Data Science Tutorial

Eliezer Kanal – *Technical Manager, CERT*

Daniel DeCapria – *Data Scientist, ETC*

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM17-0509

About us

Eliezer Kanal

Technical Manager, CERT

Recent projects:

- ML-based Malware Classifier
- Network traffic analysis
- Cybersecurity questionnaire optimization

Daniel DeCapria

Data Scientist, ETC

Recent projects:

- Cyber risk situational dashboard
- Big Learning benchmarks

Today’s presentation – a tale of two roles

The call center manager

Introduction to
data science capabilities

The master carpenter

Overview of the
data science toolkit

Call center manager

First day on job… welcome!

Goal: Reduce costs

Task: Keep calls short!

Data:

Average call time:	5.14 minutes (5:08)… very long!
Number of employees:	300
Average calls per day:	~28,000

Call center manager – *Gather data*

Get the data!

- Where is it?
- What will you use to analyze it?
- How accurate it is?
- How complete is it?
- Is it too big to easily read?

Data cleaning = 90% of the work

2 weeks (10 days) = 9 cleaning, 1 analyzing

Cleaning the Data – *Structuring the Data*

Goal: Organize data in a table, where…

Columns = descriptor (age, weight, height)

Row = individual, complete records

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
1	0.0063	18	2.3100	0	0.5380	6.5750	65.2000	4.0900	1	296	15.3000	396.9000	4.9800	24
2	0.0273	0	7.0700	0	0.4690	6.4210	78.9000	4.9671	2	242	17.8000	396.9000	9.1400	21.6000
3	0.0273	0	7.0700	0	0.4690	7.1850	61.1000	4.9671	2	242	17.8000	392.8300	4.0300	34.7000
4	0.0324	0	2.1800	0	0.4580	6.9980	45.8000	6.0622	3	222	18.7000	394.6300	2.9400	33.4000
5	0.0691	0	2.1800	0	0.4580	7.1470	54.2000	6.0622	3	222	18.7000	396.9000	5.3300	36.2000
6	0.0299	0	2.1800	0	0.4580	6.4300	58.7000	6.0622	3	222	18.7000	394.1200	5.2100	28.7000
7	0.0883	12.5000	7.8700	0	0.5240	6.0120	66.6000	5.5605	5	311	15.2000	395.6000	12.4300	22.9000
8	0.1446	12.5000	7.8700	0	0.5240	6.1720	96.1000	5.9505	5	311	15.2000	396.9000	19.1500	27.1000
9	0.2112	12.5000	7.8700	0	0.5240	5.6310	100	6.0821	5	311	15.2000	386.6300	29.9300	16.5000
10	0.1700	12.5000	7.8700	0	0.5240	6.0040	85.9000	6.5921	5	311	15.2000	386.7100	17.1000	18.9000
11	0.2249	12.5000	7.8700	0	0.5240	6.3770	94.3000	6.3467	5	311	15.2000	392.5200	20.4500	15
12	0.1175	12.5000	7.8700	0	0.5240	6.0090	82.9000	6.2267	5	311	15.2000	396.9000	13.2700	18.9000
13	0.0938	12.5000	7.8700	0	0.5240	5.8890	39	5.4509	5	311	15.2000	390.5000	15.7100	21.7000

How can you get data out of these documents?

Less structure

More structure

Cleaning the Data

Even when you think your data should be clean, it might not be…

Please tell us how many years of experience you have had working in the following domains. Enter a whole number.

Machine Learning

0.5 | 2 | 1 | 0 | 1/2 | none | 0 semesters | 6 months

Computer Science

1.5 | this semester | 3 | 2 | 1 | 0 | 6 | 5 | 4 | 8 | 11 | second | .5 | 6 months

Mathematics

0.5 | .333 | 22 | 3 | some background in calculus | 2 | 1 | 0 | 6 | 5 | 4 | 8 | 10+ | 16 | fourth | 10 | 11 years | 7 semesters | 3.5

Cleaning the Data – *Call Center Example*

Name	Mgr	Dir	Call Length	Phone Line	Problem solved?	Comment
Beth Jones	Dan Thomas	Anne Kim	1:30	1	Y	5 …
Beth Jones	Dan Thomas	Anne Kim	1:52	3	Y	…
1 Jones, Beth	Dan Thomas	Anne Kim	1 90	2	Y	…
Tom Keane	Mark Ryan	Tim Pike	88	2	N	…
Tom Keane	2 Mark Ryan	Tim Pike	144	3	1 No	…
Tom Keane	Kevin Wood	Tim Pike	200	4	Yes	…
Tom Keane	Kevin Wood	Tim Pike	94511	2	No	…
6 Tom Keane	Kevin Wood	Tim Pike	3 421	2	Yes	…

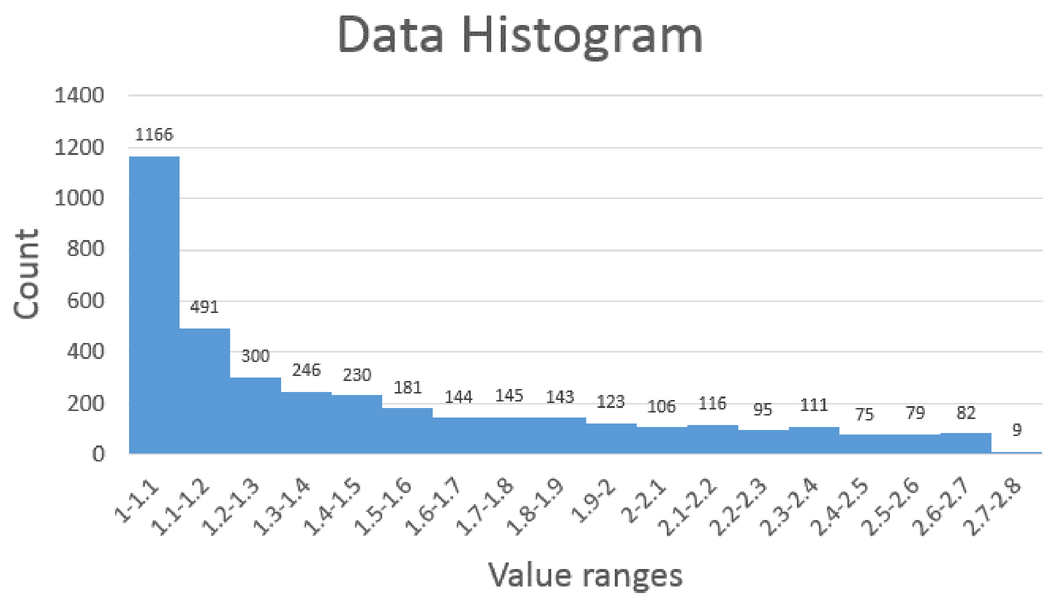
String Integer "Nominal" Unstructured

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
1	0.0063	18	2.3100	0	0.5380	6.5750	65.2000	4.0900	1	296	15.3000	396.9000	4.9800	24
2	0.0273	0	7.0700	0	0.4690	6.4210	78.9000	4.9671	2	242	17.8000	396.9000	9.1400	21.6000
3	0.0273	0	7.0700	0	0.4690	7.1850	61.1000	4.9671	2	242	17.8000	392.8300	4.0300	34.7000
4	0.0324	0	2.1800	0	0.4580	6.9980	45.8000	6.0622	3	222	18.7000	394.6300	2.9400	33.4000
5	0.0691	0	2.1800	0	0.4580	7.1470	54.2000	6.0622	3	222	18.7000	396.9000	5.3300	36.2000
6	0.0299	0	2.1800	0	0.4580	6.4300	58.7000	6.0622	3	222	18.7000	394.1200	5.2100	28.7000
7	0.0883	12.5000	7.8700	0	0.5240	6.0120	66.6000	5.5605	5	311	15.2000	395.6000	12.4300	22.9000
8	0.1446	12.5000	7.8700	0	0.5240	6.1720	96.1000	5.9505	5	311	15.2000	396.9000	19.1500	27.1000
9	0.2112	12.5000	7.8700	0	0.5240	5.6310	100	6.0821	5	311	15.2000	386.6300	29.9300	16.5000
10	0.1700	12.5000	7.8700	0	0.5240	6.0040	85.9000	6.5921	5	311	15.2000	386.7100	17.1000	18.9000
11	0.2249	12.5000	7.8700	0	0.5240	6.3770	94.3000	6.3467	5	311	15.2000	392.5200	20.4500	15
12	0.1175	12.5000	7.8700	0	0.5240	6.0090	82.9000	6.2267	5	311	15.2000	396.9000	13.2700	18.9000
13	0.0938	12.5000	7.8700	0	0.5240	5.8890	39	5.4509	5	311	15.2000	390.5000	15.7100	21.7000

Exploratory Data Analysis (EDA)

- Mean
- Median
- Standard deviation
- Histograms!

