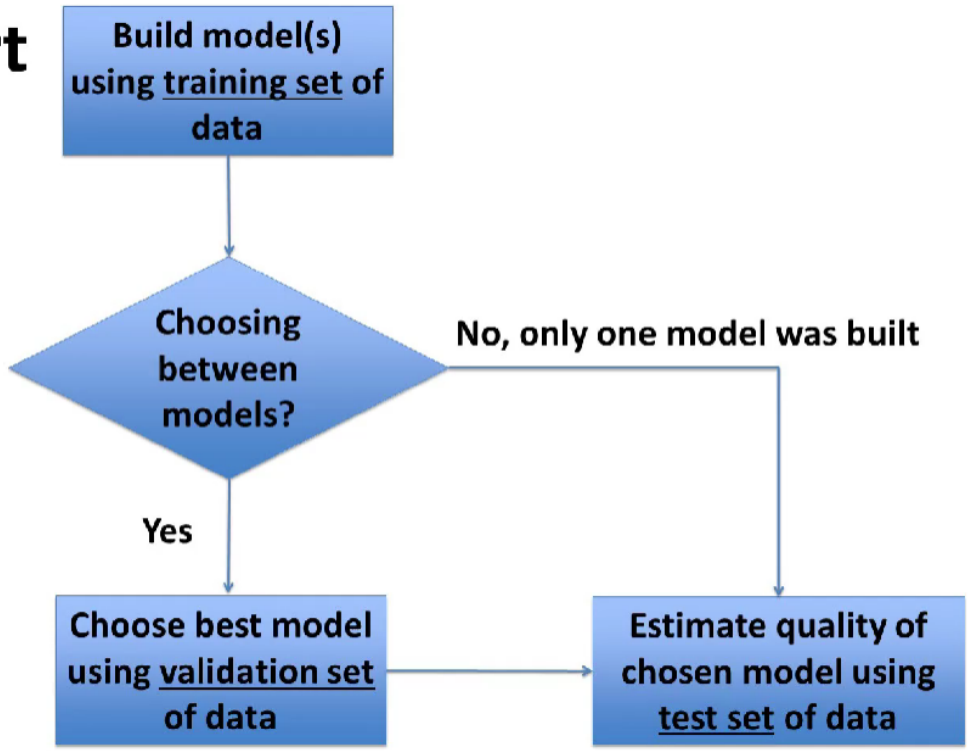
Model	1	2	3	4	5
SVM	93/100	88/100	96/100	97/100	95/100
KNN	94/100	94/100	90/100	95/100	82/100

Problem:

- Observed performance = real quality + random effects
  - High-performing models more likely to have above-average random effects

So observed performance of chosen model is probably too optimistic

Flowchart



Training Set - Building Models

Validation Set - Picking a Model

Test Set - Estimating a performance of chosen model

Further Reading on Cross-Validation (http://www.di.ens.fr/willow/pdfs/2010_Arlot_Celisse_SS.pdf (http://www.di.ens.fr/willow/pdfs/2010_Arlot_Celisse_SS.pdf))

Lesson 3.3 (C): Splitting Data

Splitting Data

- How much data goes into each set?
  - Working with one model (only training and test sets needed)
    - Rule of thumb
      - 70-90% training, 10-30% test
  - Comparing models (need training, validation, and test sets)
    - Rule of thumb
      - 50-70% training
      - split the rest equally between validation and test

Splitting Data

Example: 1000 data points: 60% training, 20% validation, 20% test

Method 1: Random

- Randomly choose 600 data points for training
- Randomly choose 200 (of the remaining 400) data points for validation
- The remaining 200 data points make up the test set

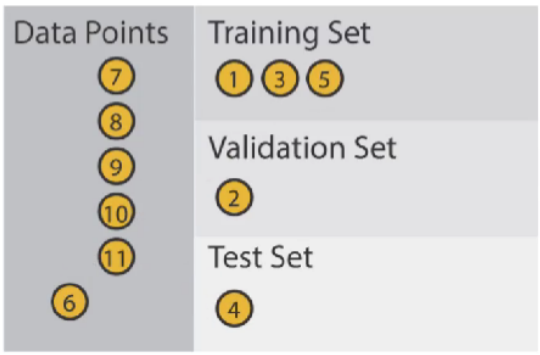
Method 2: Rotation

- Take turns selecting points

Example:

5 data point rotation sequence

Training–Validation–Training–Test–Training



Be careful about introducing bias

Example: daily sales data (Mon-Fri)

- Randomness could give one set more early or late data
  - Rotation equally separates data
- Rotation may introduce bias
  - Example: 5-data-point rotation means all Mondays are in one set, all Tuesdays are in one set, etc.

Consider combined approach?

- Example: 60% of Monday data for training, 60% of Tuesday data for training, etc.

