ISYE6501x Mid Term Quiz 1 Revision

Course Structure

Knowledge Building

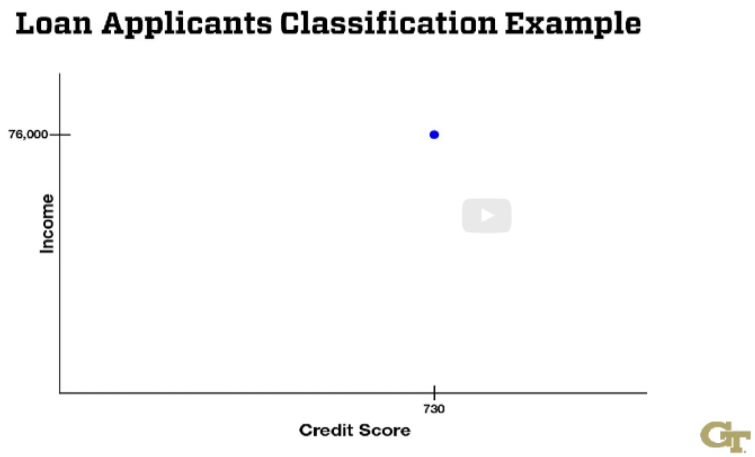
Module 1: Introduction
Module 2: Classification
Module 3: Validation
Module 4: Clustering
Module 5: Basic Data Preparation
Module 6: Change Detection
Module 7: Exponential Smoothing
Module 8: Basic Regression
Module 9: Advanced Data Preparation
Module 10: Advanced Regression
Module 11: Variable Selection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

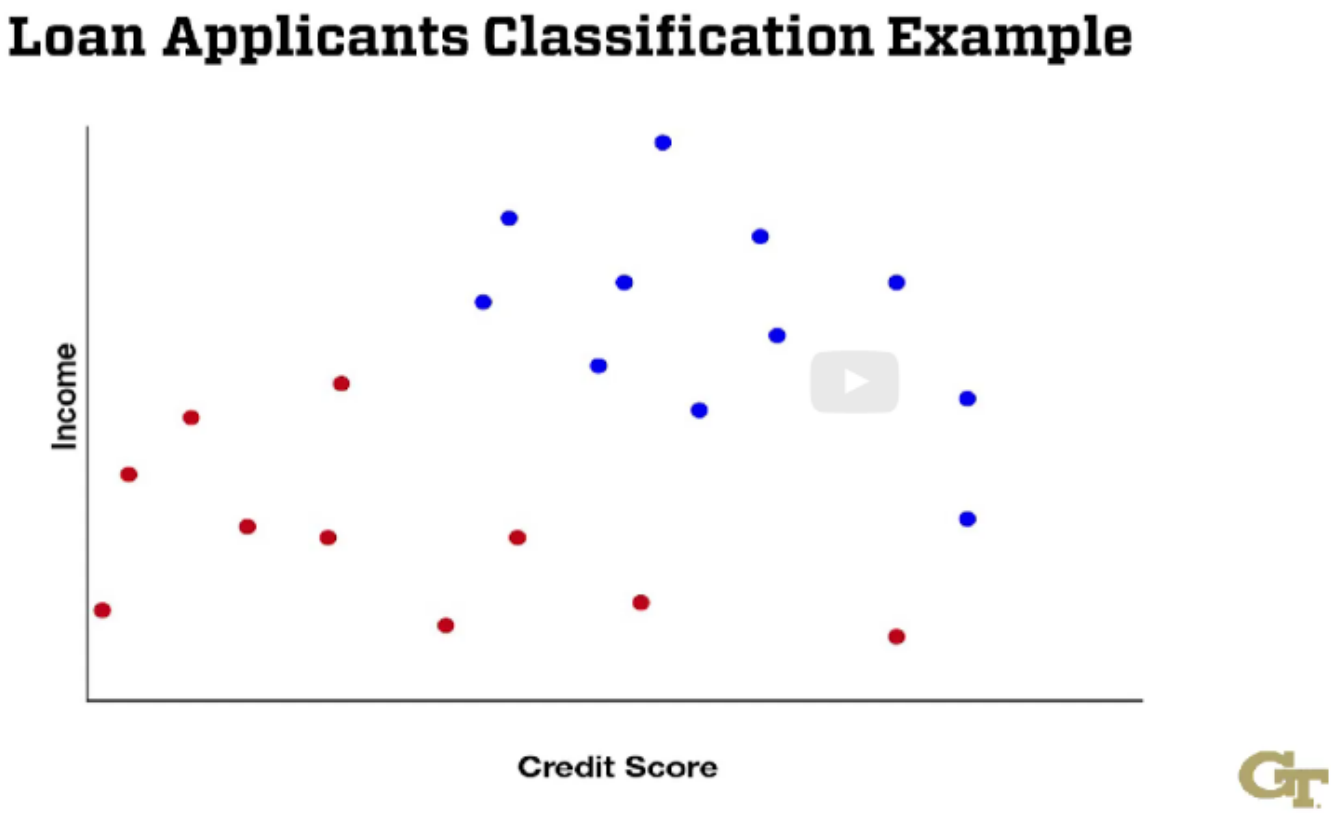
Module 1: Introduction

Nothing much here. Just course introductions.

Module 2: Classification

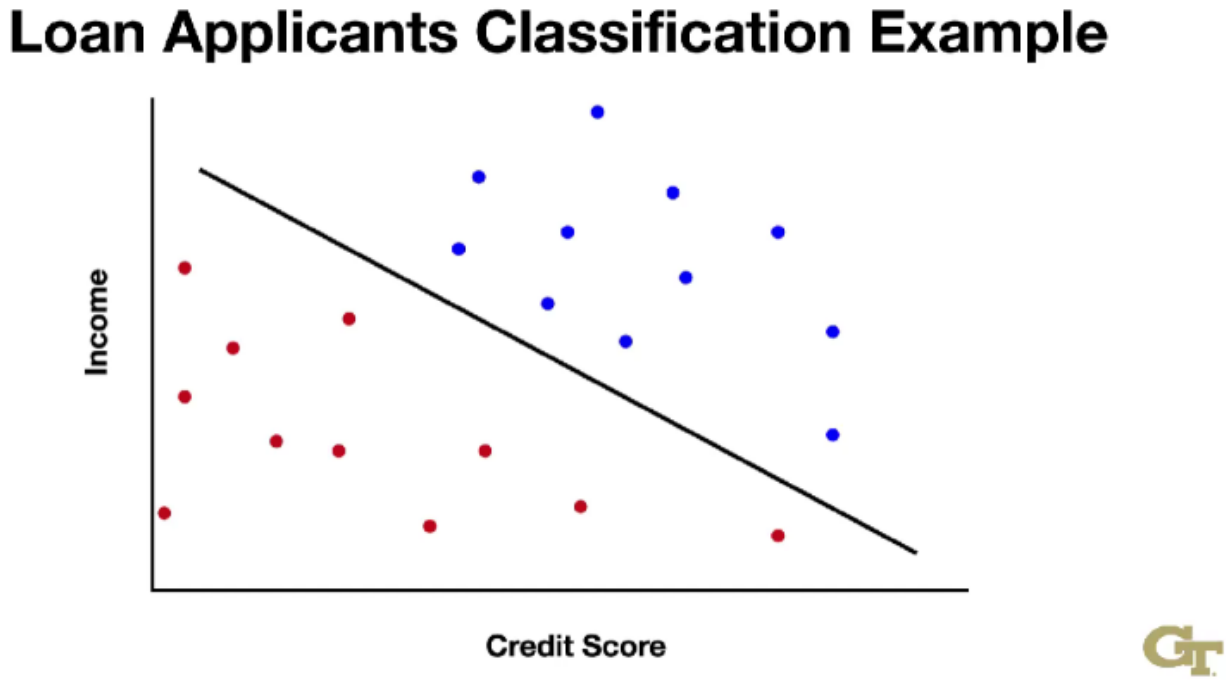
Lesson 2.1: Introduction to Classification





Blue - Loan Repaid
Red - Defaulted

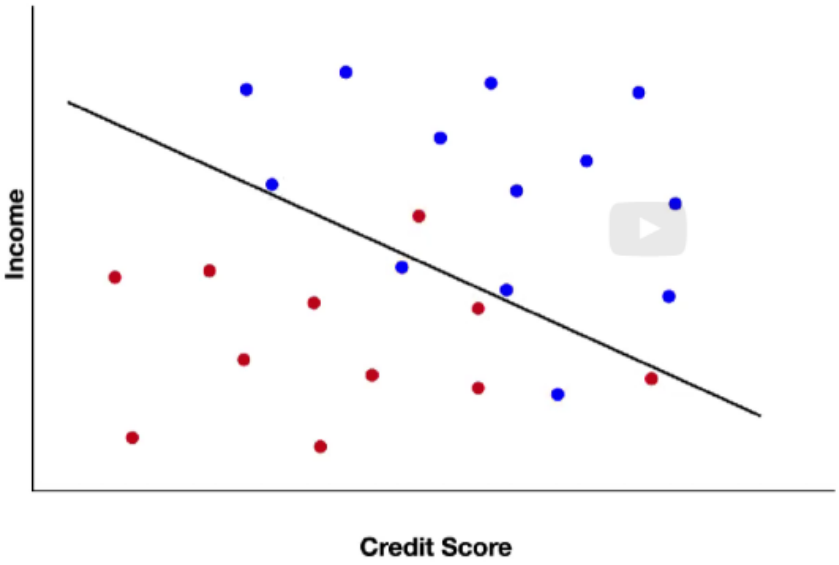
Lesson 2.2 (M): Choosing a Classifier



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

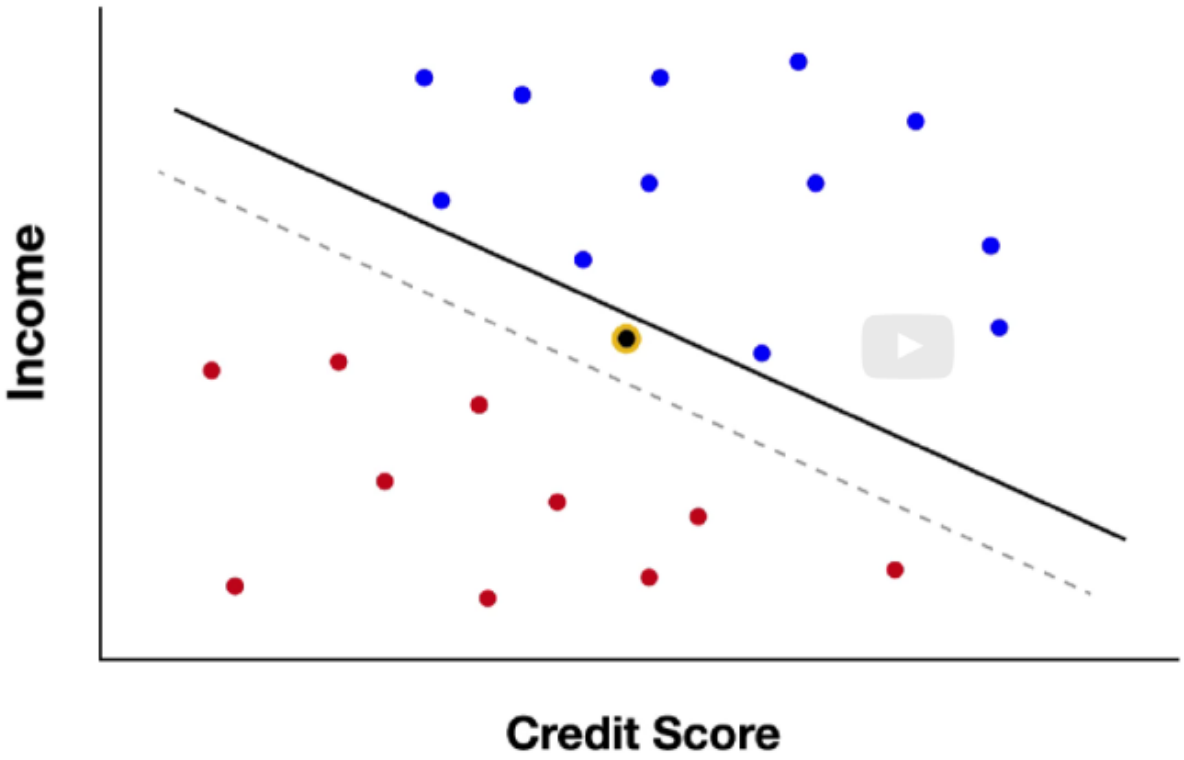
A Soft Classifier is used when you cannot perfectly separate the points.

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.

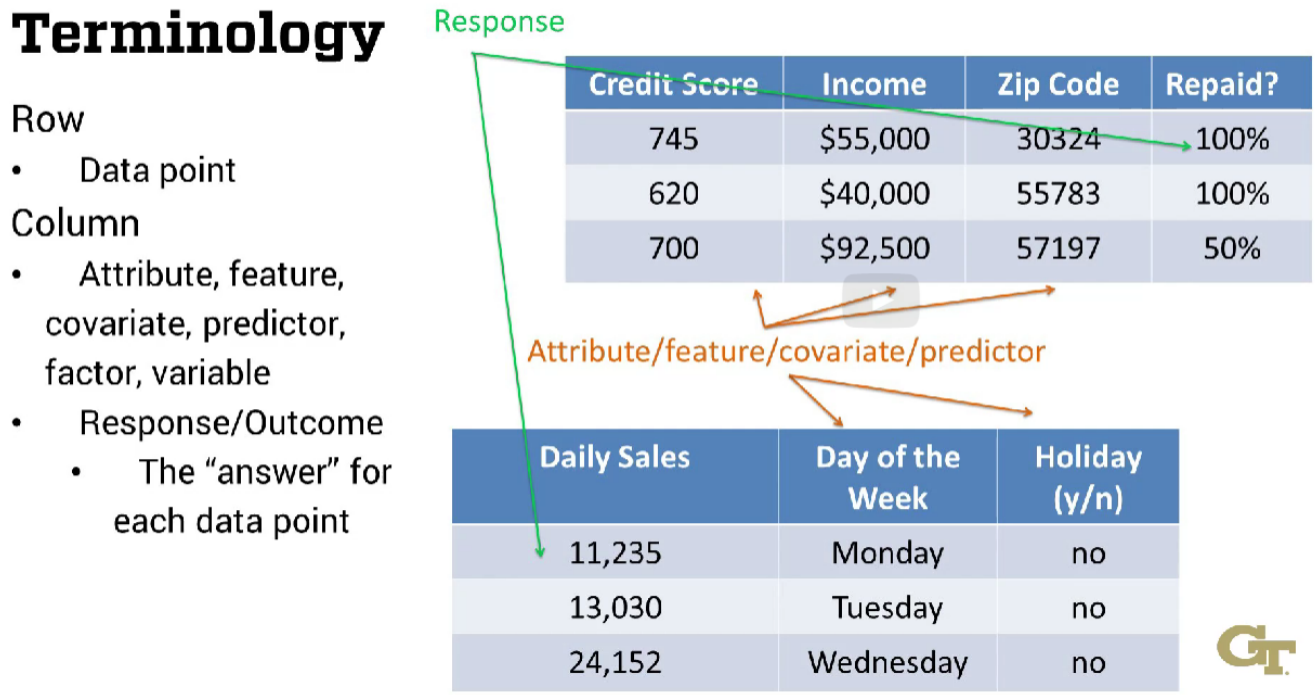
Given that realistically it is impossible to separate with no mistakes, we might be more willing to accept one type of mistake than another.