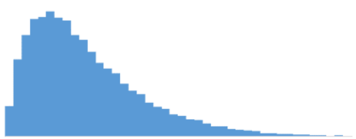
	A	B	C	D	E	F	G
1	0.735647	0.947027	0.854229	0.56088	0.273142	0.216756	0.79361
2	0.256996	0.794376	0.803345	0.128412	0.181848	0.113902	0.7303
3	0.644927	0.187543	0.959562	0.539821	0.040331	0.560651	0.4815
4	0.93258	0.467512	0.428021	0.986173	0.277735	0.600648	0.87051
5	0.228775	0.194223	0.380177	0.959407	0.202019	0.453636	0.7032
6	0.097481	0.09452	0.539209	0.366889	0.304026	0.923372	0.6992
7	0.928041	0.319983	0.99566	0.091048	0.839732	0.182044	0.0843
8	0.337074	0.997596	0.056519	0.811722	0.260549	0.774011	0.10441
9	0.899714	0.744684	0.995986	0.523544	0.387805	0.956102	0.9608
10	0.386956	0.312822	0.808444	0.467208	0.80197	0.930899	0.3256
11	0.219273	0.801165	0.111613	0.960393	0.313174	0.875519	0.3249
12	0.211368	0.831228	0.624857	0.506879	0.898247	0.830768	0.0786
13	0.210396	0.319881	0.320067	0.197561	0.868724	0.494441	0.4882
14	0.333875	0.460648	0.746342	0.368991	0.432182	0.056148	0.6036
15	0.477373	0.608657	0.75547	0.390956	0.397275	0.135327	0.2649
16	0.003593	0.308439	0.077365	0.624121	0.381396	0.41185	0.4495
17	0.967295	0.840931	0.148907	0.80862	0.028289	0.687918	0.0082
18	0.550282	0.652772	0.273055	0.912683	0.12853	0.072454	0.2460
19	0.389764	0.090453	0.351323	0.524136	0.845297	0.581504	0.8267
20	0.802131	0.307985	0.07222	0.550246	0.957613	0.67176	0.3137
21	0.61533	0.485001	0.686292	0.053164	0.704459	0.925033	0.2047
22	0.622564	0.739001	0.314398	0.456529	0.608796	0.232682	0.66591
23	0.520361	0.413769	0.777187	0.559793	0.775996	0.832615	0.7439
24	0.427441	0.616882	0.152537	0.939188	0.391867	0.888638	0.4355
25	0.690159	0.343905	0.460285	0.840465	0.196179	0.571635	0.0765
26	0.74931	0.899702	0.056719	0.19558	0.031112	0.340661	0.7560
27	0.469696	0.216476	0.580191	0.848264	0.85582	0.720294	0.3610
28	0.865221	0.690048	0.535996	0.968247	0.367861	0.122153	0.4477



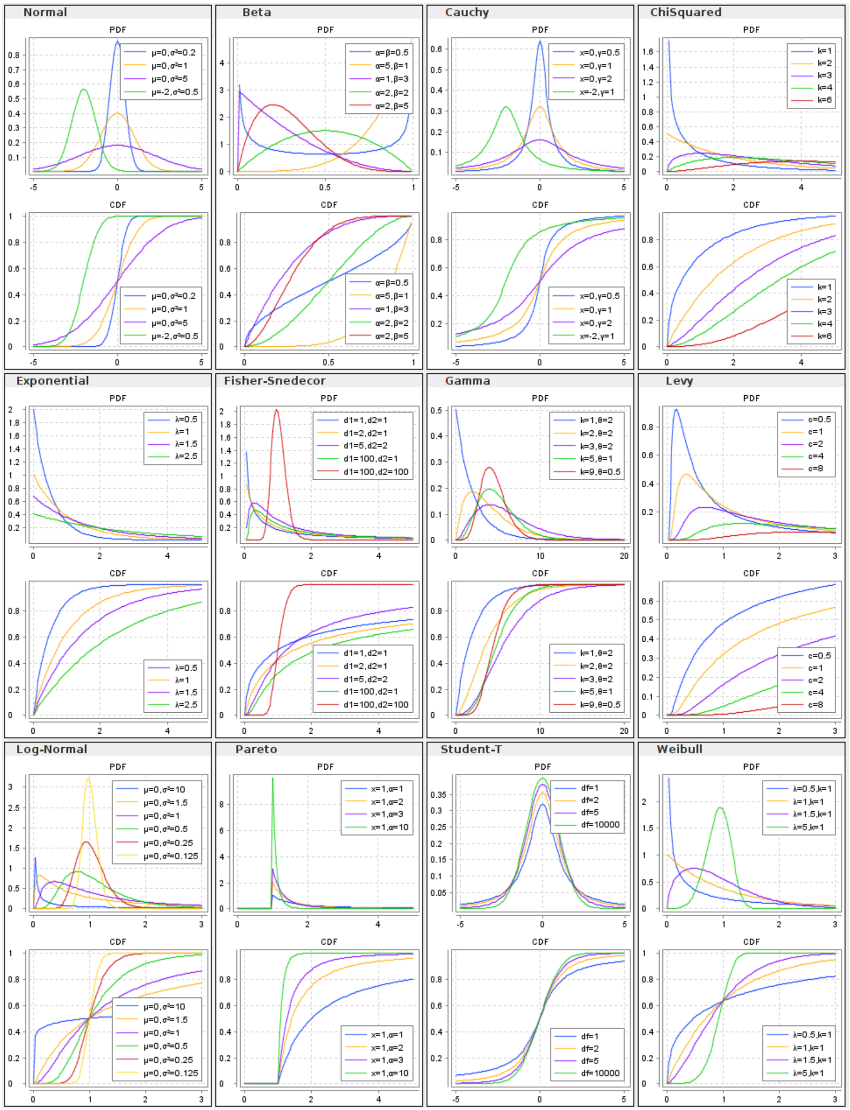
Distributions

- The majority of data will follow SOME distribution
  - o Weight of all Americans: *Gaussian*