Lesson 3.4 (C): Cross-Validation

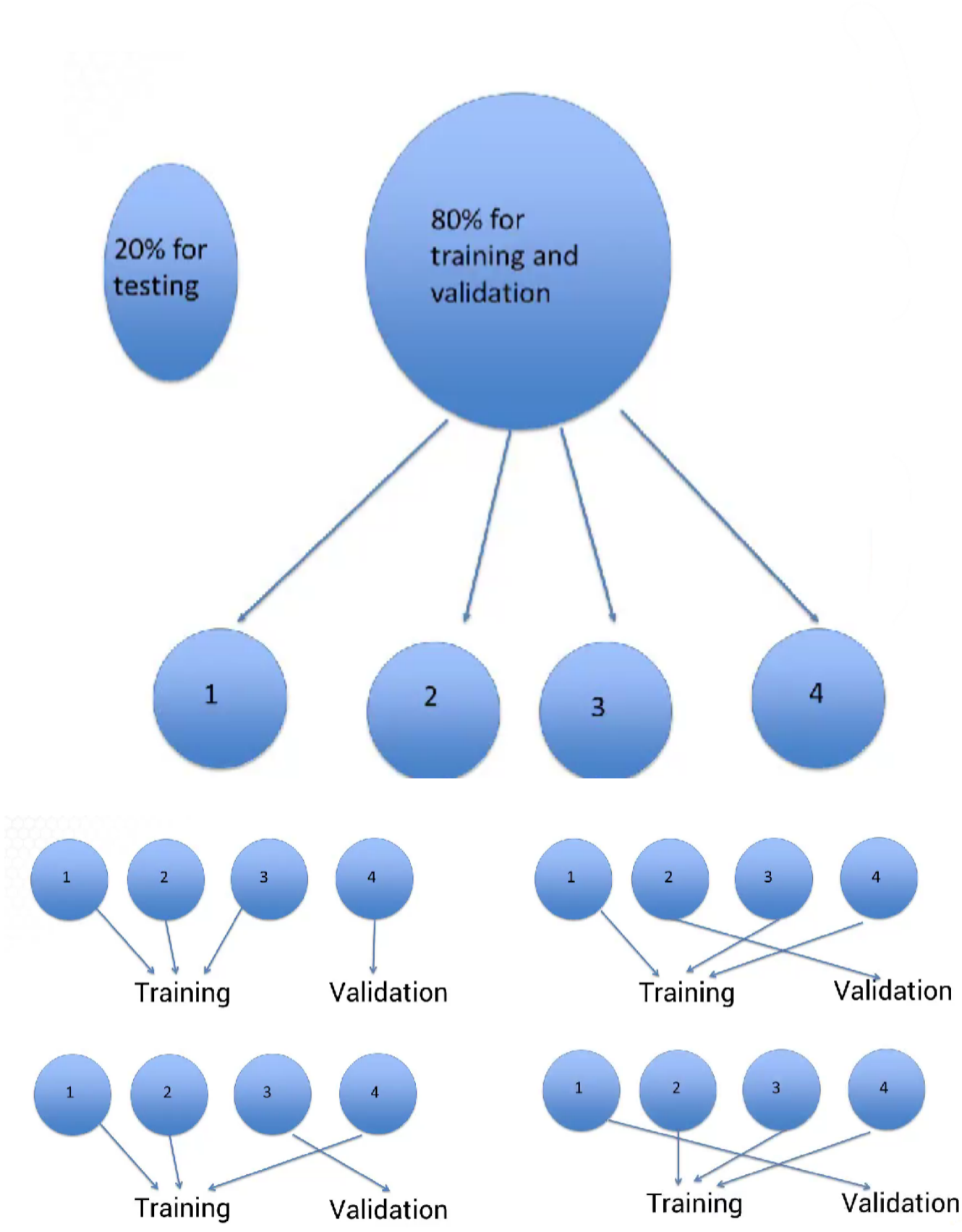
Cross Validation

Question:

- What if important data only appears in the validation or test sets?

Solution:

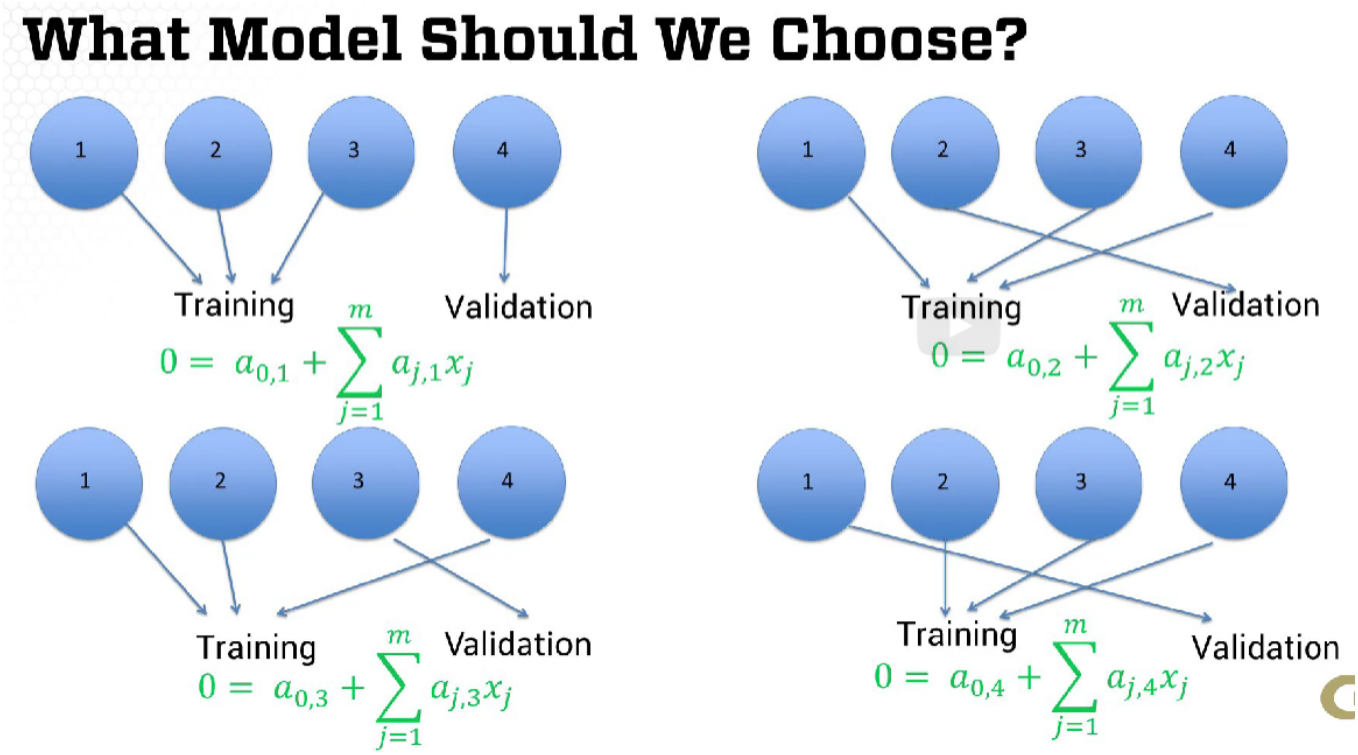
- Use cross-validation!



For each of the K parts:

- Train the model on all the other parts
- Evaluate it on the one remaining part

Average the K evaluations to estimate the model's quality



Answer: None

- do not average the coefficients across the four splits
- train the model again using all the data

Module 4: Clustering (M & C)

Lesson 4.1 (M): Introduction to Clustering

Cluster data into groups based on similar characteristics or by Euclidean Distance