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

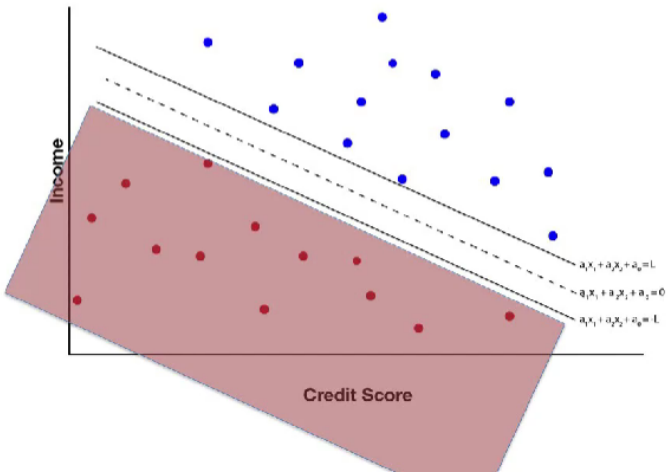
Extra reading on SVM Classifier (http://pyml.sourceforge.net/doc/howto.pdf (http://pyml.sourceforge.net/doc/howto.pdf))

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:

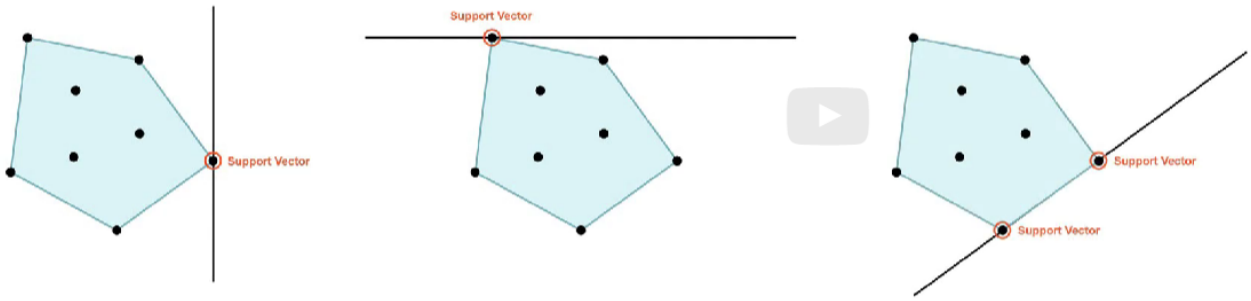
$$\underset{a_0,\ldots,a_n}{\text{Minimize}} \sum_{i=1}^{m} \max\left\{0,1-\left(\sum_{j=1}^{n} a_j x_{ij} + a_0\right) y_i\right\} + \lambda \sum_{j=1}^{n} \left(a_j\right)^2$$

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

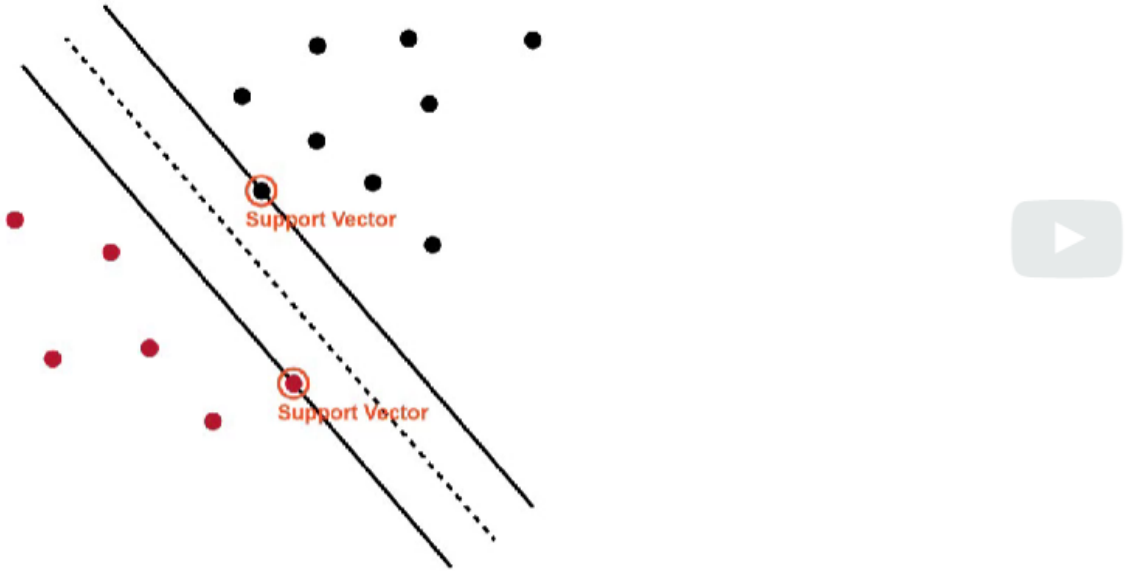
Question: In Lesson 2.4

- 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

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

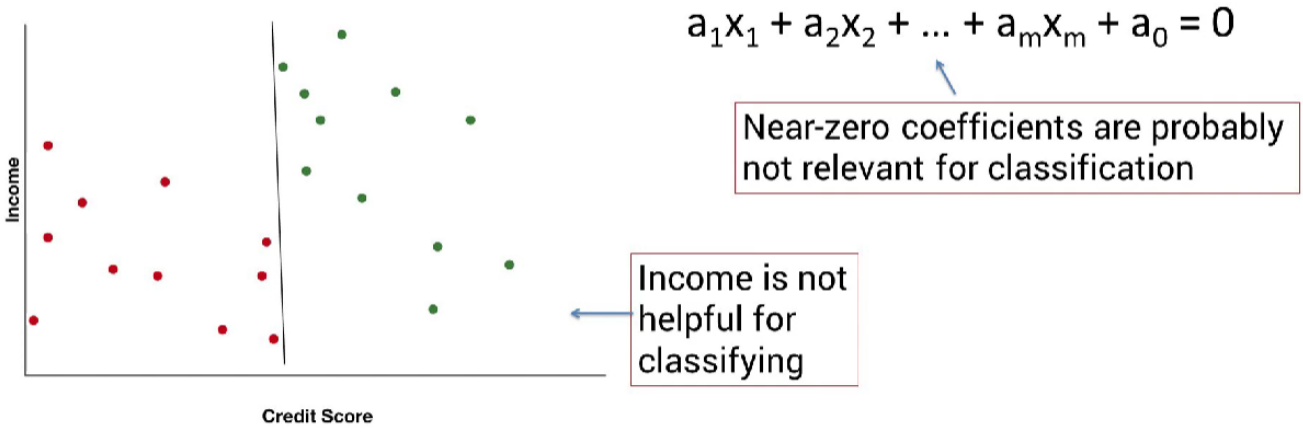
for each data point i

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

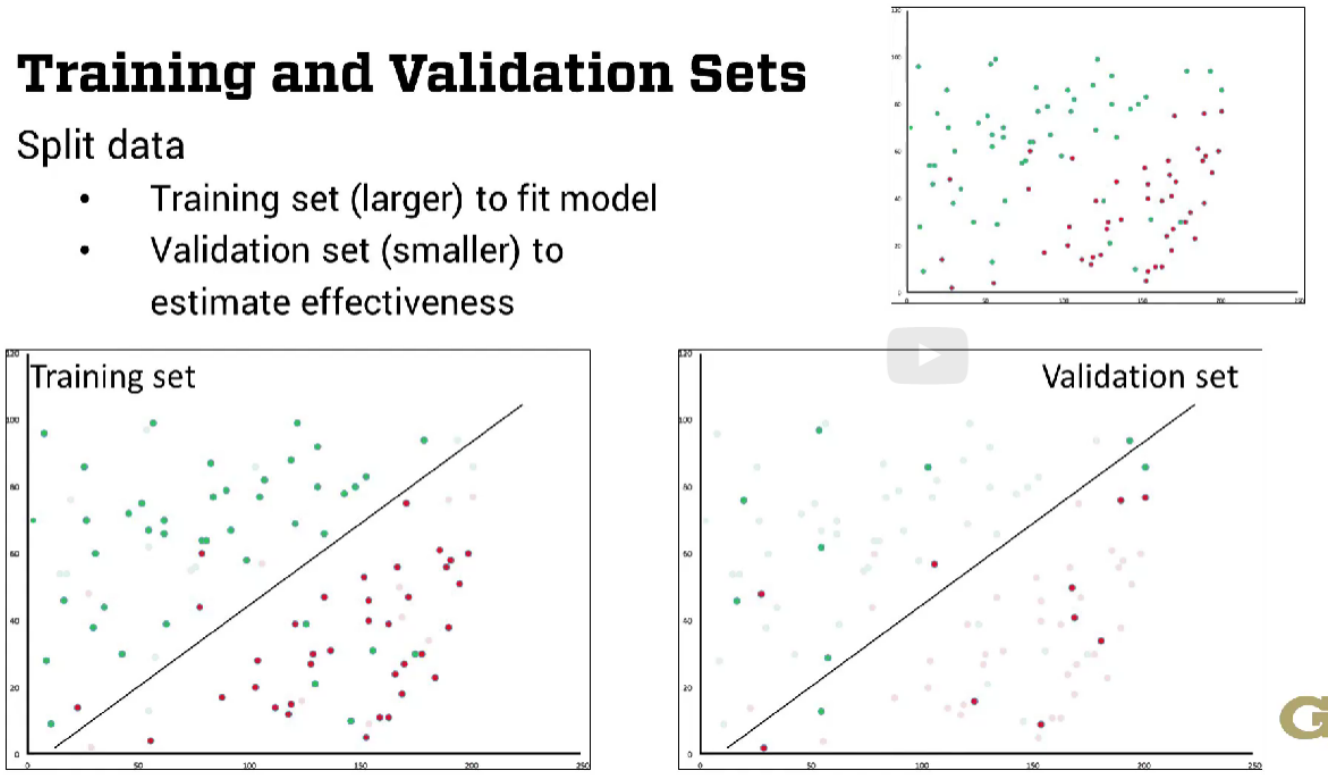
- Real Effects: same in all data sets
- Random Effects: different in all data sets

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

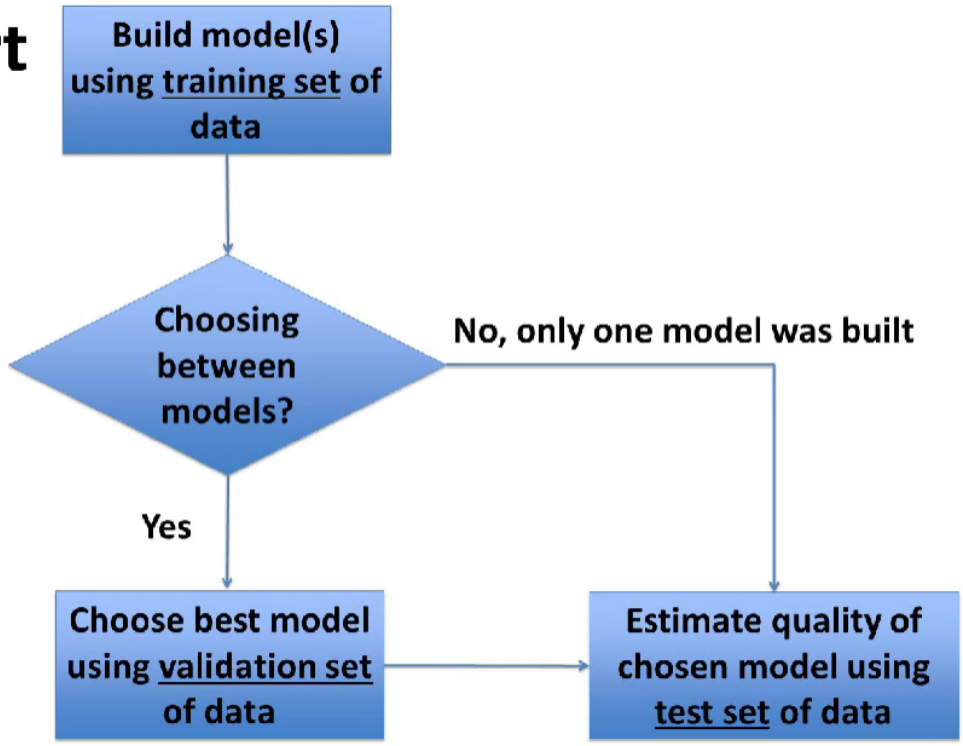
Model	1	2	3	4	5
SVM	93/100	88/100	96/100	97/100	95/100
KNN	94/100	94/100	90/100	95/100	82/100

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

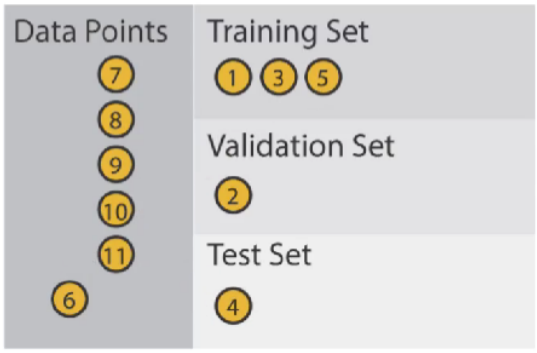
Method 2: Rotation

- 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

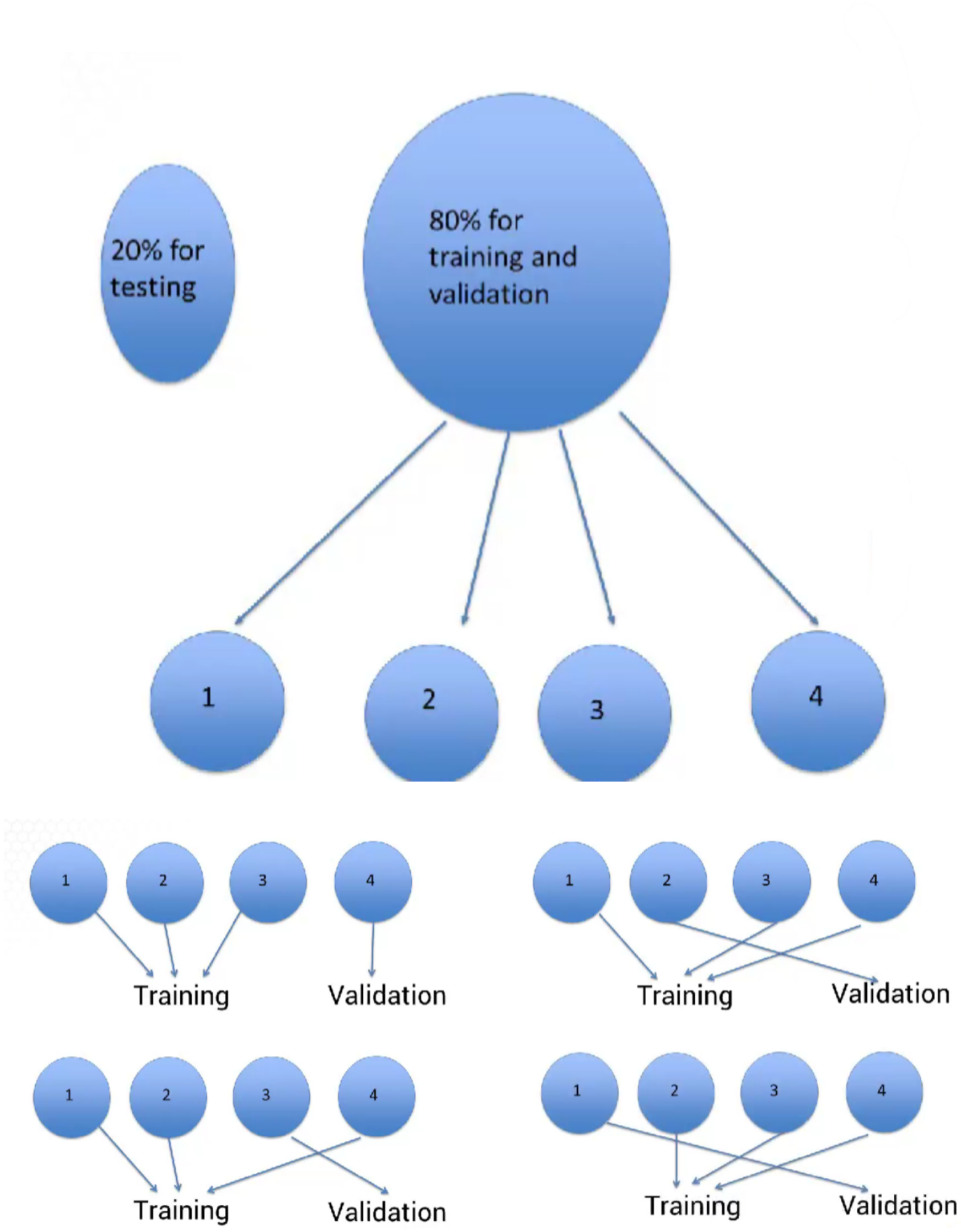
Cross Validation

Question:

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

Solution:

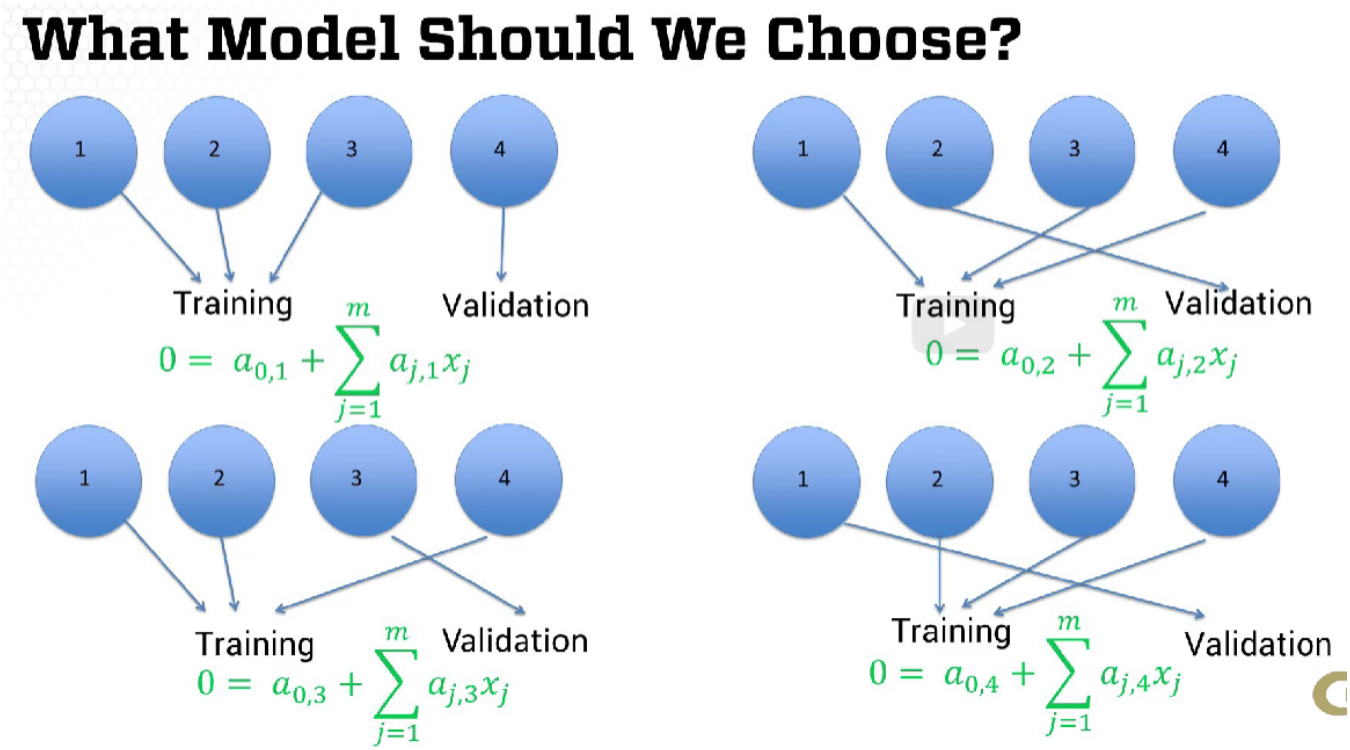
- Use cross-validation!



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

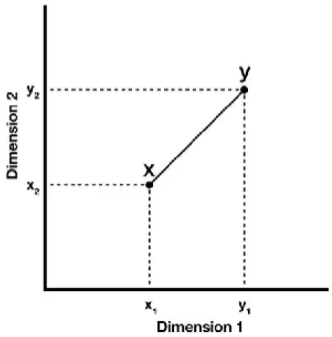
Euclidean (straight-line) distance

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

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.

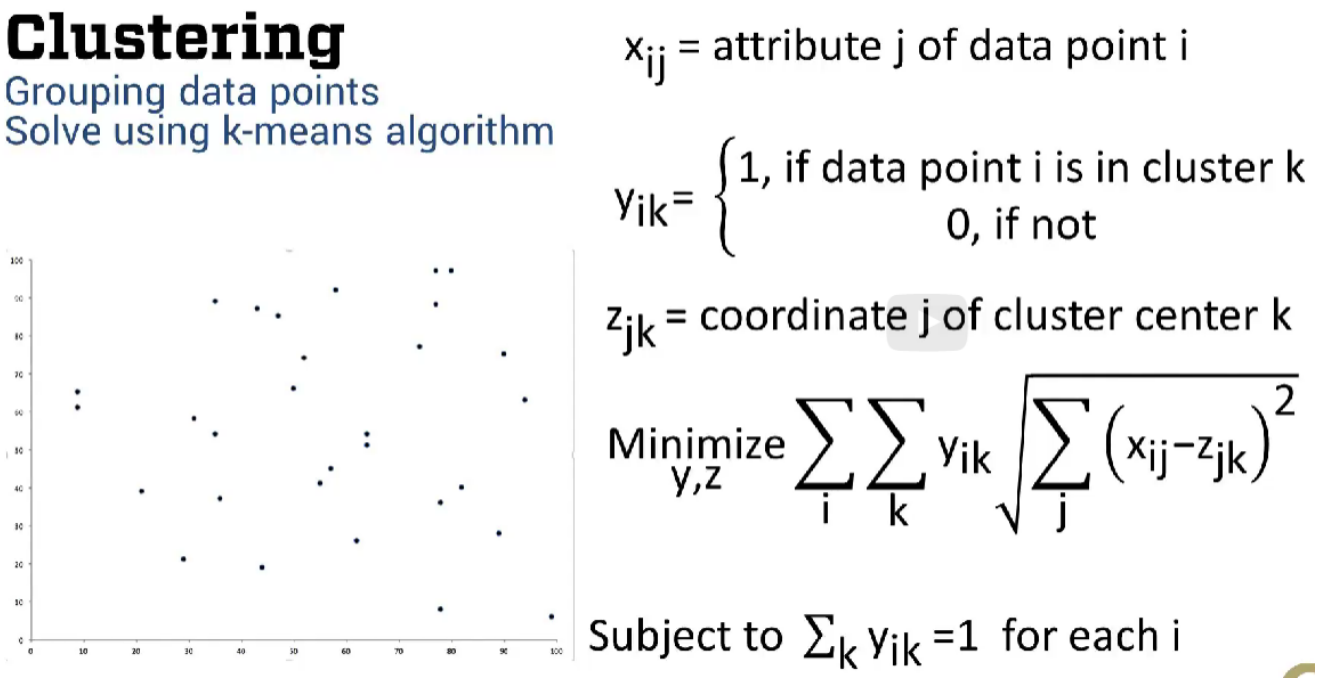
Straight-line distance $\sqrt[2]{\sum_{i=1}^{n}(x_i - y_i)^2}$ corresponds to which distance metric?

- ○ 1-norm
- ◉ 2-norm
- ○ ∞ norm

✔

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

Lesson 4.3 (M): K-means Clustering



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

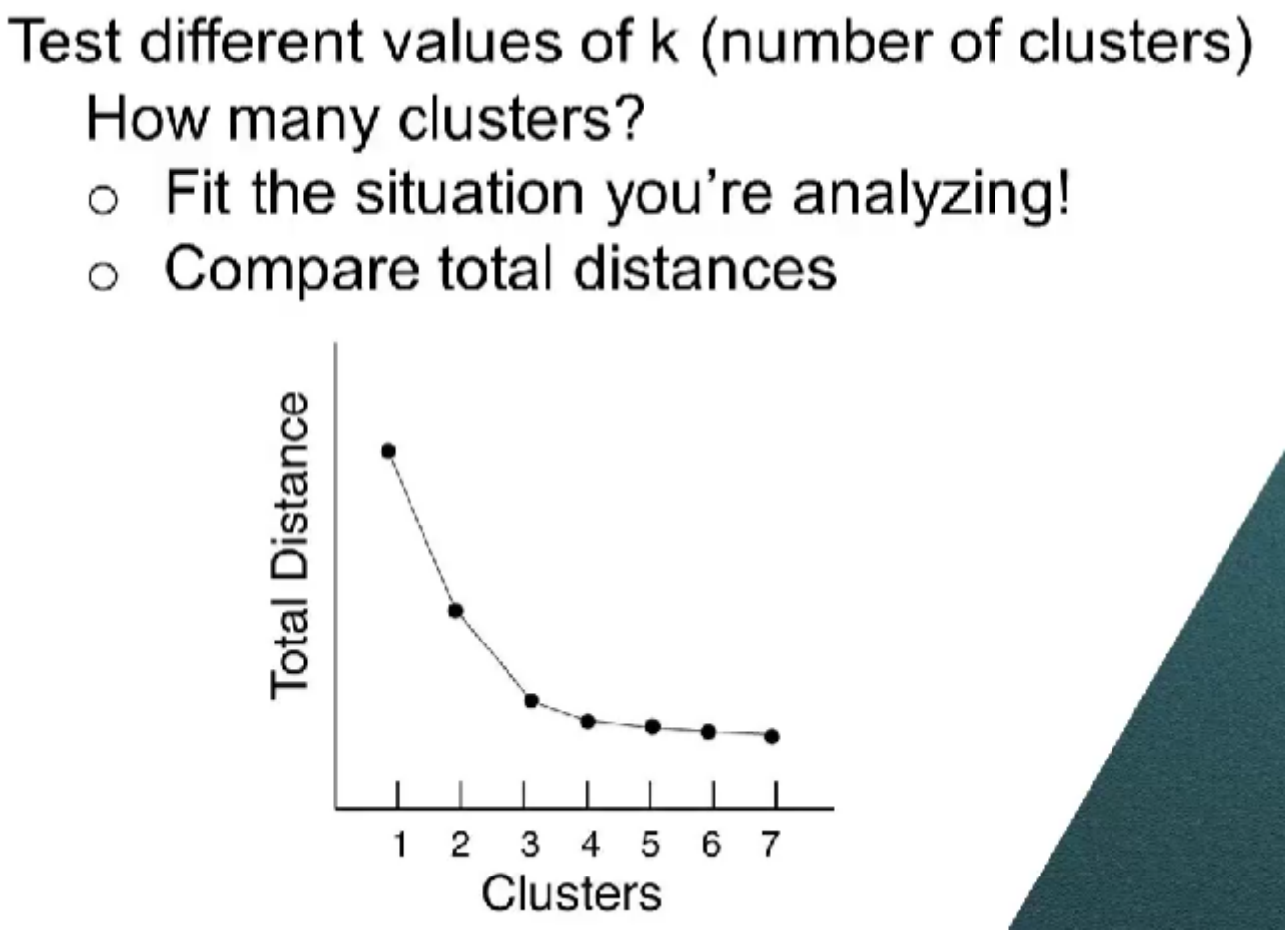
The last equation at the bottom right is just saying that each data point i can only belong to EXACTLY ONE cluster k.

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

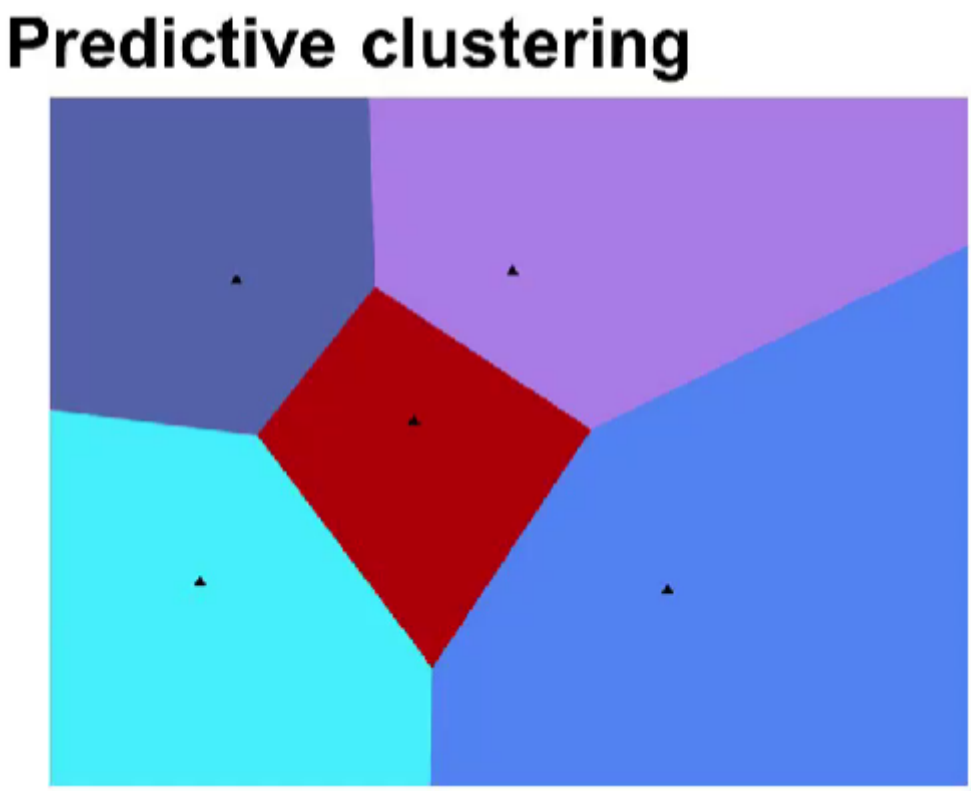


Suppose we find k-means clusterings for a bunch of different values of k, and for each one we calculate the total distance of each data point to its cluster center, we can plot that in two-dimensions.