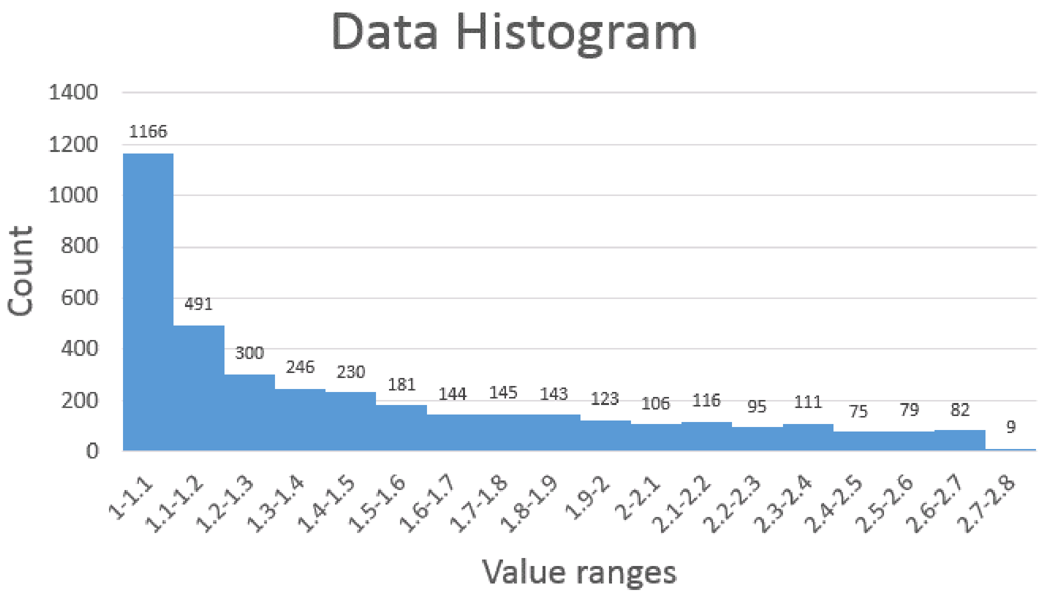
  - o phone call length: *Exponential*

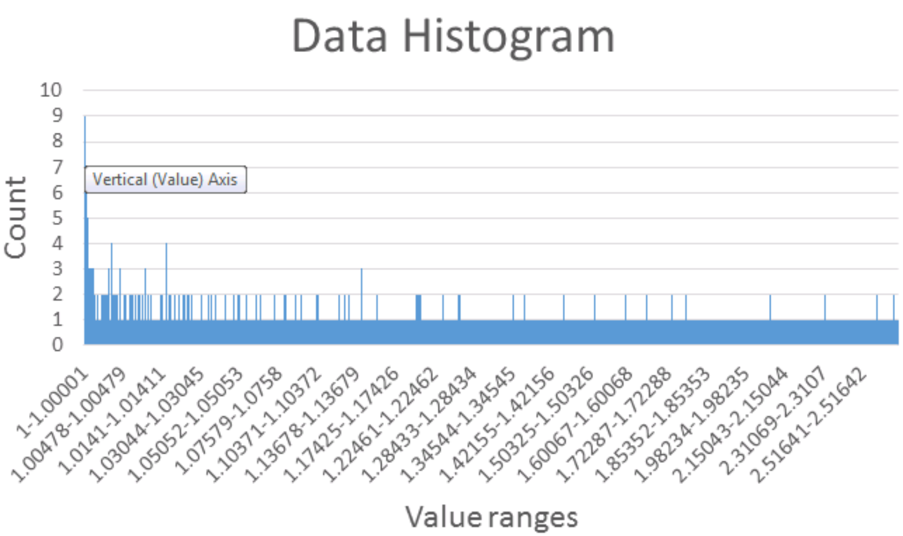


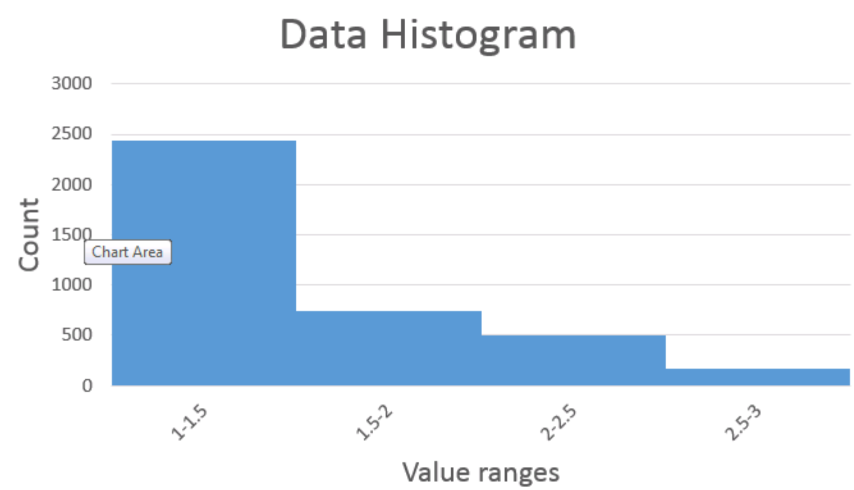
- Determining distribution is a common Data Science task
- Multidimensional outliers: Insider Threat example