Lesson 4.2 (C): Distance Norms

Distance Norms

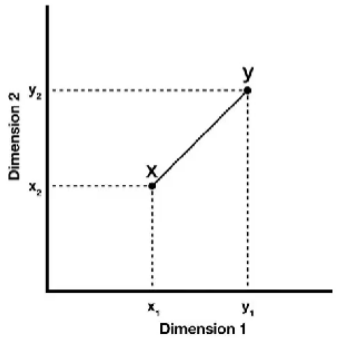
Euclidean (straight-line) distance

$$\text{distance} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

Rectilinear distance

$$\text{distance} = |x_1 - y_1| + |x_2 - y_2|$$

$$\text{Distance} = \sqrt[p]{|x_1 - y_1|^p + |x_2 - y_2|^p}$$



Distance Norms

Euclidean (straight-line) distance

$$\text{distance} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

Rectilinear distance (1-norm)

$$\text{distance} = |x_1 - y_1| + |x_2 - y_2|$$

p-norm distance
(Minkowski distance)

$$\text{Distance} = \sqrt[p]{|x_1 - y_1|^p + |x_2 - y_2|^p}$$

$$\text{Distance} = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}$$

Rectilinear Distance (1-norm) is also known as the Manhattan distance

P-Norm Distance

(Minkowski distance)

$$\text{Distance} = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}$$

∞-norm = largest (absolute) of a set of numbers

∞-norm distance??

$$\text{Distance} = \sqrt[\infty]{\sum_{i=1}^{n} |x_i - y_i|^\infty} = \sqrt[\infty]{\max_i |x_i - y_i|^\infty} = \max_i |x_i - y_i|$$

Sum equals the largest $|x_i - y_i|$ to the infinty power

$$|x_1 - y_1|^\infty + |x_2 - y_2|^\infty + \ldots + |x_n - y_n|^\infty$$

There is a very good example given in the 4.2 Lecture about the warehouse retrieval system and how in

relates to the infinity norm.

Straight-line distance $\sqrt[2]{\sum_{i=1}^{n}(x_i - y_i)^2}$ corresponds to which distance metric?

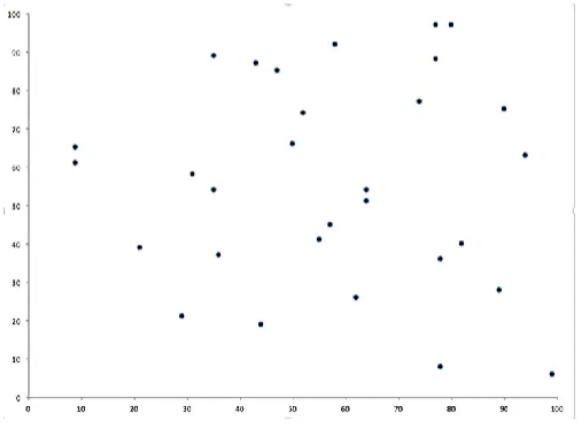
- ○ 1-norm
- ◉ 2-norm
- ○ ∞ norm

✔

Answer
Correct: The power and root are the same as the norm.

Lesson 4.3 (M): K-means Clustering

Clustering
Grouping data points
Solve using k-means algorithm



x_{ij} = attribute j of data point i

$$y_{ik} = \begin{cases} 1, & \text{if data point i is in cluster k} \\ 0, & \text{if not} \end{cases}$$

z_{jk} = coordinate j of cluster center k

$$\underset{y,z}{\text{Minimize}} \sum_i \sum_k y_{ik} \sqrt{\sum_j (x_{ij} - z_{jk})^2}$$

Subject to $\sum_k y_{ik} = 1$ for each i

The Second Last equation calculates the root sum of squared errors only for the points that belong to the particular cluster.

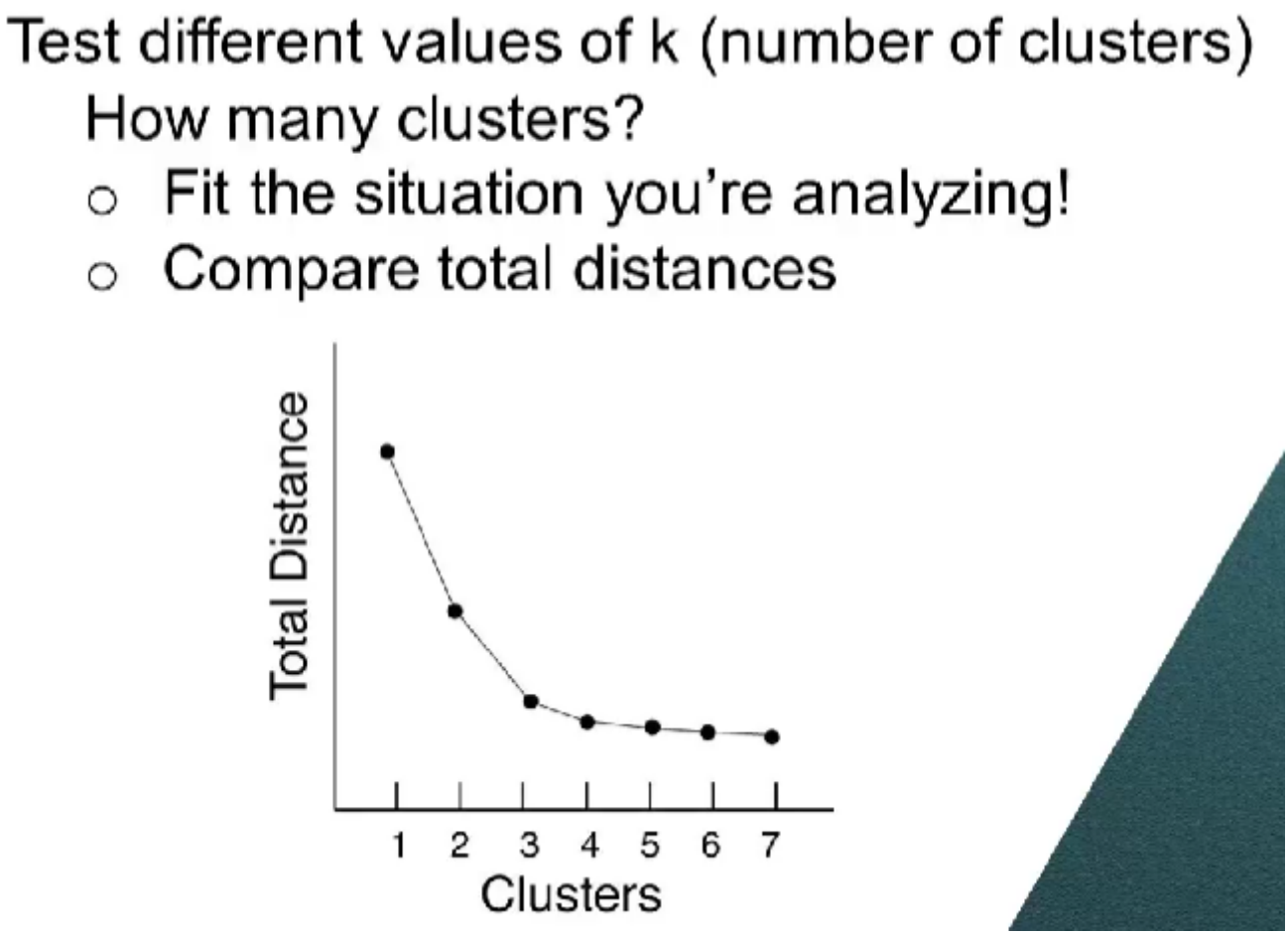
The last equation at the bottom right is just saying that each data point i can only belong to EXACTLY ONE cluster k.