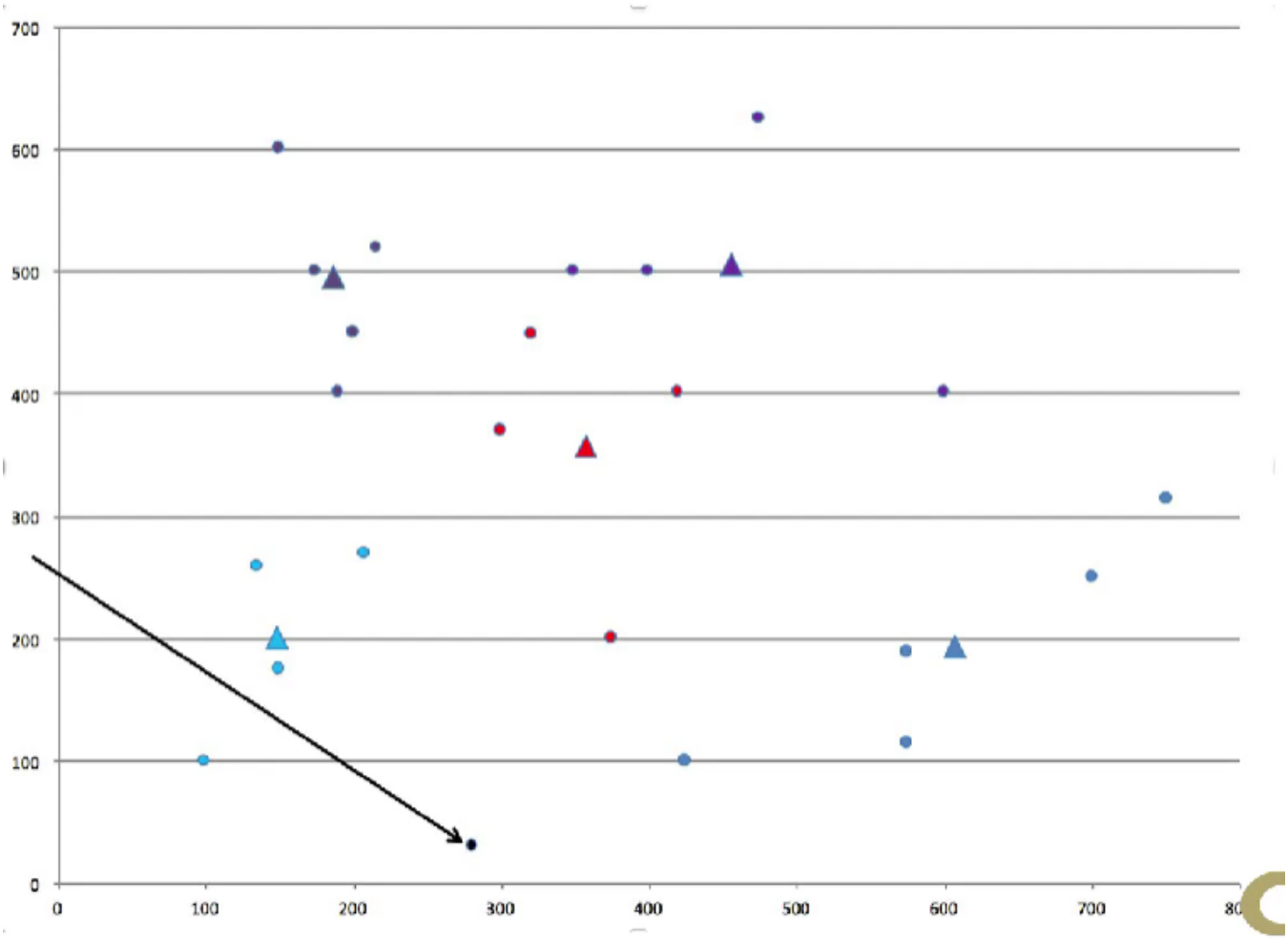
The horizontal axis is the number of clusters k, and the vertical axis is the total distance from points to cluster centers. Now we can look to see where the kink in the curve is. Here where the marginal benefit of adding another clusters starts to be small. This is an elbow plot.

Lesson 4.5 (M): Clustering for Prediction

Voronoi Diagram



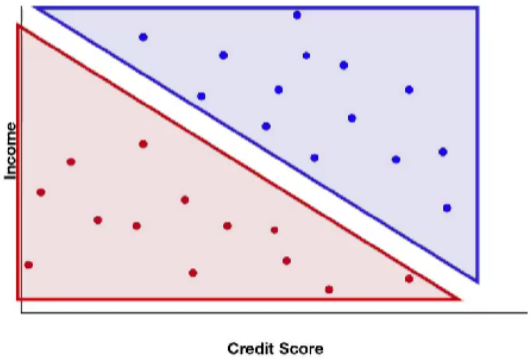
Predictive Clustering: If the new point isn't inside a cluster we can just choose whichever cluster center is closest and that's as reasonable a choice as any for predicting which cluster the new point is in.



Lesson 4.6 (M): Clustering vs Classification

Classification

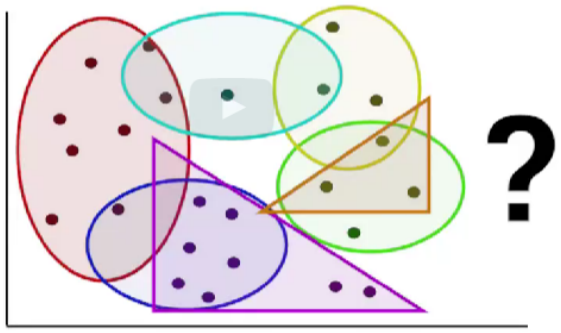
Grouping data points



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

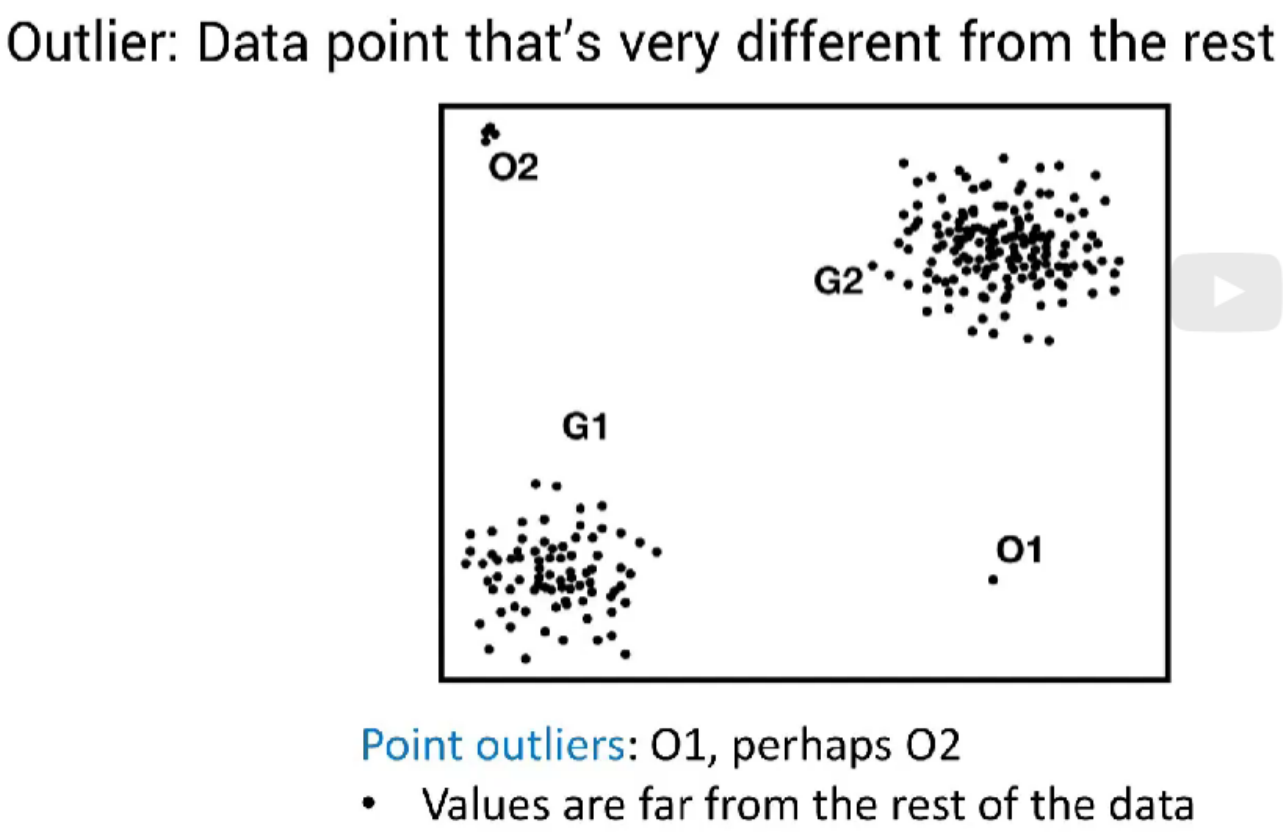
Extraneous Information

- complicates the model
- harder to correctly interpret the solution

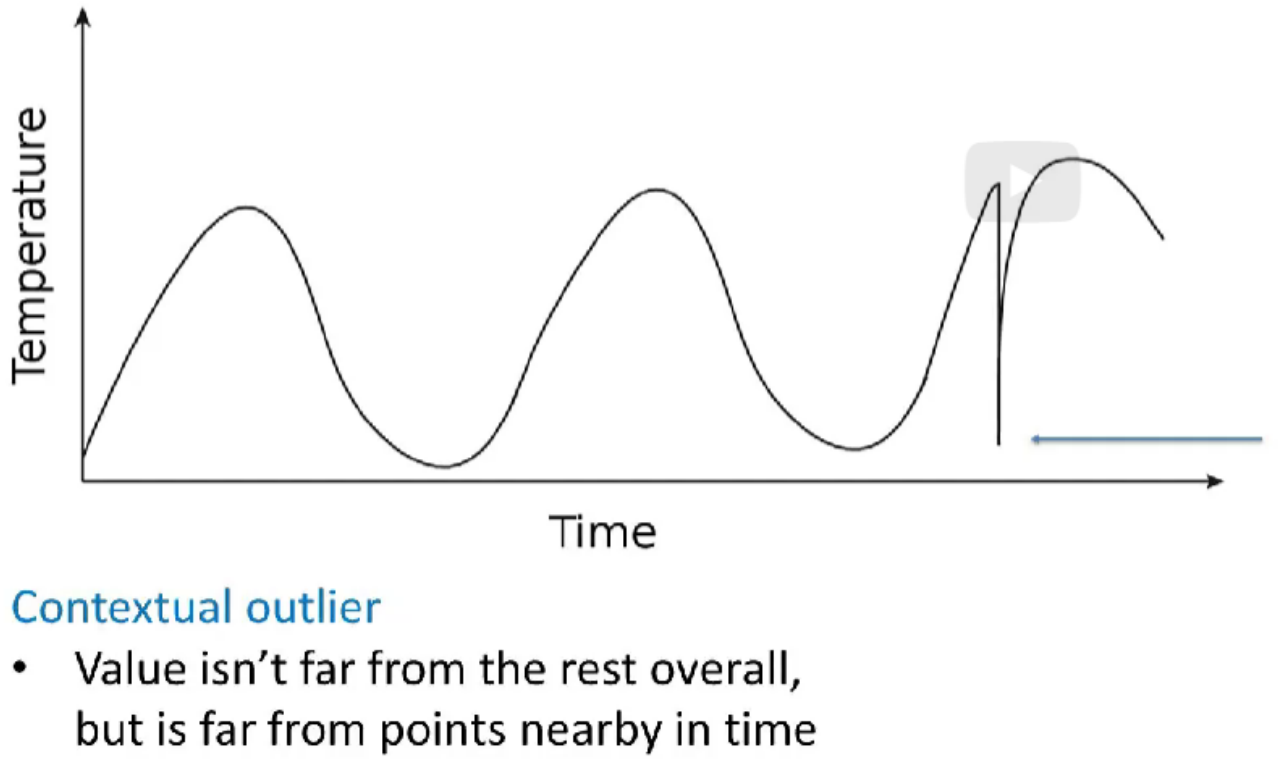
Lesson 5.2 (C): Outlier Detection

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

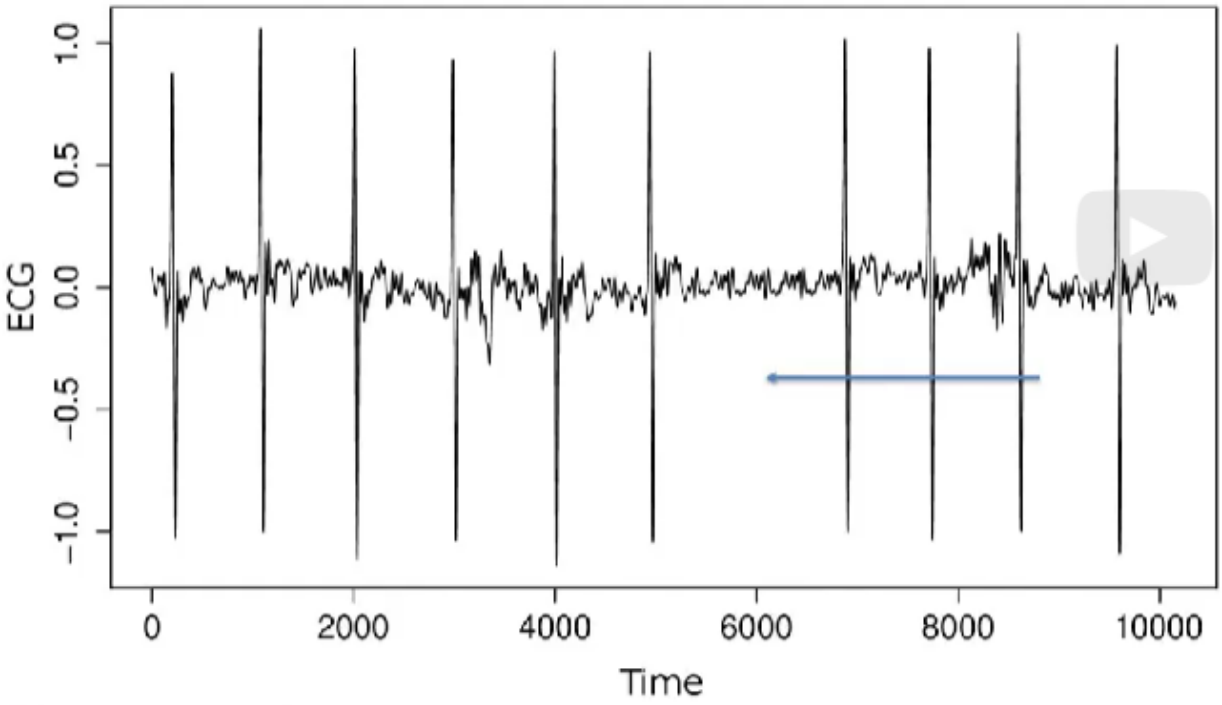


Contextual Outlier



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

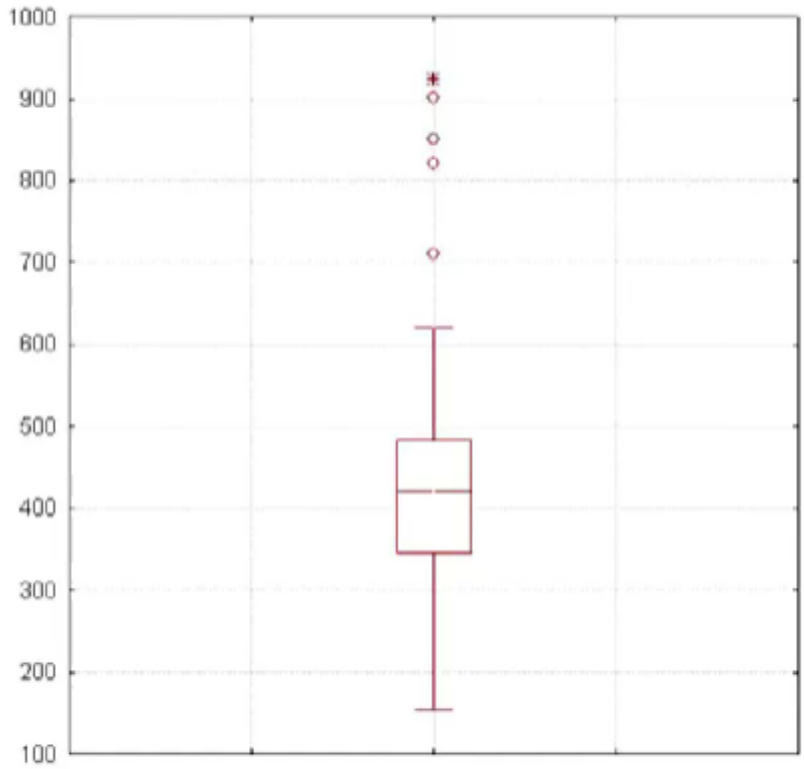
In this heartbeat data, it looks like there should be a large beat around times 6,000, but there isn't.