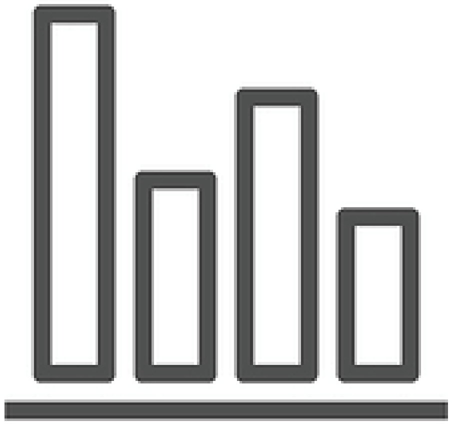


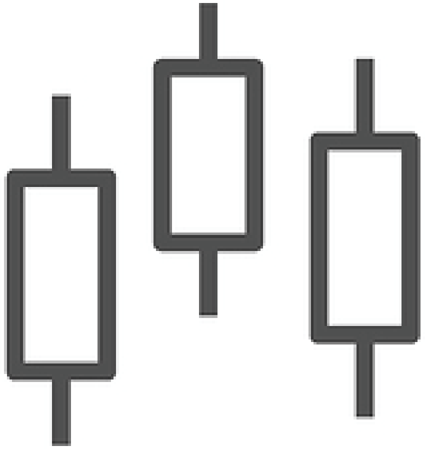
EDA – *Smart visualizations*

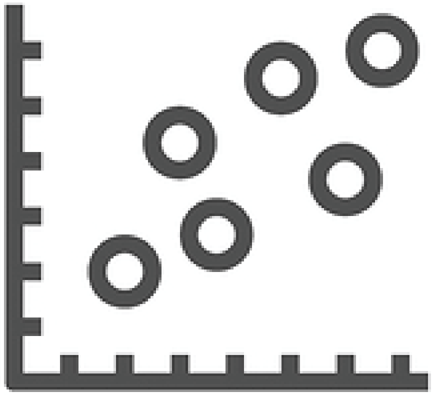


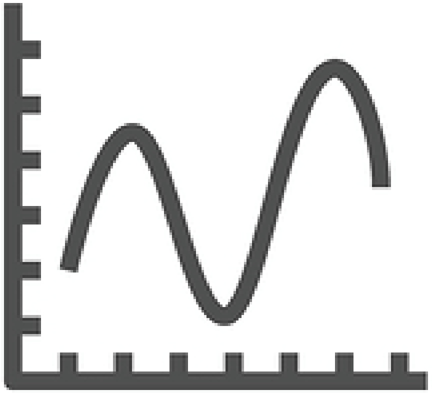


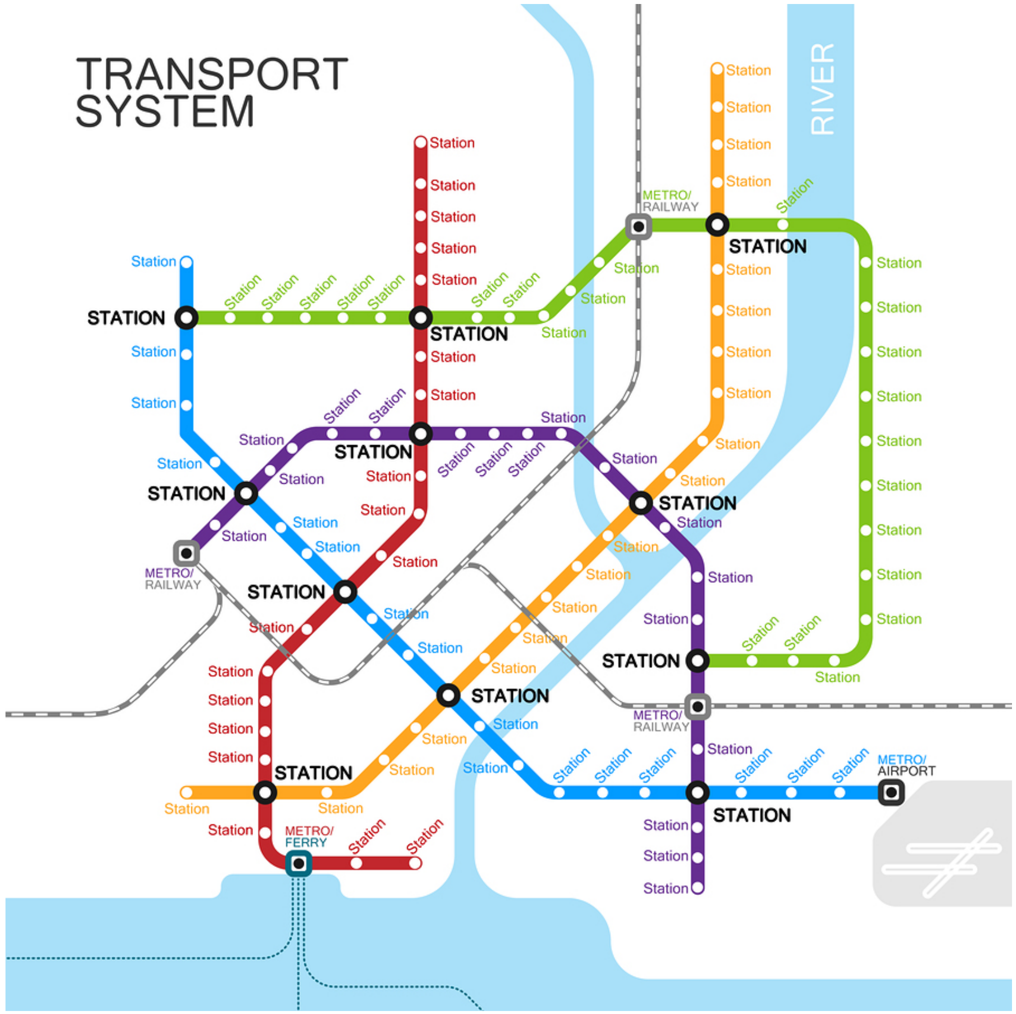


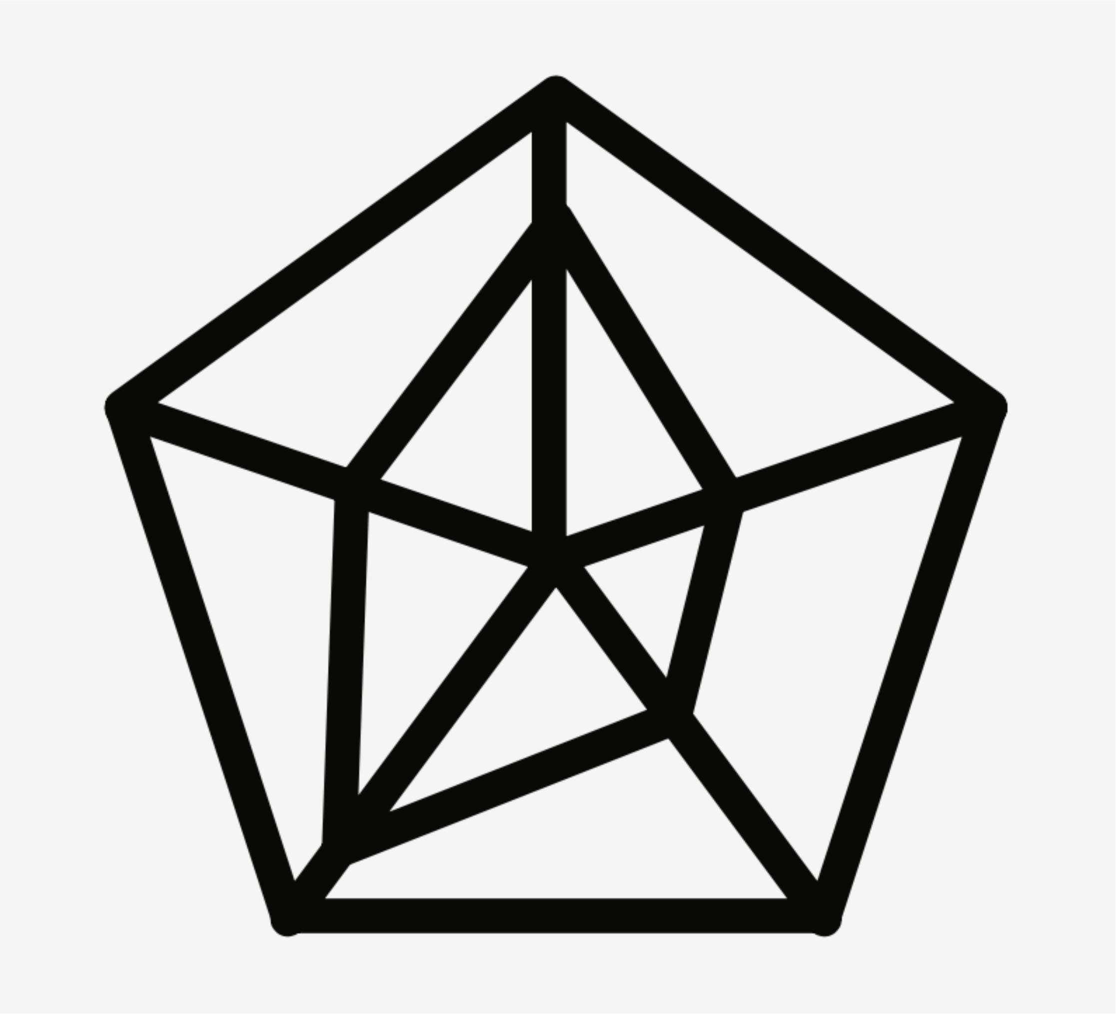









TRANSPORT
SYSTEM
RIVER
STATION
Station
METRO/
RAILWAY
METRO/
AIRPORT
METRO/
FERRY



Brief interruption

Lovely visuals and all, but THIS isn't data science! Where's the fancy predicting the future and whatnot?

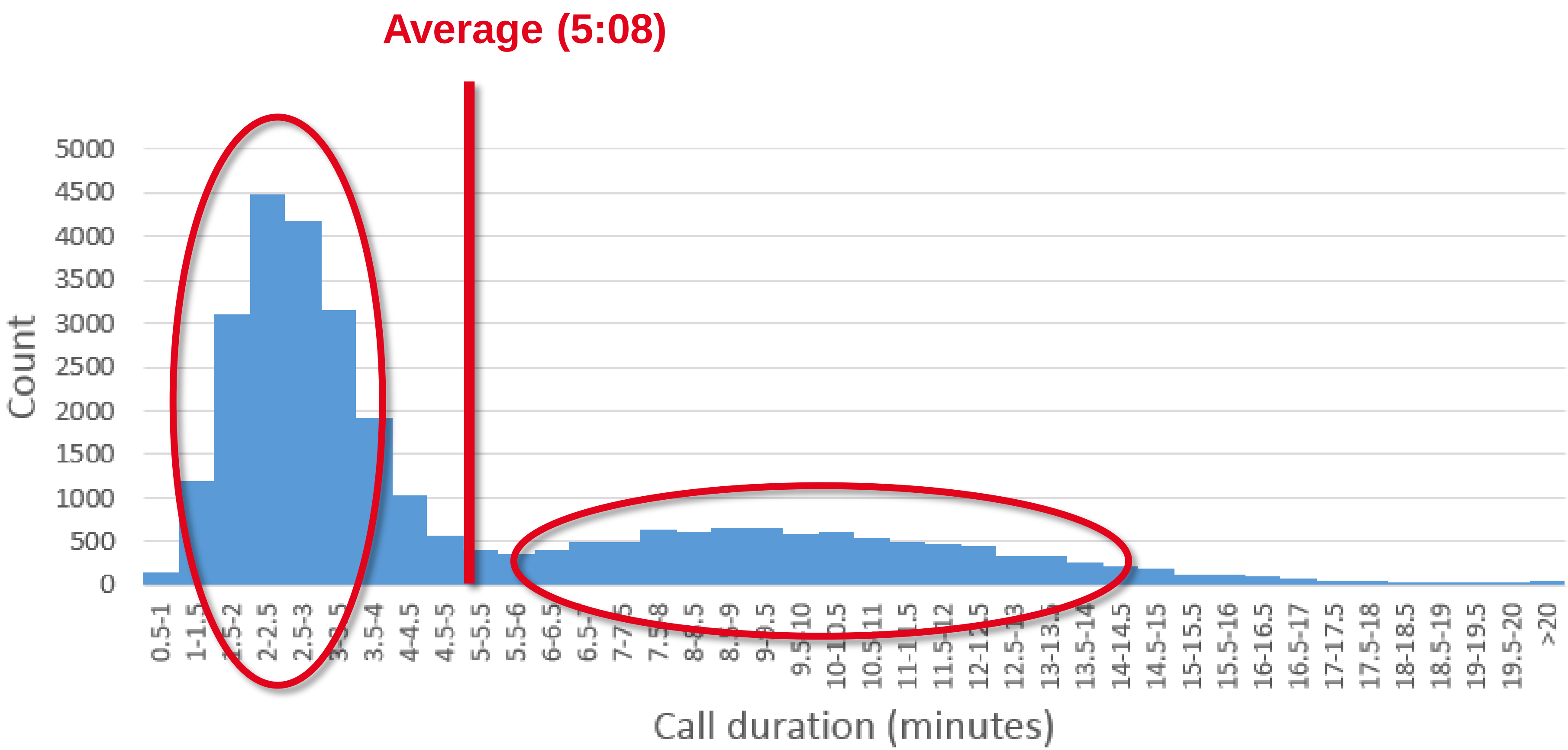
Skeptics in the audience

Brief interruption

Data Science helps you use data to get results.

This is it.

Call center manager – *call duration histogram*



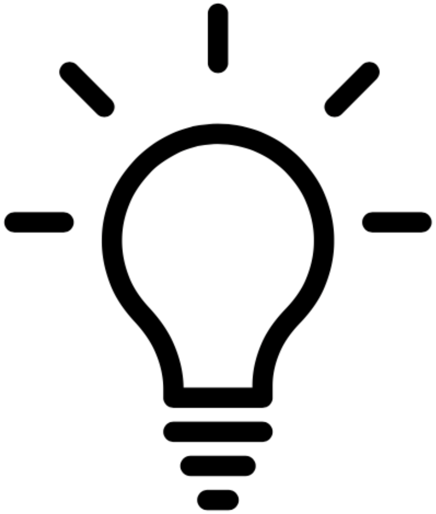
Call Center manager – *Insights!*

Strategy update:

- Goodbye “reduce call time”
- Hello “reduce callbacks”

How to measure?

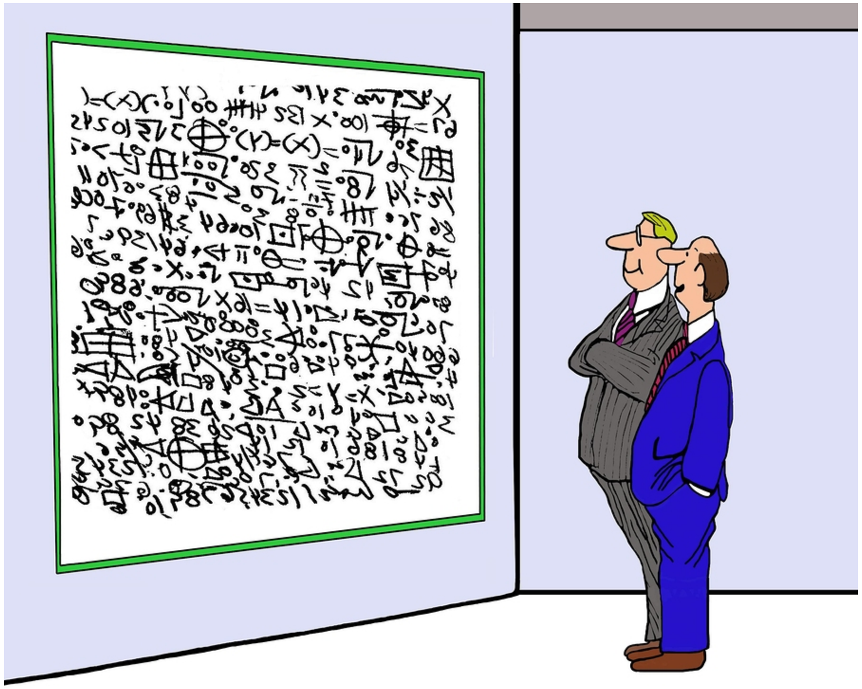
“callbacks” isn’t currently captured



Feature Engineering

Need more useful data?