k-means animation (http://shabal.in/visuals/kmeans/4.html (http://shabal.in/visuals/kmeans/4.html))

- k-mean algorithm is an example of a **heuristic** because it is fast and good but not guaranteed to find the absolute best solution
- It is an example of an **Expectation-Maximisation (EM)** algorithm:

When we calculate the cluster centers, we're taking the mean of all the points in the cluster similar to finding an expectation. And when we reassign data points to cluster centers, that's the maximization step. Really we're minimizing finding the smallest distance to a cluster center. But we could think of it as **maximizing the negative of the distance** to a cluster center. So our algorithm takes turns between taking an expectation, maximizing, expectation, maximizing, over and over. So it's called an expectation-maximization or EM algorithm.

Lesson 4.4 (M): Practical Details for K-Means

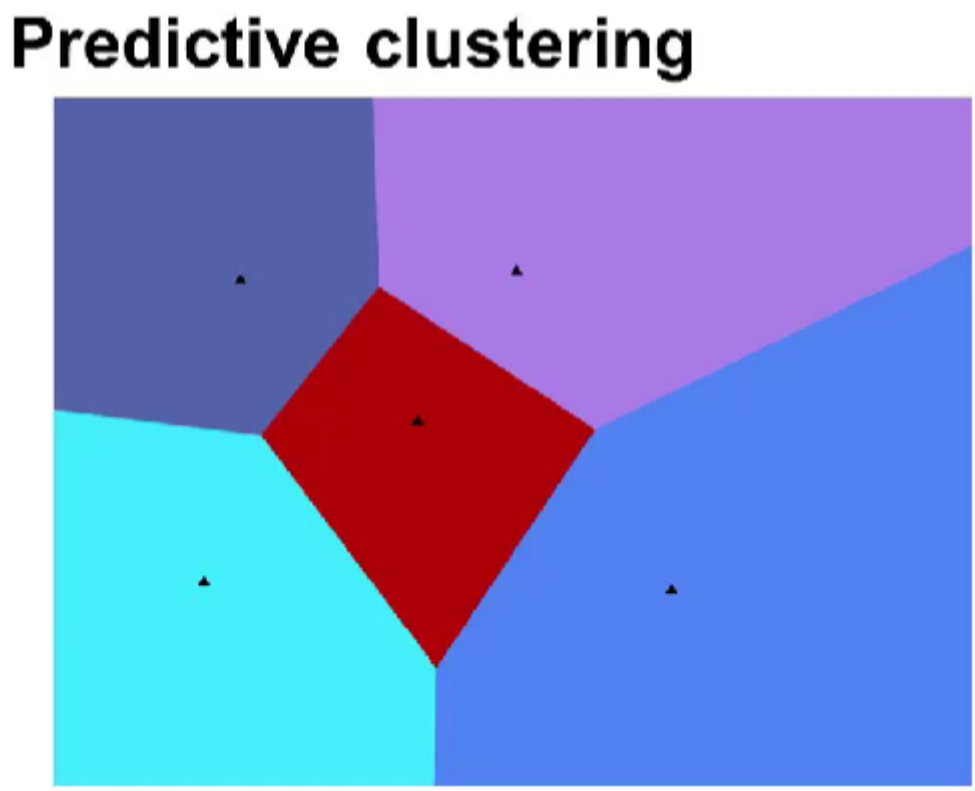


Suppose we find k-means clusterings for a bunch of different values of k, and for each one we calculate the total distance of each data point to its cluster center, we can plot that in two-dimensions.

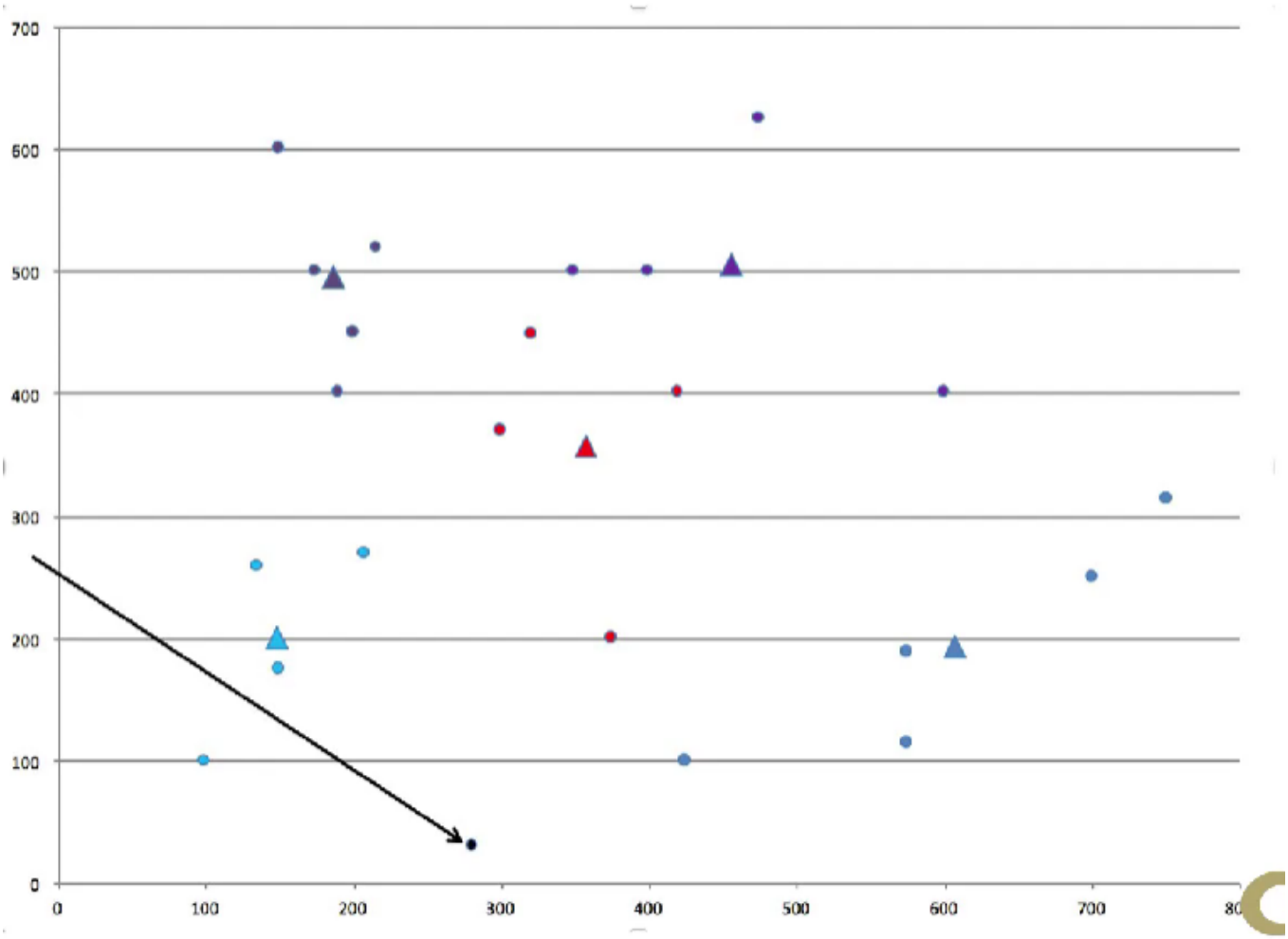
The horizontal axis is the number of clusters k, and the vertical axis is the total distance from points to cluster centers. Now we can look to see where the kink in the curve is. Here where the marginal benefit of adding another clusters starts to be small. This is an elbow plot.