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



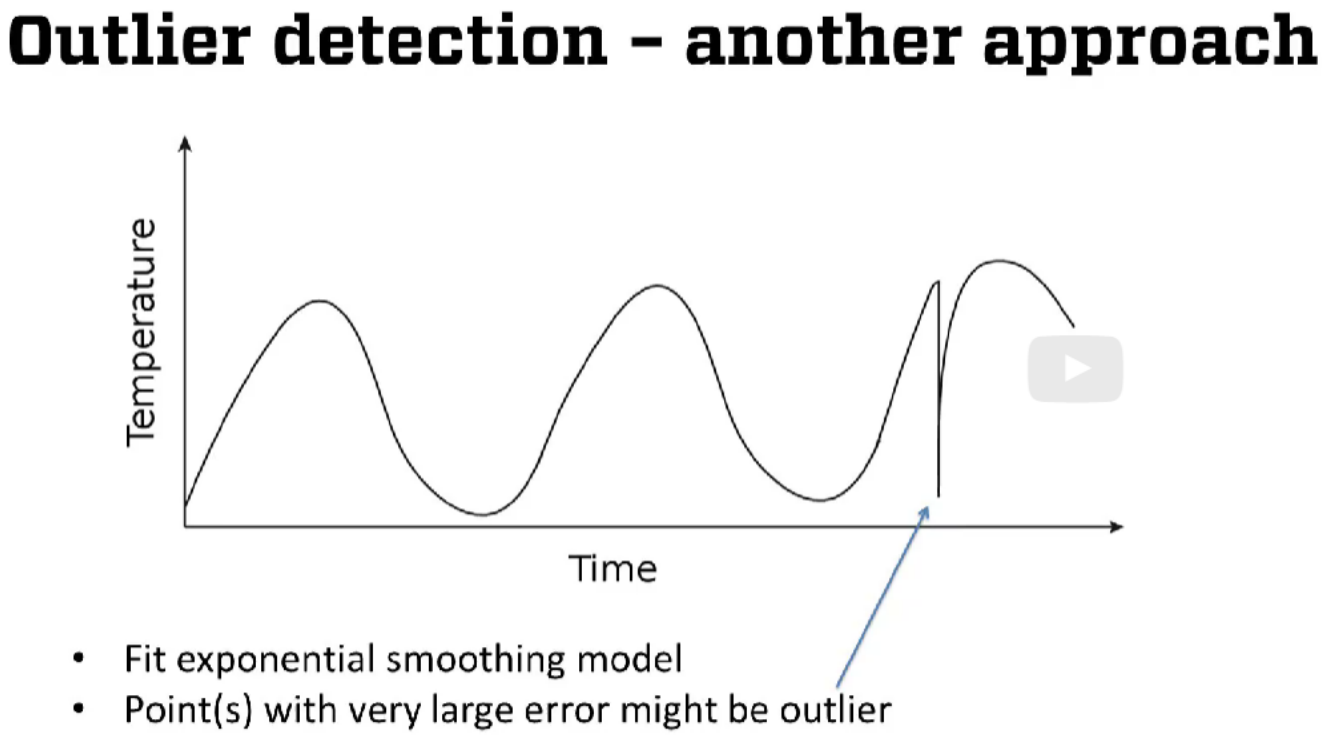
- Other automated methods also exist

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

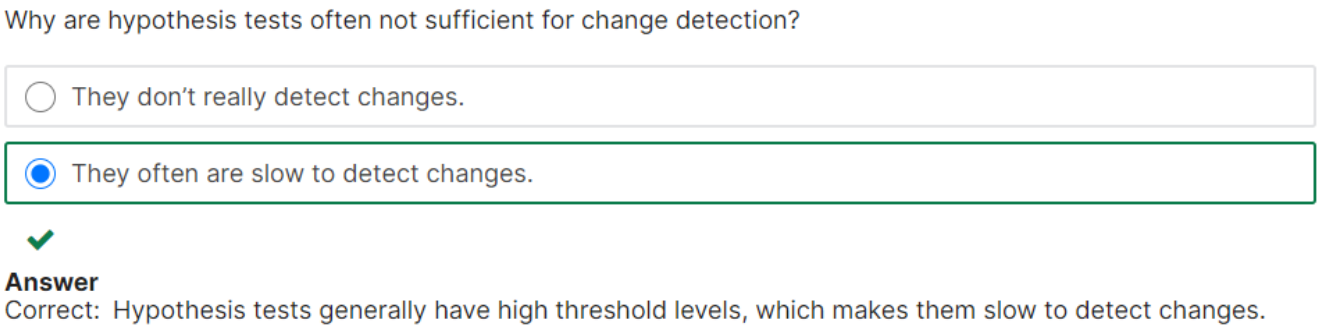
- Logistic regression model
  - Estimate probability of outliers happening under different conditions

- 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.]

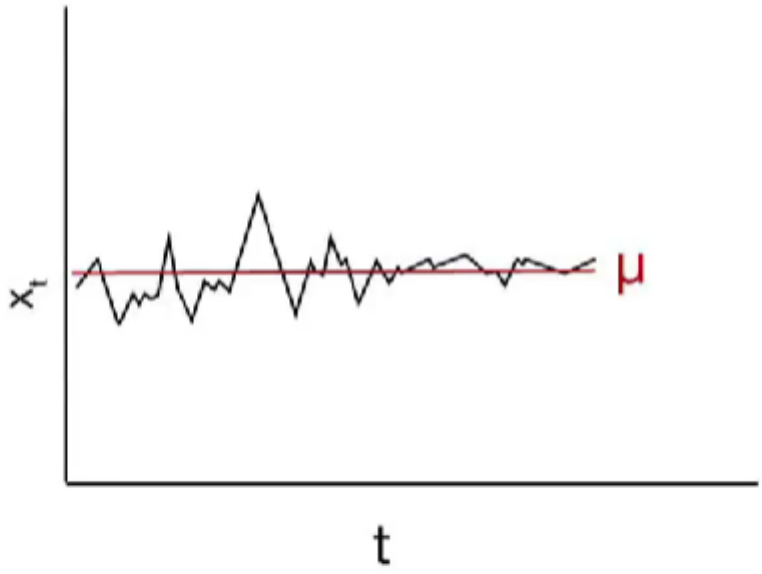
Lesson 6.2 (M): CUSUM for Change Detection

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.

C and T are model parameters that you have to use data to find the right values for.

And part of that decision depends on how costly it is if the model takes a long time to notice the change and how costly it is if the model thinks it has found the change that isn't really there.

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 $S_t \geq T$?

T = 450, C = 0

t	x_t	$X_t - \mu$	$X_t - \mu - C$	S_t
0				0
1	120	-15	-15	0
2	230	95	95	95
3	20	-115	-115	0
4	280	145	145	145
5	80	-55	-55	90
6	150	15	15	105
7	90	-45	-45	60
8	140	5	5	65
9	150	15	15	80
10	90	-45	-45	35
11	280	145	145	180
12	130	-5	-5	175
13	310	175	175	350
14	280	145	145	495
15	230	95	95	590
16	200	65	65	655
17	210	75	75	730
18	350	215	215	945
19	160	25	25	970
20	200	65	65	1035

Change

Here are the calculations using a threshold T of 450 and C equal to 0. You can see that it takes several time periods to notice the change, but it's still not too bad.

x_t = observed value at time t
μ = mean of x, if no change

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

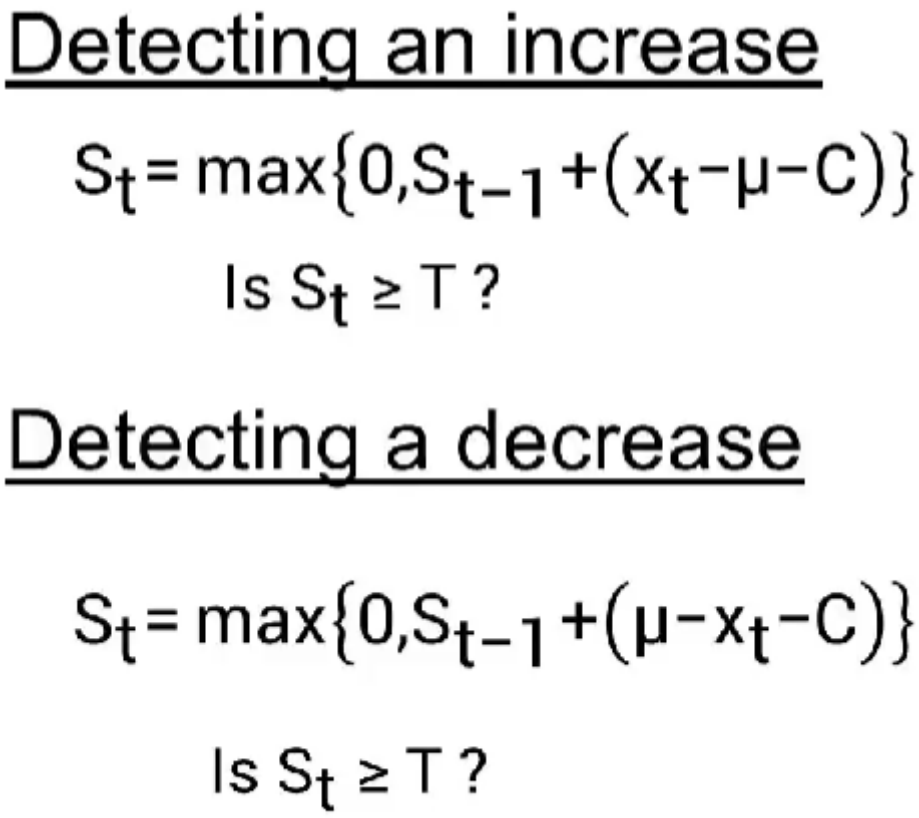
Is $S_t \geq T$?

T = 150, C = 0

t	x_t	$X_t - \mu$	$X_t - \mu - C$	S_t	
0				0	
1	120	-15	-15	0	
2	230	95	95	95	
3	20	-115	-115	0	
4	280	145	145	145	False change almost detected
5	80	-55	-55	90	
6	150	15	15	105	
7	90	-45	-45	60	
8	140	5	5	65	
9	150	15	15	80	
10	90	-45	-45	35	
11	280	145	145	180	Change detected
12	130	-5	-5	175	
13	310	175	175	350	
14	280	145	145	495	
15	230	95	95	590	
16	200	65	65	655	
17	210	75	75	730	
18	350	215	215	945	
19	160	25	25	970	
20	200	65	65	1035	

Change

And here are the same calculations using a threshold of T equals 150 and C equal to zero. As you can see, it detects the change much faster but it was also very close to falsely detecting a change early on.

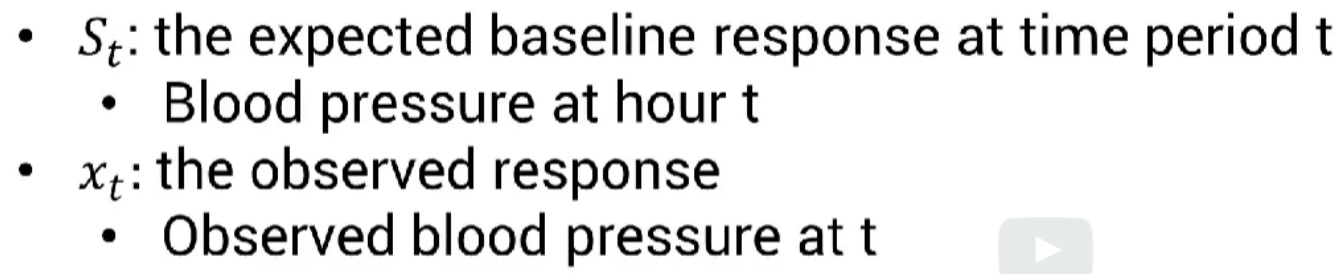


In the CUSUM model, having a higher threshold T makes it detect changes slower, and less likely to falsely detect changes.

Module 7: Time Series Models (M)

Lesson 7.1 (M): Introduction to Exponential Smoothing

Single Exponential Smoothing



We might think that the observed blood pressure is a real indicator of the baseline.

So S_t equals X_t.

Or we might think that there's no change to the baseline and the higher observed blood pressure today is just due to random luck.

So really S_t equals S_{t-1}.

Note that $S_1 = x_1$

- $S_t = \alpha x_t + (1 - \alpha) S_{t-1}$
- $0 < \alpha < 1$
  - $\alpha \rightarrow 0$: a lot of randomness in the system
  - $\alpha \rightarrow 1$: not much randomness in the system

If we think there's a lot of randomness in the system, then fluctuations are probably mostly due to randomness and we should make α closer to 0.

Yesterday's baseline is probably a good indicator of today's baseline even if we observed something different today.

And on the other hand, if there's not much randomness in the system, then we should make α closer to 1.

If we observe a fluctuation today it probably means today's baseline is close to the observed data.

In the exponential smoothing equation $S_t = \alpha x_t + (1 - \alpha) S_{t-1}$ a value of closer to 1 is chosen if...

- (●) There's less randomness, so we're more willing to trust the observation x_t
- ( ) There's more randomness, so we're more willing to trust the previous estimate S_{t-1}

✔

Answer

Correct: We put more weight on the observation x_t than the previous estimate S_{t-1}

Lesson 7.2 (M): Trends and Cyclic Effects

1. Single Exponential Smoothing

- For example
  - $S_t = \alpha x_t + (1 - \alpha) S_{t-1}$
  - $0 < \alpha < 1$
    - trade off between trusting x_t – when α is large
    - trusting S_{t-1} – when α is small
- The more randomness – trust previous estimate: S_{t-1}
- The less randomness – trust what you see: x_t

2. Double Exponential Smoothing

The inclusion of **trend** to the Single Exponential Smoothing Model makes it the Double Exponential Smoothing Model

- T_t: the trend at time period t
- $S_t = \alpha x_t + (1 - \alpha)(S_{t-1} + T_{t-1})$
- $T_t = \beta(S_t - S_{t-1}) + (1 - \beta)T_{t-1}$

- Initial condition
  - $T_1 = 0$

T_t equals another constant β times **the observed trend, which is the difference between the two baselines**, or S_t minus S_{t-1}, plus $1 - \beta$ times the previous trend estimate, T_{t-1}.