Create it yourself!



"When you put it like that, it makes complete sense."

Feature Engineering

- Feature Engineering: coming up with new, useful (i.e., informative) data
  - mean, sums, medians, etc.
  - x^2, xy, sqrt(xy), etc.
- Our case:
  - # of callbacks
  - Call during peak time?
  - Overall agent performance? (combination of factors)

The role of Listening in Data Science

Data science finds hidden patterns in data

Experts know what data & patterns are important

Talk to subject matter experts

Call Center manager – *Predictive analytics*

Can we predict staffing levels…

- …one day ahead?
- …one week ahead?
- …one month ahead?

Can we determine what types of calls to expect…

- …for a product we haven't had before?
- …for a market we've never seen before?

Example Predictive Analytics Questions

Predicting Current Unknowns

Online:	Which ads are malicious?
Security:	Is the bank transaction fraudulent?
IC:	Which names map to the same person (entity resolution)?

Predicting Future Events

Retail:	What will be the new trend of merchandise that a company should stock?
Security:	Where will a hacker next attack our network?
IC:	Who will become the next insider threat?

Determining Future Actions

Sales:	How can a company increase sales revenues?
Health:	What actions can be taken to prevent the spread of flu?
IC:	How will a vulnerability patch affect our knowledge/preparedness for future attacks?

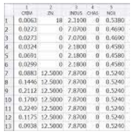
Call Center manager – *Predictive analytics*

Many techniques available, explored in next section

Call Center manager – *Review*

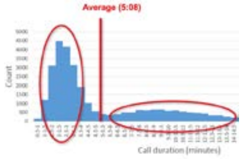
Get data

Clean data



EDA, Visualization

Interpretation



Action!

Prediction

Because we know our data, we can ask…

- …more intelligent questions
- …action-oriented questions
- …questions that can be answered

This slide intentionally left blank

The master carpenter

"The right tool for the job"

Feature Engineering – *Part 2*

"With the wrong wood, I can make nothing"

The fuel of data science is data

Data preparation is critical

Data quality $\gg$ algorithm choice

That will come up…

Types of Machine Learning Algorithms

Classification

- Naïve Bayes
- Logistic Regression
- Decision Trees
- K-Nearest Neighbors
- Support Vector Machines

Regression

- Linear Regression
- Support Vector Machines

Clustering

- K-Means Clustering

Types of Machine Learning Algorithms

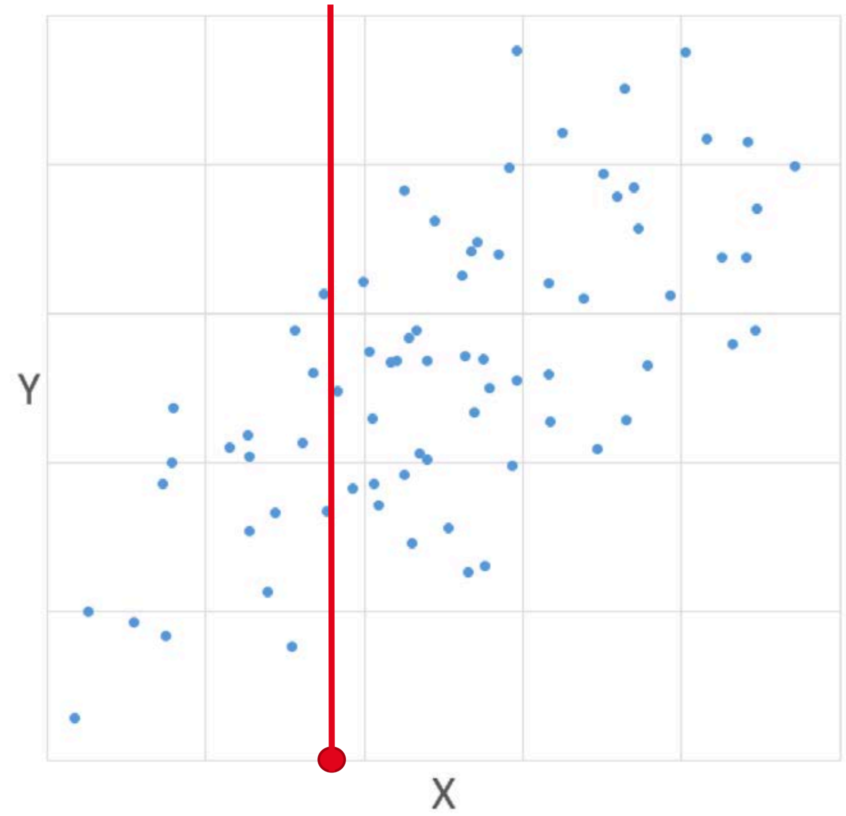
Applications: Everywhere

- Banking
- Weather
- Sports scores
- Economics
- Environmental science
- Cybersecurity

Linear Regression – *Prediction*

Problem:

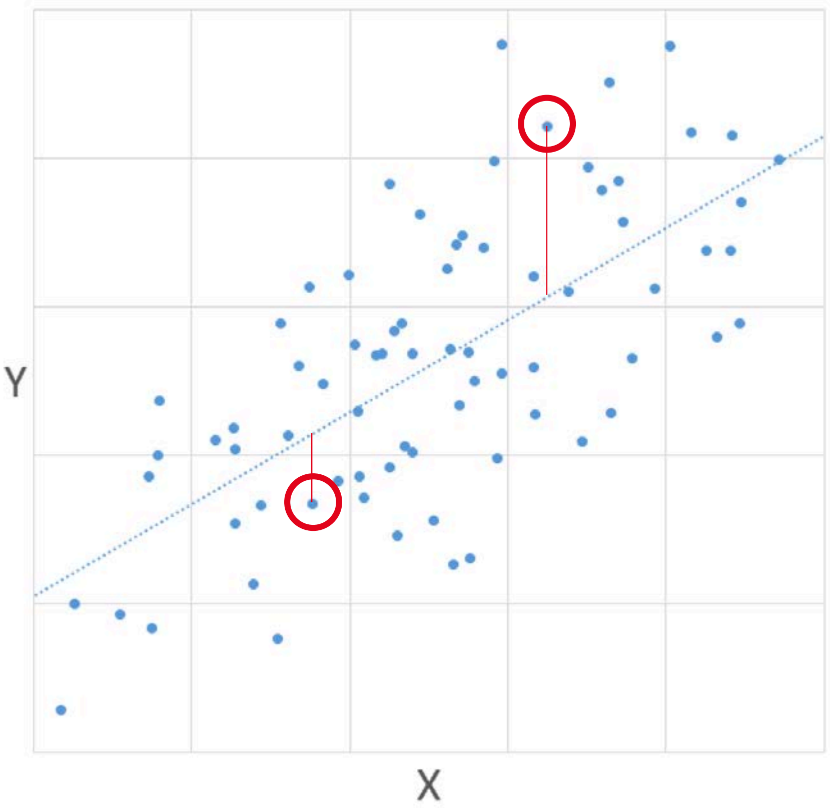
If I have examples of X and Y, when I learn a new X, can I predict Y?