Lesson 4.5 (M): Clustering for Prediction

Voronoi Diagram



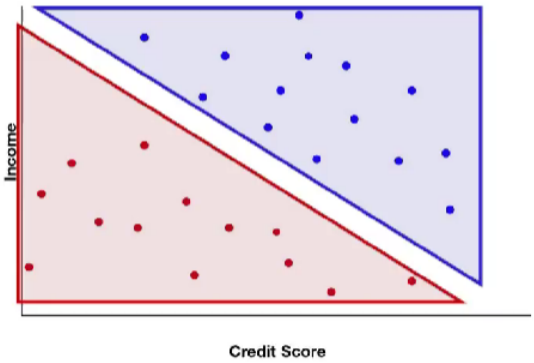
Predictive Clustering: If the new point isn't inside a cluster we can just choose whichever cluster center is closest and that's as reasonable a choice as any for predicting which cluster the new point is in.



Lesson 4.6 (M): Clustering vs Classification

Classification

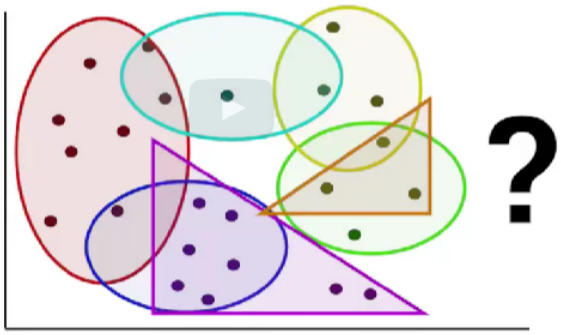
Grouping data points



Correct classification of data points is already known

Clustering

Grouping data points



Correct classification of data points is **not** known

Supervised learning

Correct answer (response) is known

For each data point

Example: Classification

Unsupervised learning

Correct answer (response) is not known

Example: Clustering

Module 5: Basic Data Preparation (C)

Lesson 5.1 (C): Introduction to Data Preparation

Recall specific data used for different analyses:

- Predictors (regression)
- Factors (classification)