3. Triple Exponential Smoothing (Winter's Method or Holt-Winter's)

The Triple Exponential Smoothing Model is the Single Exponential Smoothing Model with the addition of **trend and (cyclical) seasonality**.

Cyclic patterns

- Like trend - additive component of formula
- Alternative
  - Seasonalities: multiplicative way
    - L: the length of a cycle
    - C_t: the multiplicative seasonality factor for time t
      - inflate or deflate the observation
  - New baseline formula (including trend and seasonality)

$$S_t = \alpha x_t / C_{t-L} + (1 - \alpha)(S_{t-1} + T_{t-1})$$

Note here that when we're using a cyclic factor to inflate or deflate the observed value, we use the cyclic factor from L time periods ago.

Why? Because that's the most recent cyclic factor we have from the same part of the cycle.

For example, if today is Monday, we use last Monday's cyclic factor.

Update the seasonal, or cyclic, factor in similar way

- $C_t = \gamma(x_t / S_t) + (1 - \gamma)C_{t-L}$
- $C_1, \ldots, C_L = 1$:
  No initial cyclic effect

Starting Conditions

Trend

- $T_1 = 0$
- Shows no initial trend

Multiplicative seasonality

- Multiplying by 1
- Shows no initial cyclic effect
- First L values of C set to 1

Lesson 7.3 (M): Exponential Smoothing: What The Name Means

$$S_t = \propto x_t + (1 - \propto) S_{t-1}$$

Example: $\propto = \frac{1}{2}$

$$S_t = \frac{1}{2} x_t + \frac{1}{2} S_{t-1}$$

High x_t: S_t not as high; pulled down by $(1 - \propto) S_{t-1}$

Low x_t: S_t not as low; pulled up by $(1 - \propto) S_{t-1}$

Exponential

$S_{t-1} = \propto x_{t-1} + (1 - \propto) S_{t-2}$

$S_t = \propto x_t + (1 - \propto) S_{t-1}$

$S_{t-2} = \propto x_{t-2} + (1 - \propto) S_{t-3}$

$S_t = \propto x_t + (1 - \propto)[\propto x_{t-1} + (1 - \propto) S_{t-2}]$

$= \propto x_t + (1 - \propto) \propto x_{t-1} + (1 - \propto)^2 S_{t-2}$

$S_t = \propto x_t + (1 - \propto) \propto x_{t-1} + (1 - \propto)^2 \propto x_{t-2} + (1 - \propto)^3 S_{t-3}$

$S_t = \propto x_t + (1 - \propto) \propto x_{t-1} + (1 - \propto)^2 \propto x_{t-2} + (1 - \propto)^3 \propto x_{t-3} + (1 - \propto)^4 S_{t-4}$

$S_t = \propto x_t + (1 - \propto) \propto x_{t-1} + (1 - \propto)^2 \propto x_{t-2} + (1 - \propto)^3 \propto x_{t-3} + (1 - \propto)^4 \propto x_{t-4} + (1 - \propto)^5 \propto x_{t-5} + \ldots$

Exponential

$$S_t = \propto x_t + (1-\propto) \propto x_{t-1} + (1-\propto)^2 \propto x_{t-2} + (1-\propto)^3 \propto x_{t-3} + (1-\propto)^4 \propto x_{t-4} + (1-\propto)^5 \propto x_{t-5} + \ldots$$

Every past observation contributes to the current baseline estimate

More-recent observations are more important

- Newer observations weighted more

Lesson 7.4 (M): Forecasting

Single Exponential Smoothing

- The basic exponential smoothing equation
  - $S_t = \alpha x_t + (1 - \alpha) S_{t-1}$
- Prediction
  - $S_{t+1} = \alpha x_{t+1} + (1 - \alpha) S_t$
  - x_{t+1} is unknown
  - Best guess: $x_{t+1} = S_t$
- our forecast for time period t+1
  - $F_{t+1} = \alpha S_t + (1 - \alpha) S_t$

    so, $F_{t+1} = S_t$
- $F_{t+k} = S_t$, k = 1, 2, ...

Double Exponential Smoothing

- Include the trend
  - $S_t = \alpha x_t + (1 - \alpha)(S_{t-1} + T_{t-1})$
  - $T_t = \beta(S_t - S_{t-1}) + (1 - \beta)T_{t-1}$
- The best estimate of the next baseline
  - the most current baseline estimate
- The best estimate of the trend
  - the most current trend estimate.
- Our forecast for time period t+1
  - $F_{t+1} = S_t + T_t$
- $F_{t+k} = S_t + kT_t$, k = 1, 2, ...

Triple Exponential Smoothing

- Include a multiplicative seasonality
  - $S_t = \alpha x_t / C_{t-L} + (1 - \alpha)(S_{t-1} + T_{t-1})$
- The best estimate of the next time periods seasonal factor
  - $C_{t+1} = C_{(t+1)-L}$
- our forecast for time period t+1
  - $F_{t+1} = (S_t + T_t)\, C_{(t+1)-L}$
- $F_{t+k} = (S_t + kT_t)\, C_{(t+1)-L+(k-1)}$, k = 1, 2, ...

Lesson 7.5 (M): ARIMA

ARIMA(p,d,q) model

$$D_{(d)t} = \mu + \sum_{i=1}^{p} \alpha_i D_{(d)t-i} - \sum_{i=1}^{q} \theta_i (\hat{x}_{t-i} - x_{t-i})$$

- dth order differences
- pth-order autoregression
- qth-order moving average

- Short-term forecasting
  - Better than exponential smoothing
    - When the data is more stable, with fewer peaks, valleys, and outliers
- Need 40 past data points for ARIMA to work well

Lesson 7.6 (M): GARCH

GARCH method - Generalized Autoregressive Conditional Heteroscedasticity

GARCH is a common approach for estimating variance

GARCH

- GARCH

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 + \sum_{i=1}^{q} \gamma_i \epsilon_{t-i}^2$$

- ARIMA

$$D_{(d)t} = \mu + \sum_{i=1}^{p} \alpha_i D_{(d)t-i} - \sum_{i=1}^{q} \theta_i (\hat{x}_{t-i} - x_{t-i})$$

Two differences from ARIMA

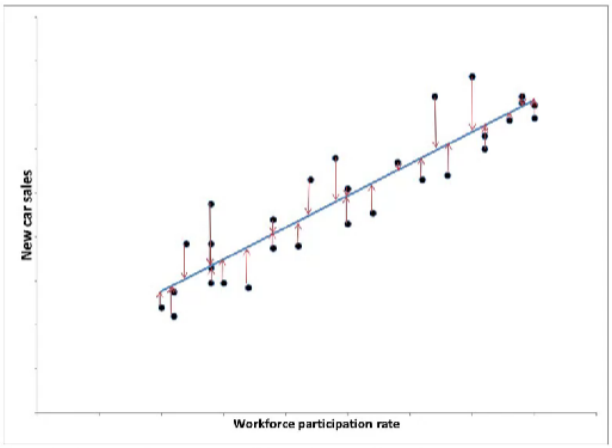
- Variances/squared errors
  - not observations/linear errors
- Raw variances
  - not differences of variances

Module 8: Basic Regression (M & C)

Lesson 8.1 (M): Introduction to Regression

Simple Linear Regression (SLR)

- Linear regression with one predictor



Look for linear relationship between predictor and response

y_i = cars sold for data point i
$\hat{y}_i$ = model's prediction of cars sold

Data point i prediction error

$$y_i - \hat{y}_i = y_i - (a_0 + a_1 x_{i1})$$

Sum of squared errors

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$= \sum_{i=1}^{n} \left(y_i - (a_0 + a_1 x_{i1})\right)^2$$

Best-fit regression line

- Minimizes sum of squared errors
- Defined by a_0 and a_1

Underlying math

- Minimize convex quadratic function
  - Set partial derivatives to zero
  - Solve simultaneous equations

Lesson 8.2 (C): Maximum Likelihood and Information Criteria

The most basic measure of model quality is likelihood.

Basically we assume that the observed data is the correct value and that we have information about the variance.

Then for any set of parameters we can measure the probability, really, the probability density that the model would generate the estimates it does.

Whichever set of parameters gives the highest probability density, called the maximum likelihood, is the best-fit set of parameters.

Maximum Likelihood – Example

- Error ~ N(0, σ^2), i.i.d.
- Observations: $z_1, \ldots, z_n$
- Model estimates: $y_1, \ldots, y_n$

Probability density for observing z_i if true value is y_i

$$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z_i-y_i)^2}{2\sigma^2}}$$

And because the errors are all **independent**, the joint probability of observing z_1 through z_n if the true values are y_1 through y_n is just the products of each of the n terms.

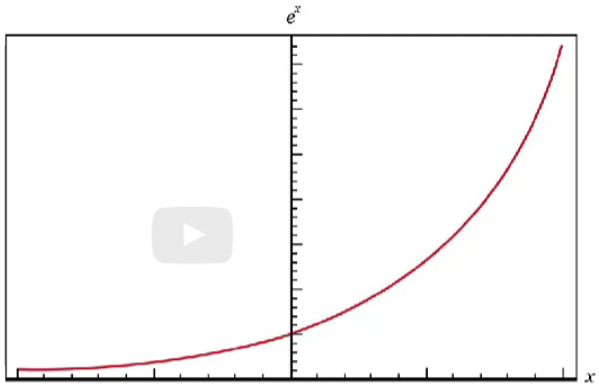
Joint density over all n terms

$$\prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z_i-y_i)^2}{2\sigma^2}}$$

$$= \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(z_i-y_i)^2}$$

Maximum Likelihood - Example

- Error ~ N(0, σ^2), i.i.d.
- Observations: z_1, ..., z_n
- Model estimates: y_1, ..., y_n
- MLE
  - the set of parameters that minimizes the sum of squared errors



$$maximize \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(z_i-y_i)^2} \rightarrow minimize \sum_{i=1}^{n}(z_i-y_i)^2$$

Linear regression

- $y_i = a_0 + \sum_{j=1}^{m} a_j x_{ij}$
- $\Sigma(z_i - y_i)^2$
- Minimize $\sum_{i=1}^{n}(z_i - y_i)^2$ over the parameters $a_0, \ldots, a_m$

$$Minimize \sum_{i=1}^{n} (z_i - (a_0 + \sum_{j=1}^{m} a_j x_{ij}))^2$$

AIC and BIC

The math behind these other methods is more complex and not so instructive to see at this point.

Akaike Information Criterion or AIC

Akaike Information Criterion (AIC)

- L^*: maximum likelihood value
- k : number of parameters being estimated

$$AIC = 2k - 2\ln(L^*)$$

- Penalty term: balances likelihood with simplicity
  - Helps avoid overfitting

In the case of regression we can substitute the likelihood function and the number of parameters is just m plus 1 for a_0 through a_m.

So the AIC could be calculated this way. Whichever model has the smallest AIC would be preferred.

Making AIC smaller encourages fewer parameters, k, and higher likelihood.

$$AIC = 2(m+1) - 2\ln\left(\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(z_i-\left(a_0+\sum_{j=1}^{m}a_j x_{ij}\right)\right)^2}\right)$$

- Prefer models with smaller AIC

Corrected AIC (AIC_c)

- Use for smaller data sets

$$AIC_c = AIC + \frac{2k(k+1)}{n-k-1}$$

$$= 2k - 2\ln(L^*) + \frac{2k(k+1)}{n-k-1}$$

Bayesian Information Criterion or BIC

Bayesian Information Criterion (BIC)

$$AIC = 2k - 2\ln(L^*)$$
$$BIC = k\ln(n) - 2\ln(L^*)$$

- L^*: maximum likelihood value
- k : number of parameters being estimated
- n : number of data points

- Similar to AIC
  - BIC's penalty term > AIC's penalty term
    - BIC encourages models with fewer parameters than AIC does
  - Only use BIC when there are more data points than parameters.

Lesson 8.3 (M): Using Regression

- $$Runs\ Scored = a_0 + a_1[Number\ of\ HR] + a_2[Number\ of\ Triples] + \ldots + a_7[Number\ of\ Stolen\ Bases]$$

- $$Runs\ Scored = a_0 + 1.4[Number\ of\ HR] + a_2[Number\ of\ Triples] + \ldots + a_7[Number\ of\ Stolen\ Bases]$$

1.4 for the coefficient of home runs.

In other words, every homerun hit will add an average of 1.4 runs to the team's total.

- $$Runs\ Scored = a_0 + a_1[Number\ of\ HR] + a_2[Number\ of\ Triples] + \ldots + a_7[Number\ of\ Stolen\ Bases]$$

- $$Adult\ Height = a_0 + a_1[Father's\ Height] + a_2[Mother's\ Height] + \ldots + a_4\ [Male\ or\ Female]$$

- Components of Analytics
  - **Descriptive Analytics**
  - **Predictive Analytics**

Regression is often good for describing and predicting, but is not as helpful for suggesting a course of action

---

Lesson 8.4 (C): Causation vs Correlation

Correlation does not imply causation

Lesson 8.5 (M): Transformation and Interactions

Transforming the Data

- We could adjust the data so the fit is linear

- Quadratic Regression: $y = a_0 + a_1 x_1 + a_2 x_1^2$
- Something a bit more fun: $y = a_0 + a_1 x_1^{1.5} + a_2 x_1^2 x_3 + a_3 \sin(x_2)$

- Response Transform: $\log(y) = a_0 + a_1 x_1 + \ \ldots \ + a_m x_m$
- Fun version 2: $\ln(y) = a_0 + a_1 x_1^2 x_2 x_3 + a_2 \log(x_3)$

- Box-Cox transformations: can be automated

Lesson 8.6 (M): Regression Output

P-Values

- Estimates the probability: the coefficient = 0
  - hypothesis testing
- p-value > 0.05
  - remove the corresponding attribute from the model
- Other thresholds besides 0.05 can also be used
  - higher thresholds - more factors can be included
    - possibility of including irrelevant factor
  - lower thresholds – less factors can be included
    - possibility of leaving out a relevant factor

Two warnings:

- With large amounts of data
  - p-values get small even when attributes are not at all related to the response
- P-values are only probabilities even when meaningful
  - For example:
    - 100 attributes - p-value of 0.02 each
    - so each will have 2% chance of not being significant
    - expect that 2 of them are really irrelevant

Coefficient

- When multiplied by the attribute value
  - Not much difference even if very low p-value

For example:

- Estimate household income with age as one of the attributes
  - if the regression coefficient is 1
  - even with a very low p-value,
  - the attribute really isn't very important
  - it's unlikely to make even a $100 difference

R-squared Value

- Estimate of how much variability your model accounts for
- For example:
  - R-squared value = 59%
  - accounts for about 59% of the variability in the data
  - the remaining 41%
    - randomness, or
    - other factors