Linear Regression – *Prediction*

Solution: Find the line that is closest to every point

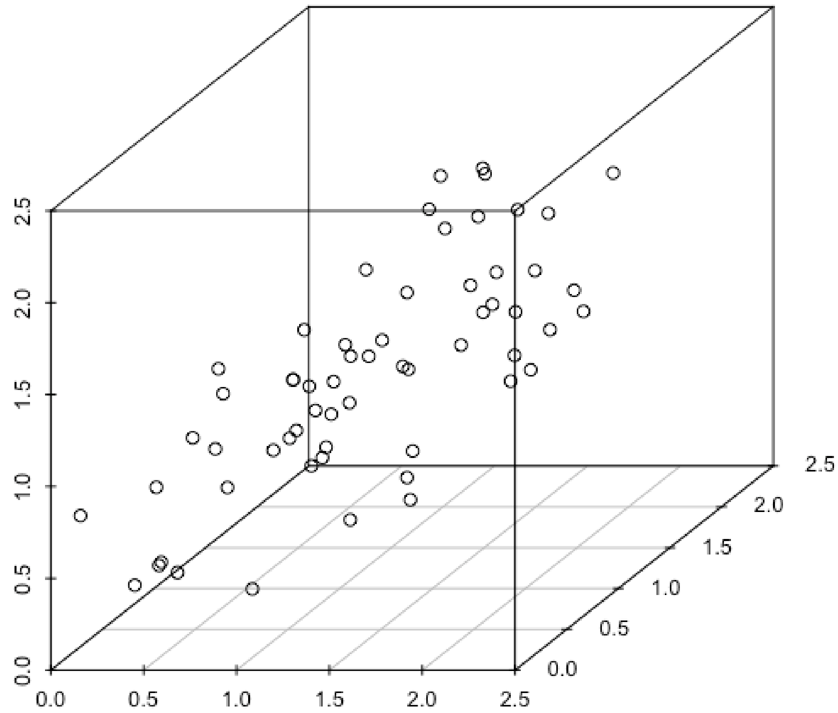
Said differently: Find the line that the SUM of all errors is smallest



Linear Regression – *Prediction*

Three dimensions,
same concept

HUNDREDS of dimensions,
same concept



Linear Regression

Very widely used

- Simple to implement
- Quick to run
- Easy to interpret
- Works for many problems
- First identified in early 1800's; very well studied

When applicable:

- Works best with numeric data (usually)
- Works for predicting specific numeric outcome

Logistic Regression – *Classification*

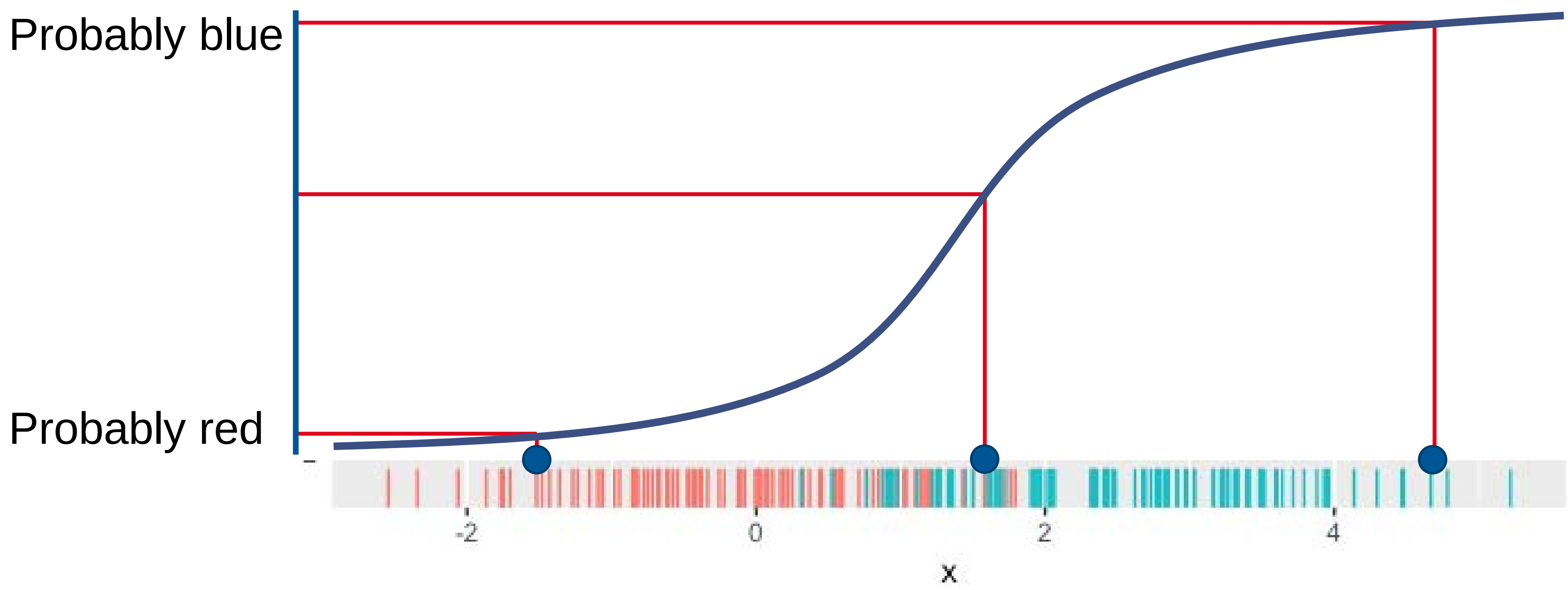
Idea: Classification using a *discriminative* model

- Predict future behavior based on existing labeled data
- Draws a line to assign labels

Mainly used for binary classification: either “red” or “blue”

Logistic Regression – *Classification*

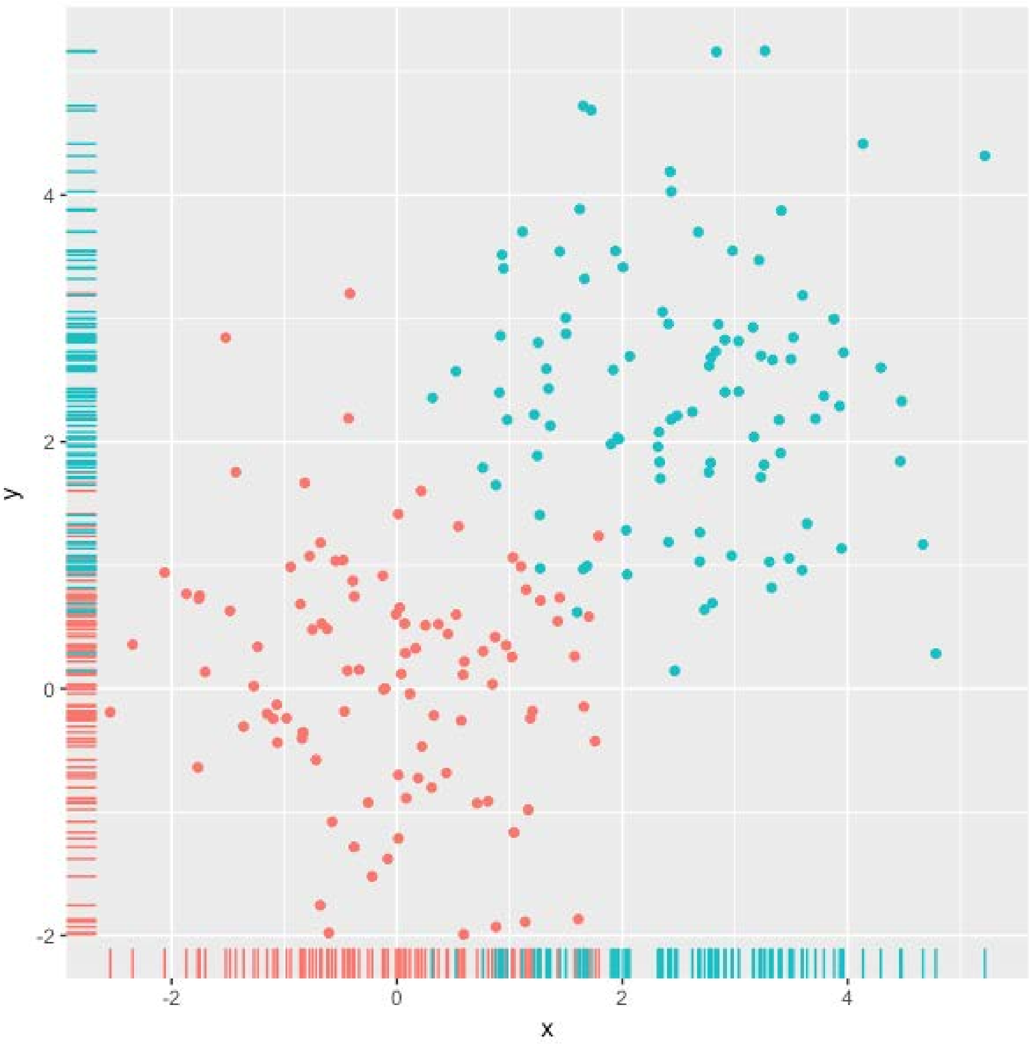
Look at *distribution*, what's likely based on current data