Scale the Data

- standardisation
- normalisation

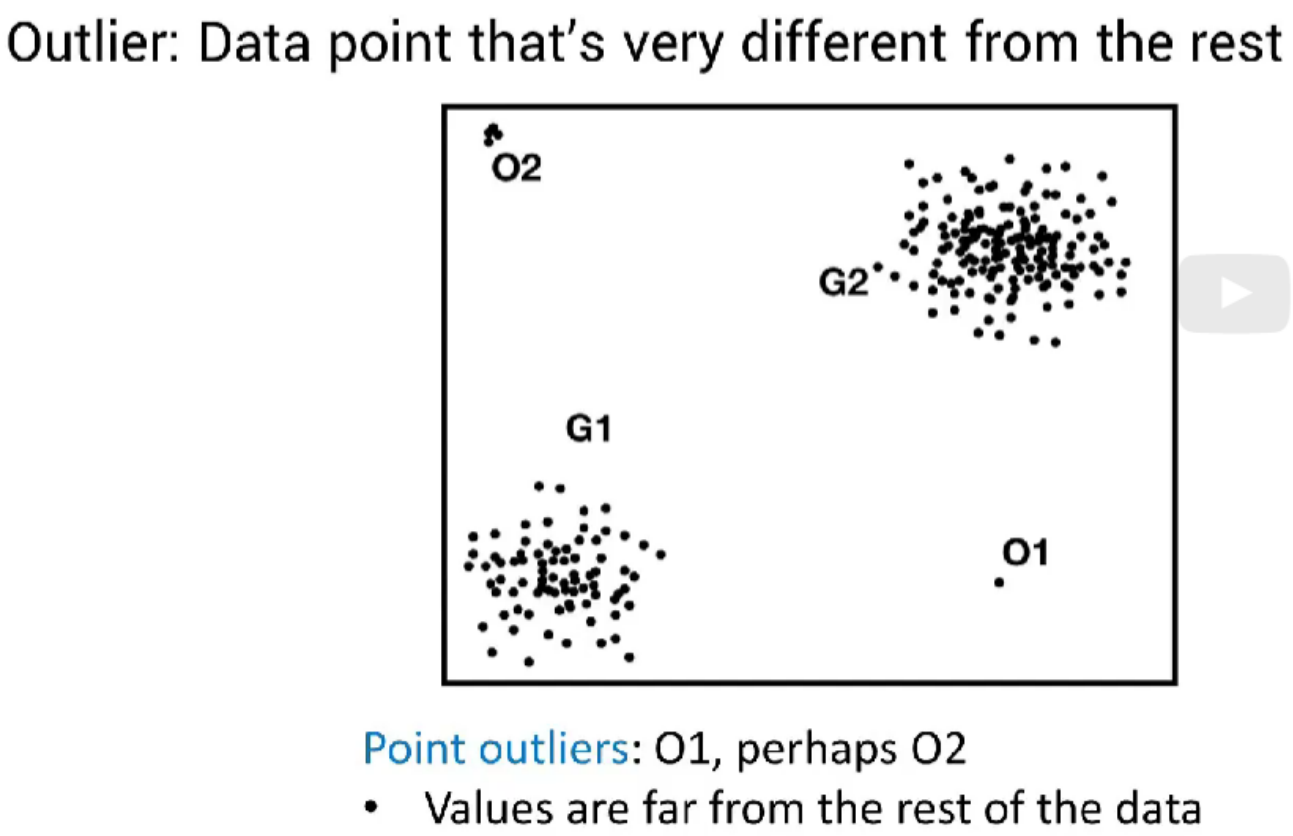
Extraneous Information

- complicates the model
- harder to correctly interpret the solution

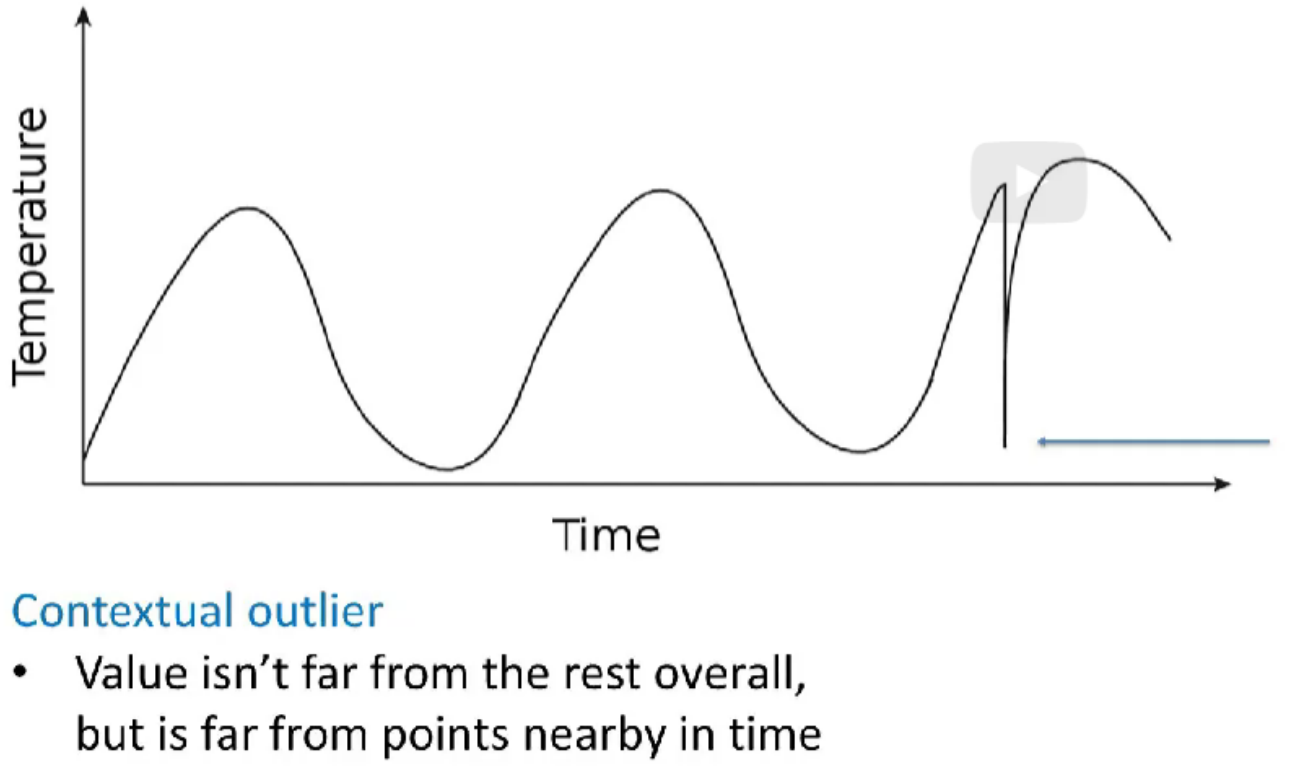
Lesson 5.2 (C): Outlier Detection

An outlier is a data point that's very different from the rest of the dataset, the most obvious form of outliers where the value of a data point is very different from the rest of the data.

Point Outlier

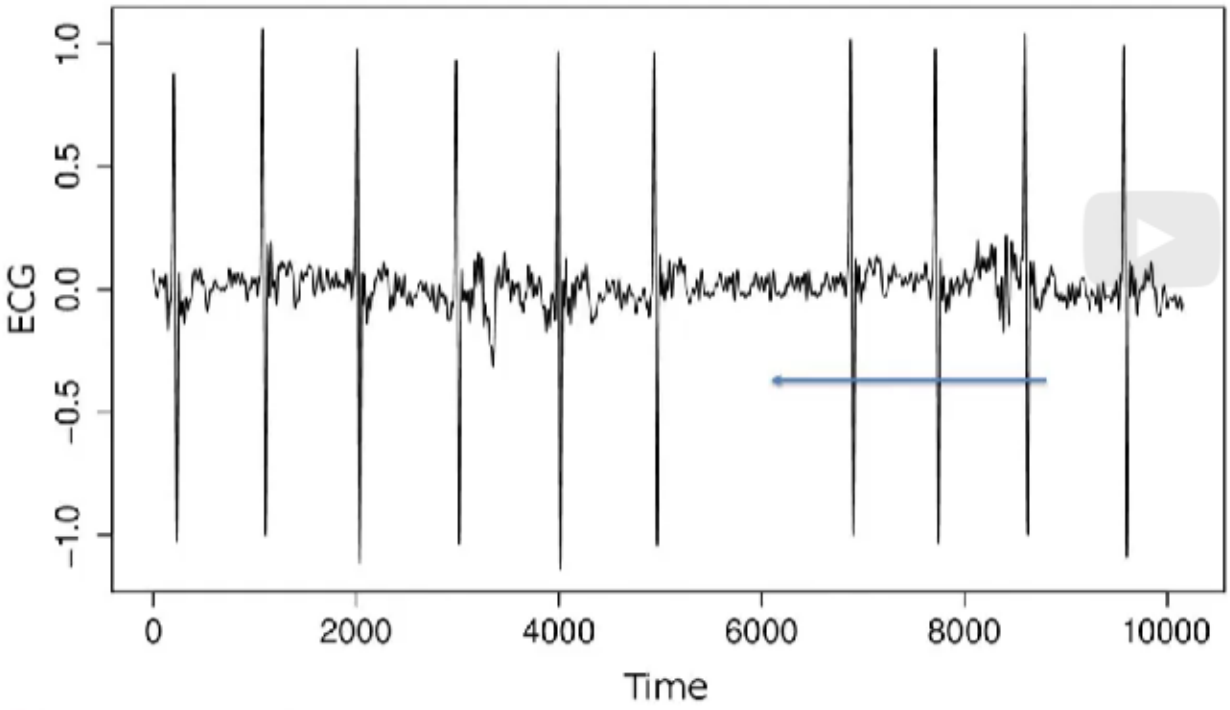


Contextual Outlier



Here's a picture of an outlier in time-series data. It has just one point that's far from the rest of the curve, the temperature value at this point isn't itself an outlier, but the **time at which it occurs** makes it an outlier compared to the rest of the data. This type of outlier is sometimes called the contextual outlier because it relies on the context provided by the other points.