Adjusted R-squared

- adjusts for the number of attributes used

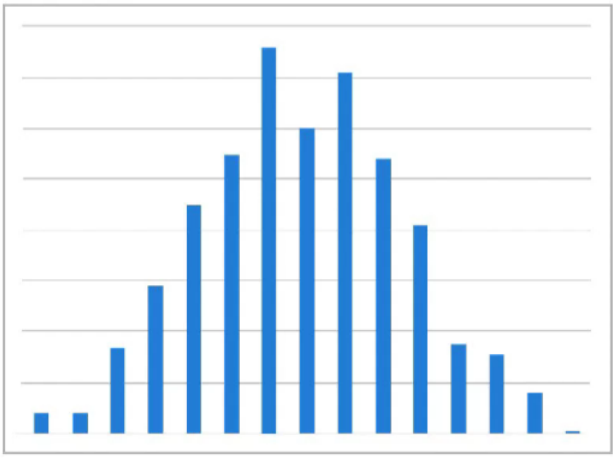
Most real-life systems have so much uncertainty and so many factors that such high R^2 values are very rare, and there's often significant value in understanding even 20-30% of the variability

Module 9: Advanced Data Preparation (C)

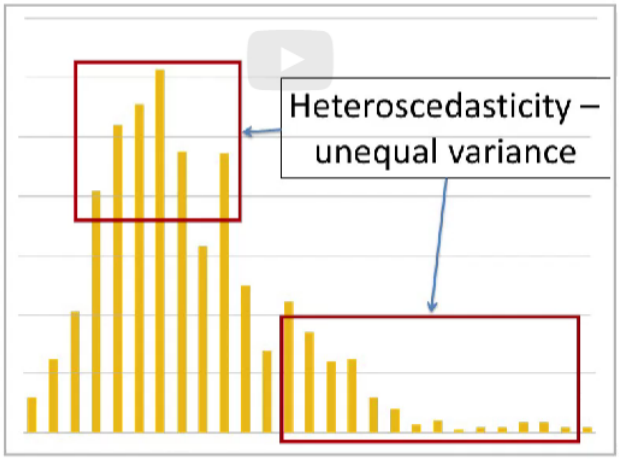
Lesson 9.1 (C): Box-Cox Transformation

Normality assumption

- Some models assume data is normally distributed
  - Results have bias when assumption is wrong



Height of American women in their 40s



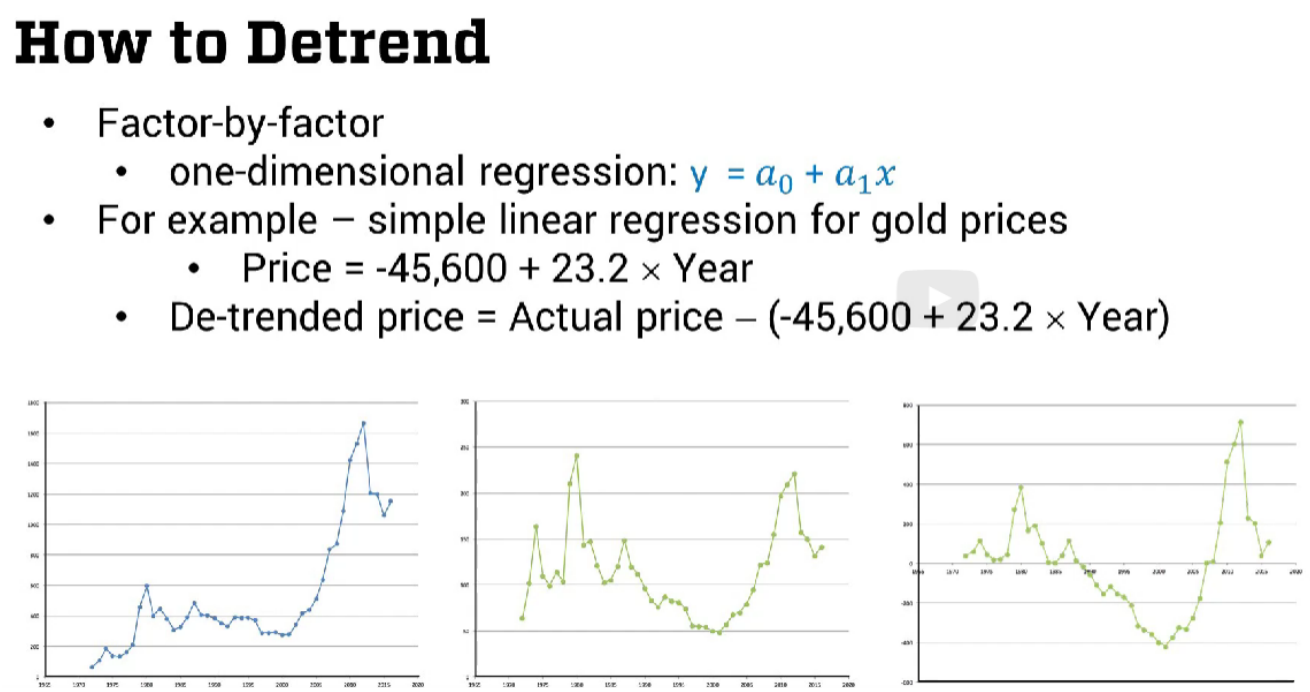
Weight of same American women in their 40s

Dealing with Heteroscedasticity.

Box-Cox transformation

- logarithmic transformation:
  - stretches out the smaller range to enlarge its variability,
  - shrinks the larger range to reduce its variability.
- $t(y) = (y^{\lambda} - 1)/\lambda$
  - $t(y)$ can become close to normal distribution
- Software can do it for you
  - Check whether you need the transformation (e.g., Q-Q plot)

Lesson 9.2 (C): De-Trending



As you can see, it's not so different from the inflation-adjusted graph. It's a little different because the linear fit assumes a constant rate of inflation, and the inflation-adjusted graph accounts for different inflation rates year by year. But if you don't have inflation data, you can see that the simple Detrending procedure

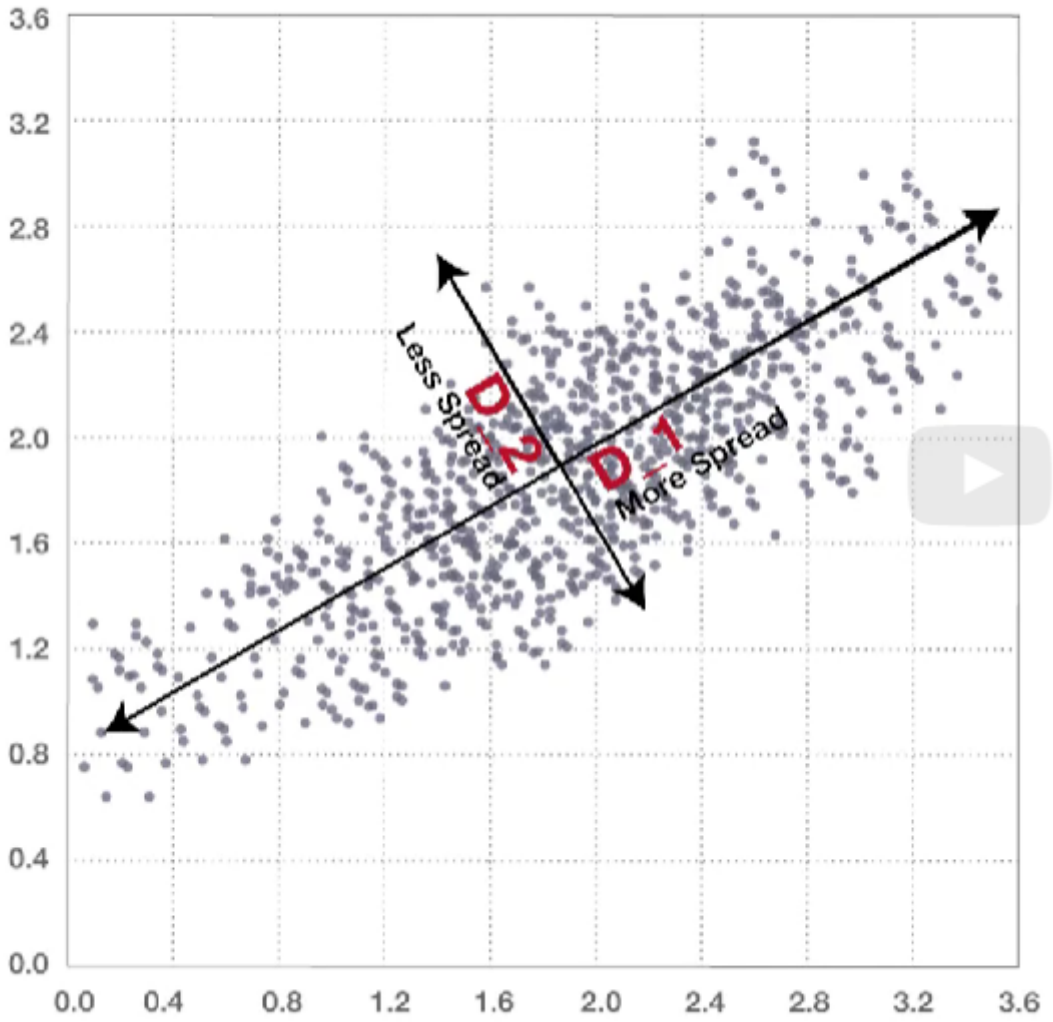
Detrending approach is pretty simple, and it usually works pretty well as a way to remove trend effects from time series data. When you want to use that data in a factor-based analysis.

You might want to de-trend data before using time-series data in a regression model

Lesson 9.3 (C): Introduction to Principal Component Analysis

PCA is a way to transform data to do two things.

- It changes the coordinates to remove correlation and
- it ranks the coordinate dimensions in order of the amount of variance in each so the most important coordinates are first.



As you can see the data has a much wider spread of values in the d1 direction than it does in the d2 direction.

PCA will recognize that and automatically make d1 the first dimension and d2 the second dimension.

If we just want to use a one-dimensional factor we know that d1 is a better bet than d2.

Lesson 9.4 (C): Using Principal Component Analysis

X: Initial matrix of data; x_{ij} is the j[th] factor of data point i

- Scale such that $\frac{1}{m}\sum_i x_{ij} = \mu_j = 0$

Find all of the eigenvectors of $X^T X$

- V: Matrix of eigenvectors (sorted by eigenvalue)
- $V = [V_1\ V_2\ \ldots]$, where V_j is the j[th] eigenvector of $X^T X$

PCA – linear transformation

- First component is XV_1, second component is XV_2, etc.
- k[th] new factor value for the i[th] data point: $t_{ik} = \sum_{j=1}^{m} x_{ij} v_{jk}$

Then the principal components are just the matrix X times the matrix V.

The first principal component is X times the first column of V.

The second principal component is X times the second column of V, etc.

In other words, V is a linear transformation of the data from X to the principal components.

Unpack the above slide

X: Initial matrix of data; x_{ij} is the j[th] factor of data point i

- Scale such that $\frac{1}{m}\sum_i x_{ij} = \mu_j = 0$

This line just means that the data is normalised since μ is zero

i is the row index and j is the column index which corresponds to the respective factors/categories

Find all of the eigenvectors of $X^T X$

- V: Matrix of eigenvectors (sorted by eigenvalue)
- $V = [V_1\ V_2\ \ldots]$, where V_j is the j[th] eigenvector of $X^T X$

Each Eigenvector corresponds to each factor/category

They are sorted by eigenvalue in **descending** order

PCA – linear transformation

- First component is XV_1, second component is XV_2, etc.
- k[th] new factor value for the i[th] data point: $t_{ik} = \sum_{j=1}^{m} x_{ij} v_{jk}$

XV_1 will give a column vector (first principal component) with i rows and XV_1 will correspond with the first highest ranked factor by eigenvalue.

V is a linear transformation of the data from X to the principal components.

Each new factor will be a linear combination of the original factors.

Note that each v_{jk} are real numbers and can be seen as the principal components' regression coefficients. (double check this fact)

Here's the formula for each t_{ik}, the kth new factor value for the ith data point.

A Summary of the PCA Approach

Source: https://sebastianraschka.com/Articles/2015_pca_in_3_steps.html (https://sebastianraschka.com/Articles/2015_pca_in_3_steps.html)

- Standardize the data.
- Obtain the Eigenvectors and Eigenvalues from the covariance matrix or correlation matrix, or perform Singular Value Decomposition.
- Sort eigenvalues in descending order and choose the k eigenvectors that correspond to the k largest eigenvalues where k is the number of dimensions of the new feature subspace ($k \leq d$).
- Construct the projection matrix W from the selected k eigenvectors.
- Transform the original dataset X via W to obtain a k dimensional feature subspace Y

PCA - Regression

Interpret the new model, in terms of the original factors?

Example (Regression): PCA finds new L factors $\{t_{ik}\}$, then regression finds coefficients $b_0, b_1, \ldots, b_L$

$$\begin{aligned} y_i &= b_0 + \sum_{k=1}^{L} b_k t_{ik} \\ &= b_0 + \sum_{k=1}^{L} b_k \left[\sum_{j=1}^{m} x_{ij} v_{jk}\right] \\ &= b_0 + \sum_{j=1}^{m} x_{ij} \left[\sum_{k=1}^{L} b_k v_{jk}\right] \\ &= b_0 + \sum_{j=1}^{m} x_{ij} [a_j] \end{aligned}$$

Implied regression coefficient for x_j

$$a_j = \sum_{k=1}^{L} b_k v_{jk}$$

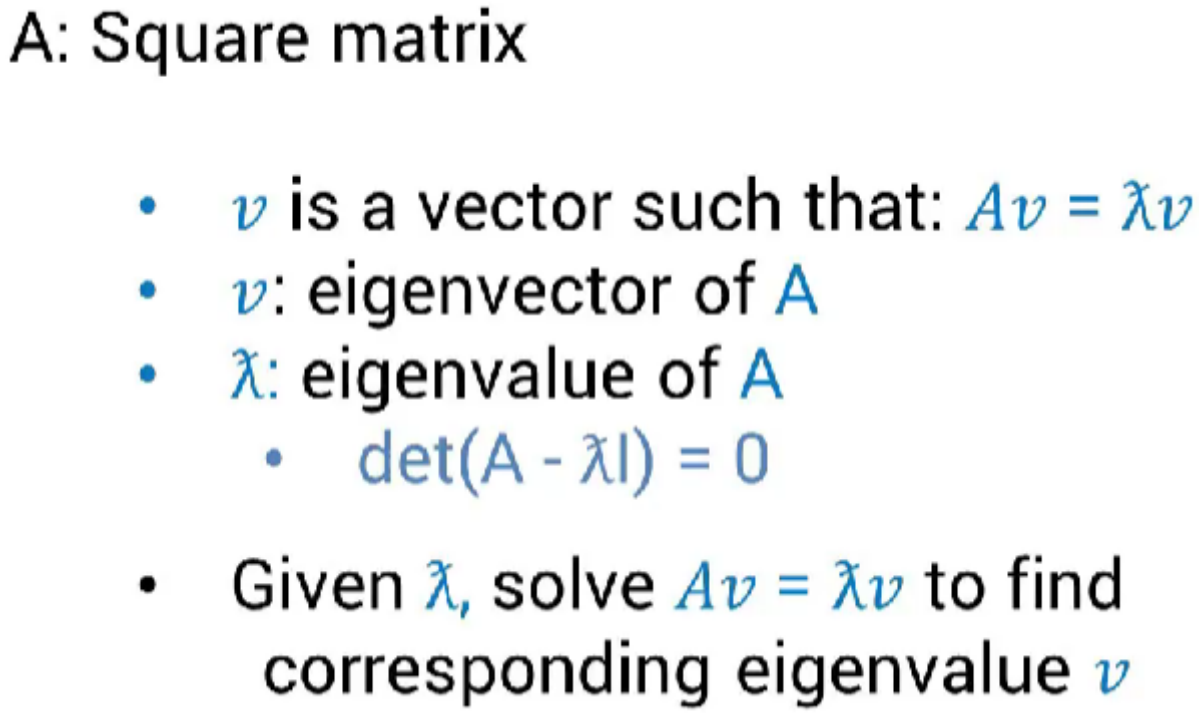
If you use principal component analysis (PCA) to transform your data and then you run a regression model on it, how can you interpret the regression coefficients in terms of the original attributes?