Logistic Regression

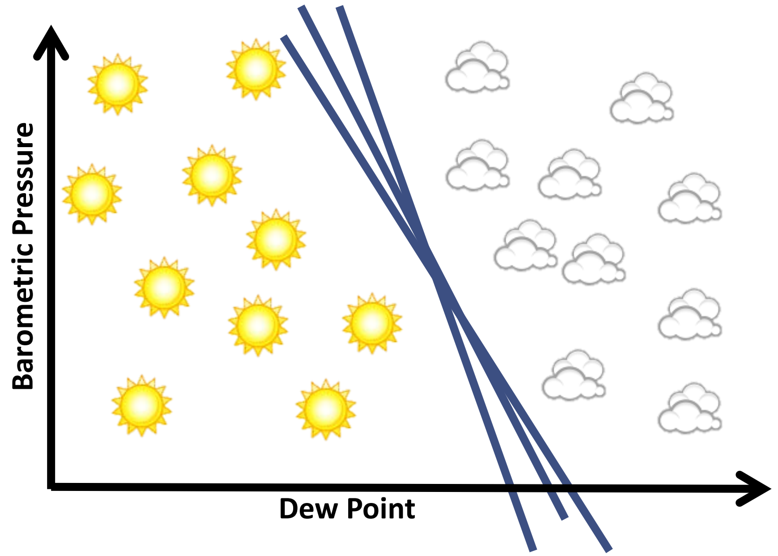
Three dimensions, same concept

HUNDREDS of dimensions, same concept



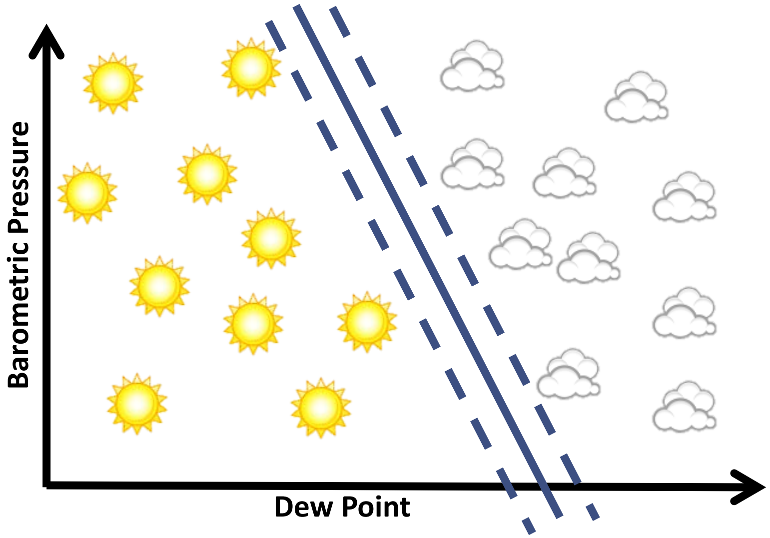
Classification: Support Vector Machine

Idea: The optimal classifier is the one that is the farthest from both classes



Classification: Support Vector Machine

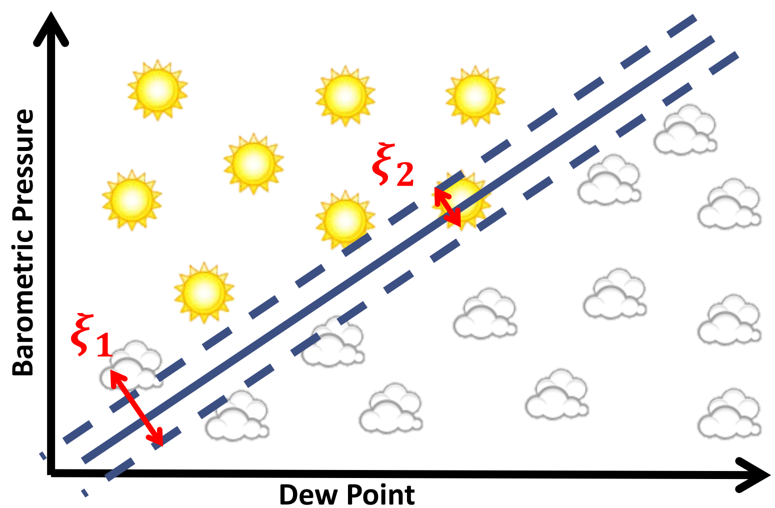
Idea: The optimal classifier is the one that is the farthest from both classes



Classification: Support Vector Machine

Algorithm:

- Find lines like before
- Assign a cost to misclassified data points based on distance from the classification line

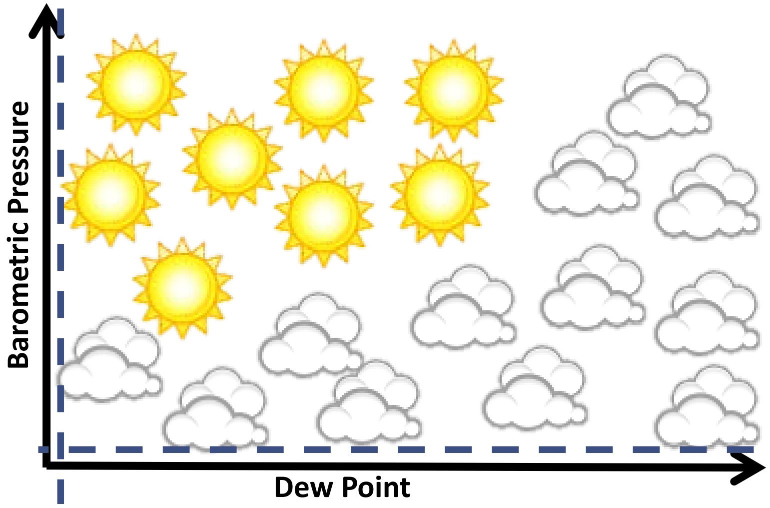


Classification: Decision Trees

Idea: Instead of drawing a single complicated line through the data, draw many simpler lines.

Algorithm:

- Scan through all values of all features to find the one that “helps the most” to determine what data gets what label.
- Divide the data based on that value, and then repeat recursively on each part.

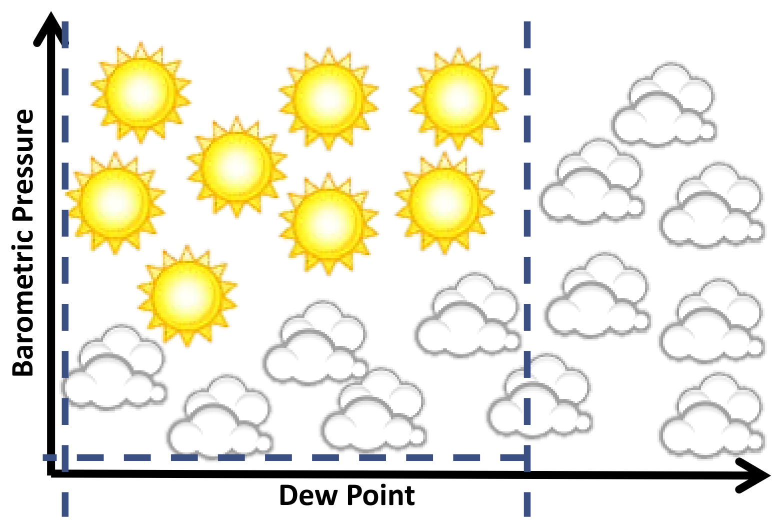


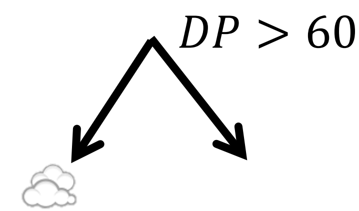
Classification: Decision Trees

Idea: Instead of drawing a single complicated line through the data, draw many simpler lines.

Algorithm:

- Scan through all values of all features to find the one that “helps the most” to determine what data gets what label.
- Divide the data based on that value, and then repeat recursively on each part.



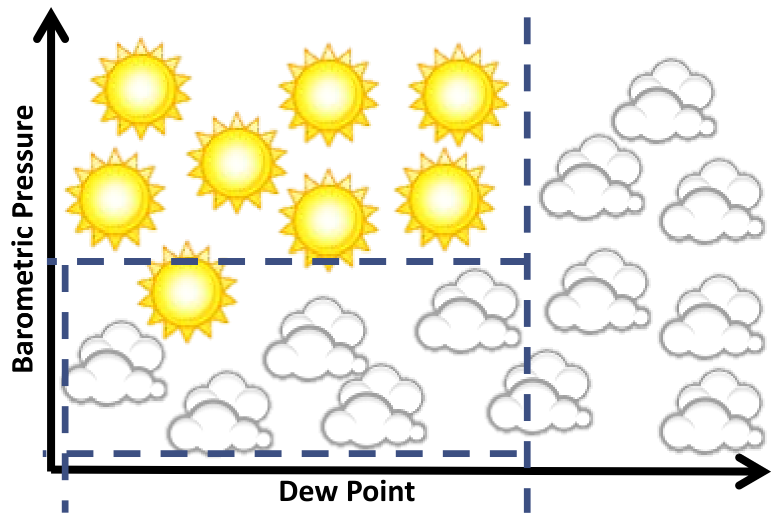


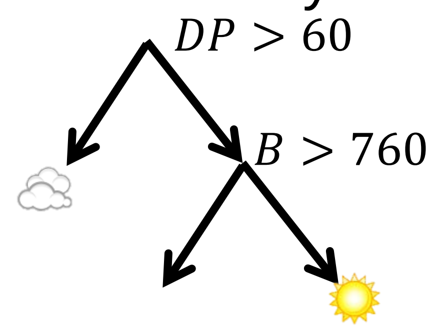
Classification: Decision Trees

Idea: Instead of drawing a single complicated line through the data, draw many simpler lines.

Algorithm:

- Scan through all values of all features to find the one that “helps the most” to determine what data gets what label (“information gain”).
- Divide the data based on that value, and then repeat recursively on each part.