Collective Outlier (Outlier by Omission)



Collective outlier

- Something is missing in a range of points, but can't tell *exactly* where

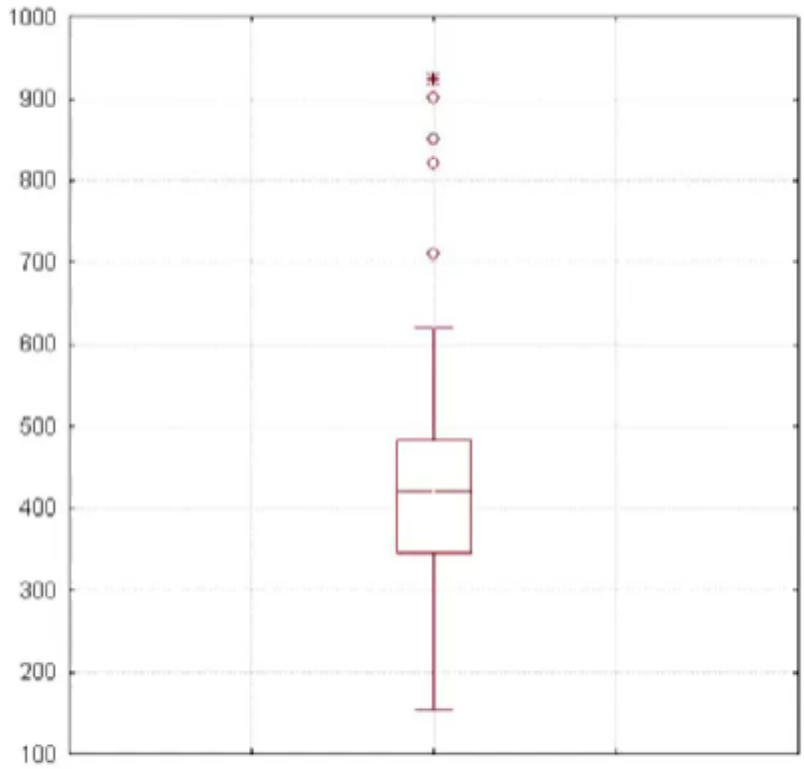
In this heartbeat data, it looks like there should be a large beat around times 6,000, but there isn't.

It's hard to tell exactly which data points should be called wrong, but sometime in there between time 5,000 milliseconds and time 7,000 milliseconds, it seems like there's something missing.

This type of outlier is sometimes called the collective outlier because the data points collectively seem to be an outlier, we could also think of this in terms of the time between beats.

Finding Outliers

- Box-and-whisker plot



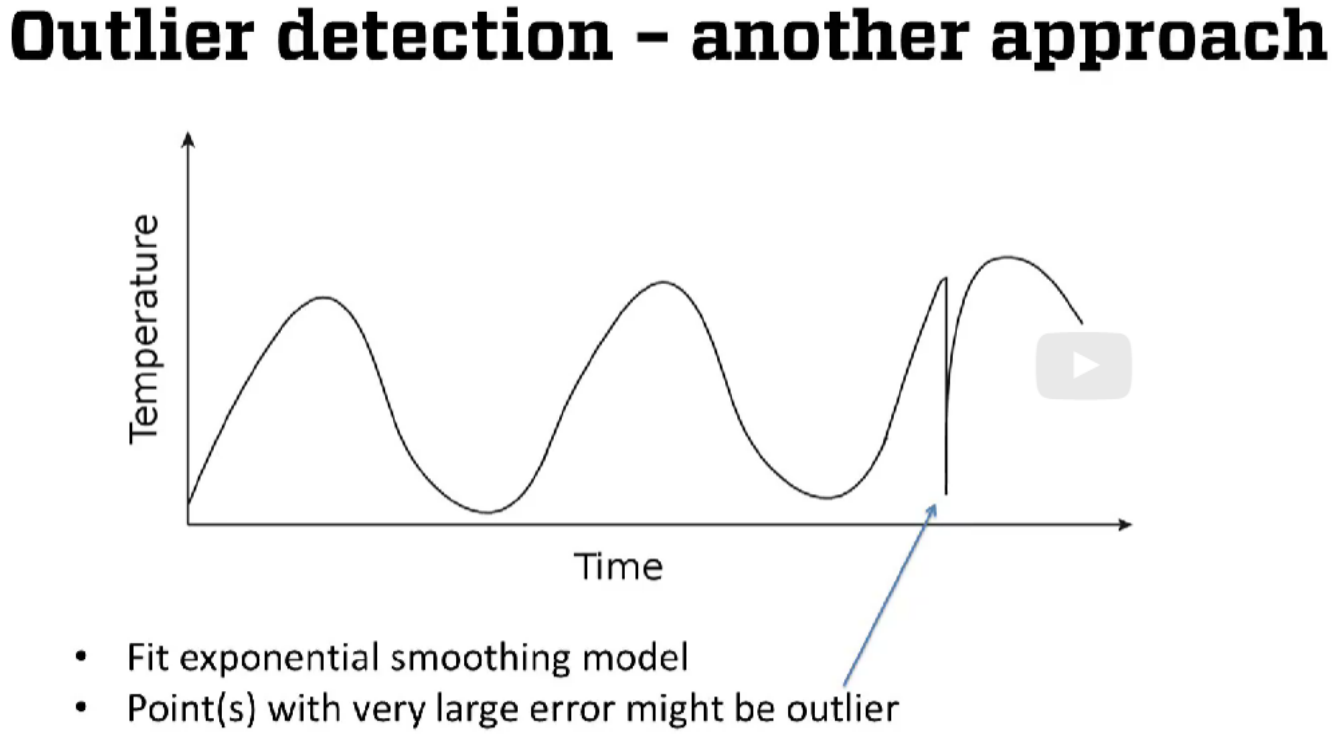
- Other automated methods also exist

To find point outliers in just one dimension, we could use a box and whisker plot.