- ○ The first coefficient corresponds to the first attribute in your original data set, the second coefficient corresponds to the second attribute, etc.
- ◉ Each original attribute's implied regression coefficient is equal to a linear combination of the principal components' regression coefficients.
- ○ You can't.

✔

Answer
Correct: This is equivalent to using the inverse transformation.

Lesson 9.5 (C): Eigenvalues and Eigenvectors



Eigenvectors are orthogonal to each other

Lesson 9.6 (C): The Good and Bad of PCA

Just watch the video.

So using PCA to reduce the dimension of the dataset without losing too much explanatory or predictive power isn't always helpful, but overall, it's often an approach that can help reduce dimension without losing too much.

Module 10: Advanced Regression (M&C)

Lesson 10.1 (M): Introduction to CART

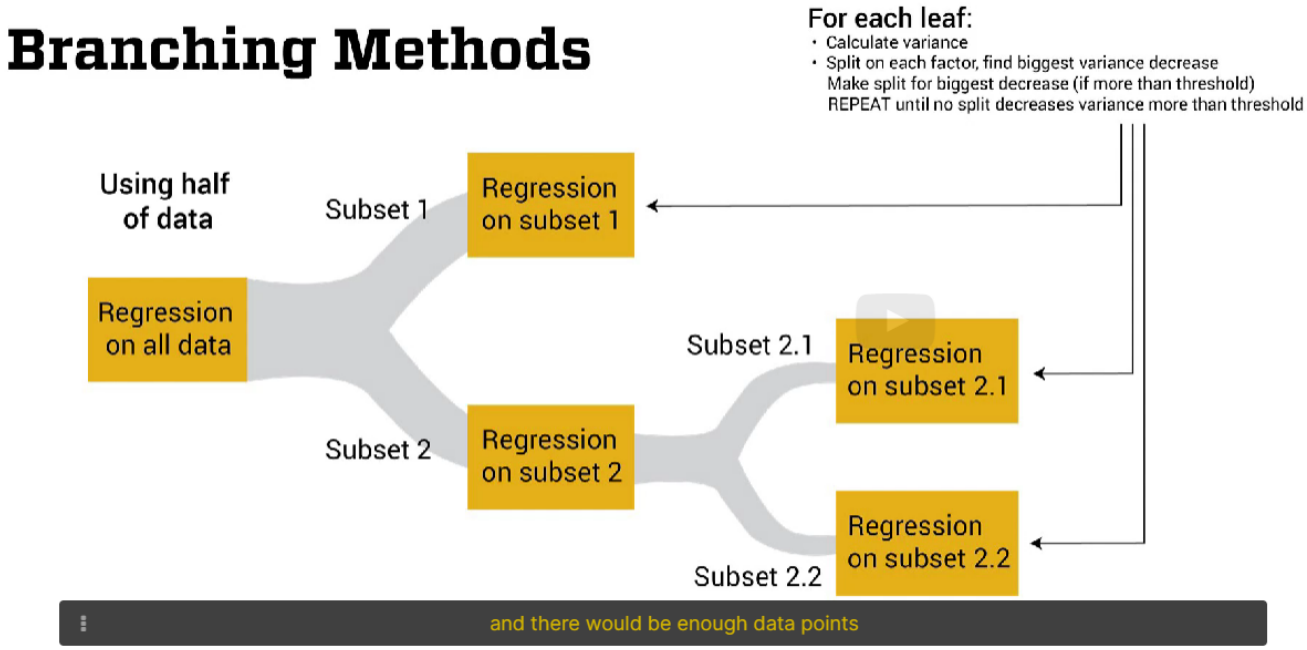
CART = Classification and Regression Trees

Endpoints of Trees are called Leaves

Each leaf's individual model is tailored to the subset of data points that follow all of the branches leading to the leaf.

Tree-based approaches can be used for other models besides regression.
For example, a classification tree might have a different SVM or KNN model at each leaf. It might even use SVM at some leaves and KNN at others (though that's probably rare).

Lesson 10.2: (M): Branching



By Variance I believe what they mean is similar to the derivation of the Gini Purity. Where Variance denotes impurity, ie a mix of different factors. We want the leaf to be as pure as possible.

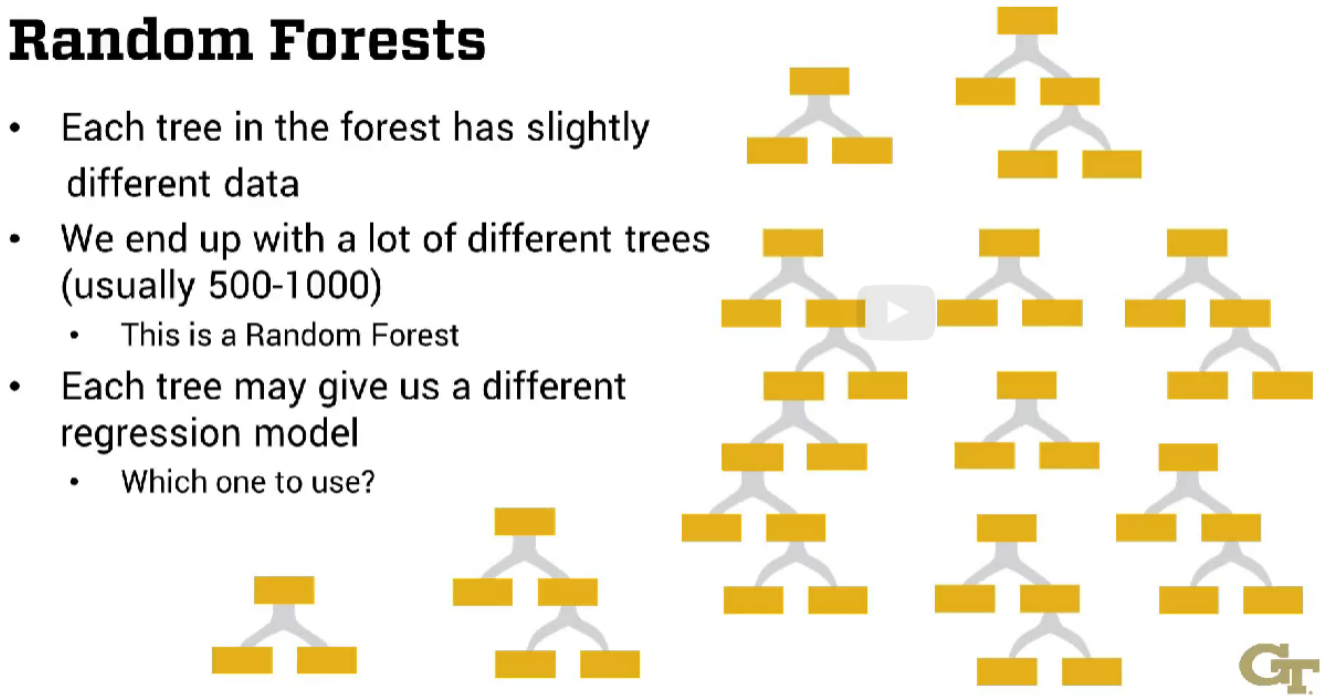
- Key Ideas
  - Using a metric related to the model's quality
  - Find the "best factor" to branch with
  - Check: *Did this really improve the model?*
    - If not, prune the branch back

- Rejecting a potential branch
  - Low improvement benefit
  - One side of the branch has too few data points now
    - Rule of thumb: Each leaf contains at least 5% of the original data

"Overfitting our model can be costly; make sure the benefit of each branch is greater than its cost"

Fitting to very small subsets of data will cause overfitting.

Lesson 10.3: (M): Random Forests



If it's a regression tree, we use the average predicted response over all of the trees in our forest. And if it's a classification tree, we use the mode, the most common predicted response over all the trees in our forest.

Benefits	Drawbacks
• Better overall estimates • Averages between trees somewhat neutralizes over-fitting	• Harder to explain/ interpret results • Can't give us a specific regression or classification model from the data

Lesson 10.3a (C): Explainability / Interpretability

Linear Regression Example

$$y = a_0 + \sum_{j=1}^{n} a_j x_{ij}$$

How is the value of y affected by different values of the predictors?

$a_0 = 1{,}000{,}000 \quad a_1 = 0.25 \quad a_2 = -1{,}000{,}000$
$a_3 = -1{,}000{,}000 \quad a_4 = 20{,}000$

Baseline = 1,000,000 tickets (a_0)
Star salary: each dollar increases sales by 1/4 (a_1)
Similar movies: each decreases sales by 1,000,000 (a_2)
Restrictive rating: decreases sales by 1,000,000 (a_3)
Days left in year: each day increases sales by 20,000 (a_4)

y	Number of tickets to this movie sold this year
x_1	Salary of top four stars
x_2	Number of movies with similar plots this year
x_3	Rated R or more restrictive? (1=yes, 0=no)
x_4	Number of days left in year

Tradeoff

Less-explainable models

- Can give more value
  - by fitting to more-complex patterns

More-explainable models

- Can be more likely to be adopted
  - because they're easier for decision-makers to believe in
- Sometimes legally required

Pay attention to the tradeoffs when suggesting a model

Lesson 10.4: (M): Logistic Regression

Standard Linear Regression

- $y = a_0 + a_1x_1 + a_2x_2 + \ldots + a_jx_j$

Logistic Regression Model

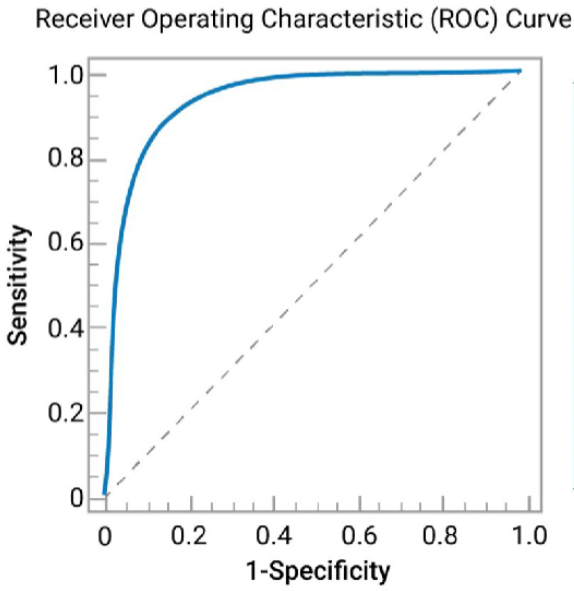
- p: the probability of the event you want to observe
- $log\frac{p}{1-p} = a_0 + a_1x_1 + a_2x_2 + \ldots + a_jx_j$
- $p = \frac{1}{1+e^{-(a_0+a_1x_1 + a_2x_2 + \ldots + a_jx_j)}}$
  - If $a_0+a_1x_1 + a_2x_2 + \ldots + a_jx_j = -\infty$ then p = 0
  - If $a_0+a_1x_1 + a_2x_2 + \ldots + a_jx_j = +\infty$ then p = 1

Measures of model quality

- Linear regression
  - R-squared value
    - Fraction of variance explained by model
- Logistic regression
  - Pseudo R-squared value
    - Not really measuring fraction of variance

R-Squared (R^2 or the coefficient of determination) is a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable.

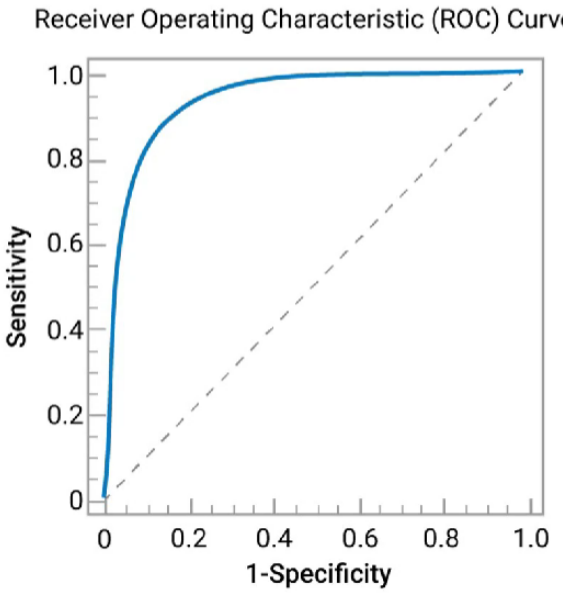
ROC Curve



Sensitivity = $\frac{\text{(True Positives)}}{\text{(True Positives + False Negatives)}}$

Specificity = $\frac{\text{(True Negatives)}}{\text{(True Negatives + False Positives)}}$

See the Lesson (10.5) on Confusion Matrices



Joe – Repaid the loan
Moe – Did not repay the loan

AUC = Probability that the model give Joe's data point a higher response value than Moe's

For reference: AUC = 0.5 – Just guessing

A common way of looking at this is to use a receiver operating characteristic curve or ROC. In this graph, we plot the sensitivity and 1 minus the specificity of the model for each threshold. The area under that curve, creatively called AUC, shows the probability that if we choose a random person from the yes group and one from the no group, the yes person has a higher estimate in the model.

Lesson 10.5: (C): Confusion Matrices

Confusion Matrix

- True positive (TP): point in the category, correctly classified
- False positive (FP): point not in category, model says it is
- True negative (TN): point not in category, correctly classified
- False negative (FN): point in the category, model says no

True Classification \ Model's Classification	Yes	No
Yes	True Positive (TP)	False Negative (FN)
No	False Positive (FP)	True Negative (TN)

Guidelines:

- Positive - model says it's in the category
- Negative - model says it's not in the category
- True - model got it right
- False - model got it wrong

Lesson 10.6: (C): Situationally-Driven Comparison

Evaluating a model's quality

- Example using confusion matrix

Cost of lost productivity

$0 for correct classifications

$0.04 to read spam

$1 to miss a real message

Example from spam detection

True Classification \ Model's Classification	Real	Spam
Real	True Positive (TP) 490	False Negative (FN) 10
Spam	False Positive (FP) 100	True Negative (TN) 400

- If 50% of email is spam:
  Total cost = **1.4 cents/email**
  17% of inbox is spam

Total cost =

$450 \times \$0 + 50 \times \$1 + 50 \times \$0.04 + 450 \times \$0 = \$52$

True Classification \ Model's Classification	Real	Spam
Real	True Positive (TP) 450	False Negative (FN) 50
Spam	False Positive (FP) 50	True Negative (TN) 450

- If 50% of email is spam:
  Total cost = **5.2 cents/email**
  10% of inbox is spam

Lesson 10.7: (M): Advanced Topics in Regression

- poisson regression

- regression splines
- bayesian regression
- KNN Regression