The top and bottom of the box, or the 25th and 75th percentiles of the values and the horizontal line through the middle of the box is the median, the 50th percentile.

The vertical lines up and down from the box are called the whiskers and they stretch up and down to what you might think is a reasonable range of values, for example, we might pick the 10th, 90th percentiles are the fifth and 95th.

Beyond that, we plot a point for each value that's outside the reasonable range, and those points are possible outliers.



Unfortunately, there's not a good all-purpose way of detecting multi-dimensional outliers or other types of outliers that we could use upfront.

But one thing we could do is to build a model, fit the parameters, and then see which points have a lot of error, for example, suppose we fit an exponential smoothing model to this data, it's a nice smooth function and in each time period, the errors between the actual value on the model's estimate will be small, except here.

At this point, the model's error will be very large, the model will expect a point that's right up on the smooth curve and the actual value is far from it.

Lesson 5.3 (C): Dealing with Outliers

Outliers could be bad data

- sensor fails
- contaminated experiments
- wrong data input

Outliers could also be real data that cannot be removed. They could be data that happen occasionally which still have to be considered.

Need to investigate

- where the data came from
- how it was compiled
- unique situations

Dealing with Outliers

Bad data

- Omit data points
- Use imputation

Real/correct data

- Outliers expected in large data sets
- Example (normally-distributed)
  - 4% of data outside two standard deviations
  - With 1,000,000 data points, >2000 expected outside three standard deviations

- Removing real data outliers can be too optimistic

- Example
  - Time to transport perishable medicine from US to Africa
  - Outlying data points - weather events or political issues
    - These events can and do occur

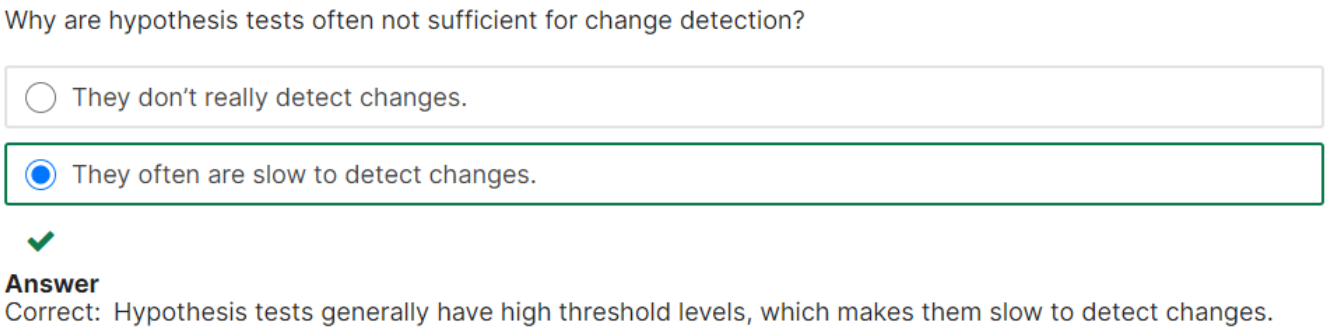
- Logistic regression model
  - Estimate probability of outliers happening under different conditions

- Second model
  - Estimate length of delivery under normal conditions
  - Use data without outliers

Module 6: Change Detection (M)

Lesson 6.1 (M): Introduction to Change Detection

Just an introduction to why change detection is used for time series data. Nothing technical here.



Further Reading on CUSUM: https://support.sas.com/documentation/onlinedoc/qc/132/cusum.pdf (https://support.sas.com/documentation/onlinedoc/qc/132/cusum.pdf)

[Start on p.577; before that, it's mostly specifics about how to use SAS software.]

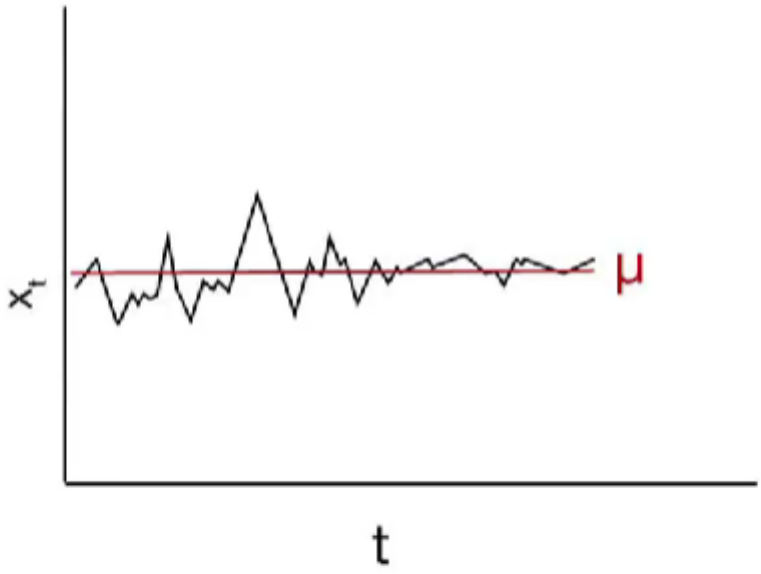
Lesson 6.2 (M): CUSUM for Change Detection

The name CUSUM is short for a cumulative sum and it answers the question, has the mean of the observed distribution gone beyond a critical level?

CUSUM can detect when a process gets to a higher level than before, or to a lower level than before, or both.

x_t = observed value at time t

μ = mean of x, if no change



$$S_t = \max\{0, S_{t-1} + (x_t - \mu - C)\}$$

Is $S_t \geq T$?

Sometimes in fact, maybe about half the time, X_t will be higher than the expectation just at random. So we include a value C to pull the running total down a little bit.

The bigger C is the harder it is for S_t to get large and the less sensitive the method will be.

And the smaller C gets, the more sensitive the method is because S_t can get larger, faster.

STOPPED AT 3:41 FOR WEEK 3 LESSON 6.2

Type *Markdown* and LaTeX: α^2

Type *Markdown* and LaTeX: α^2

Type *Markdown* and LaTeX: α^2

Type *Markdown* and LaTeX: α^2

Type *Markdown* and LaTeX: α^2

Type *Markdown* and LaTeX: α^2

Type *Markdown* and LaTeX: α^2

Type *Markdown* and LaTeX: α^2