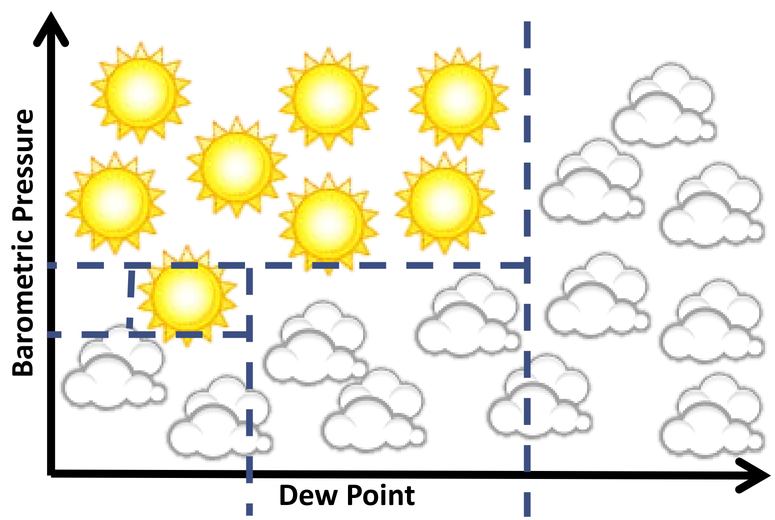
Classification: Decision Trees

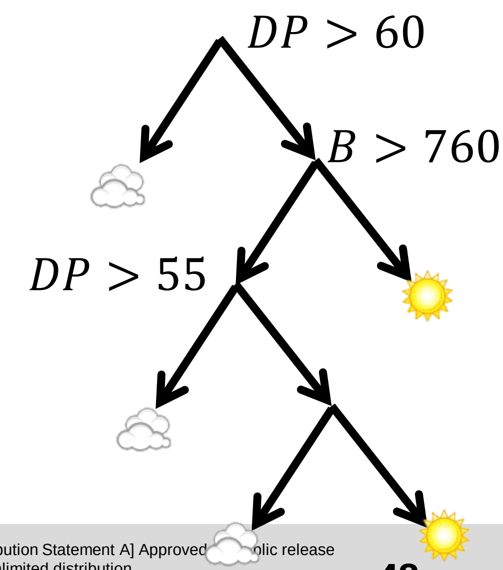
Benefits:

- Works well when small.
- Very easy to understand!

Challenges:

- Trees overfit easily
- Very sensitive to data; Random Forests



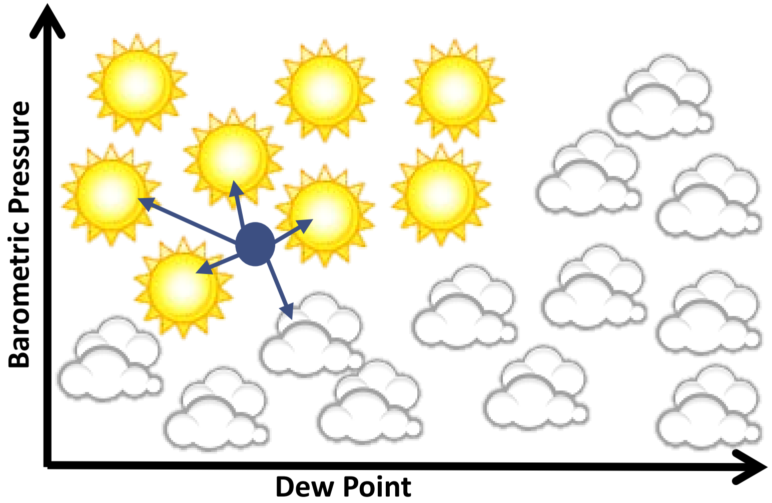


Classification: K-Nearest Neighbors

Idea: A new point is likely to share the same label as points around it.

Algorithm:

- Pick constant k as number of neighbors to look at.
- For each new point, vote on new label using the k neighbor labels.

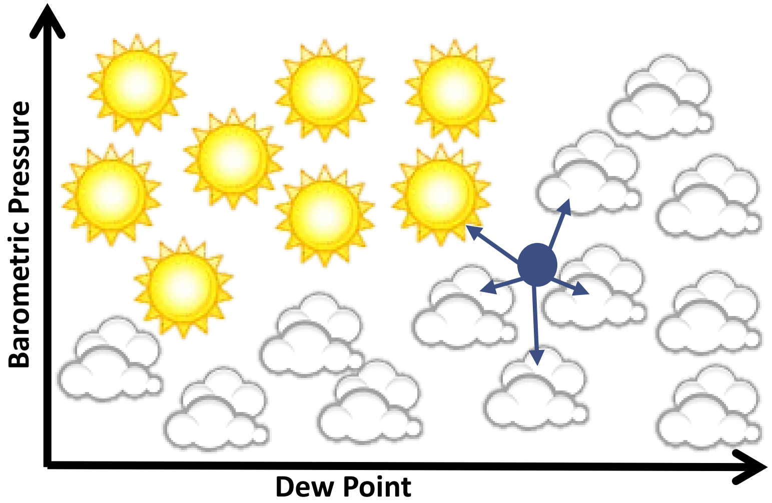


Classification: K-Nearest Neighbors

Idea: A new point is likely to share the same label as points around it.

Algorithm:

- Pick constant k as number of neighbors to look at.
- For each new point, vote on new label using the k neighbor labels.

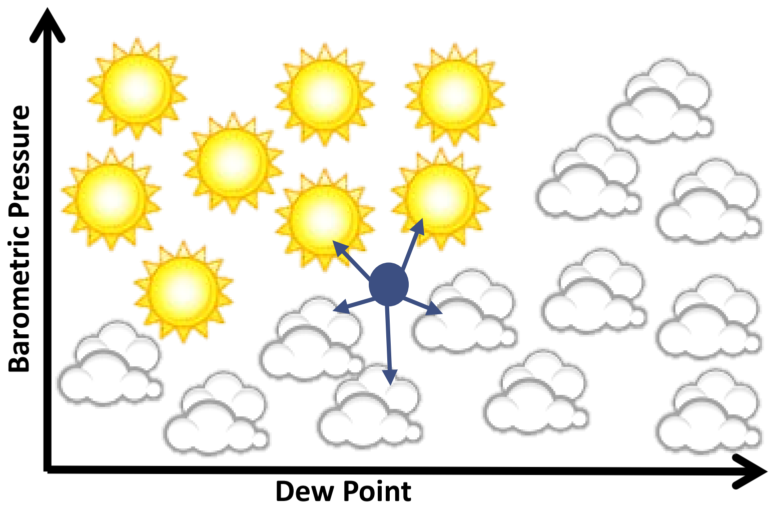


Classification: K-Nearest Neighbors

Idea: A new point is likely to share the same label as points around it.

Algorithm:

- Pick constant k as number of neighbors to look at.
- For each new point, vote on new label using the k neighbor labels.



Classification: K-Nearest Neighbors

Works well when

- there is a good distance metric and weighting function to vote on classification

Challenges:

- Not a smooth classifier; points near each other may get classified differently
- Must search all your data every time you want to classify a new point
- When k is small (1,2,3,4), essentially it is overfitting to the data points

Clustering

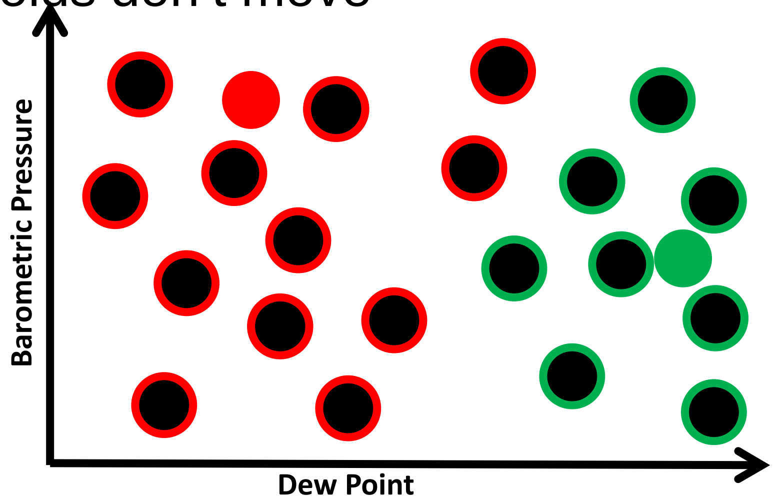
- Unsupervised learning
- Structure of un-labeled data
- Organize records into groups based on some similarity measure
- Cluster is the collection of records which are similar

Clustering: K-means

Idea: Find the clusters by minimizing distances of cluster centers to data.

Algorithm:

- Instantiate k distinct random guesses μ_i of the cluster centers
- Each data point classifies itself as the μ_i it is closest to it
- Each μ_i finds the centroid of the points that were closest to it and jumps there
- Repeat until centroids don't move

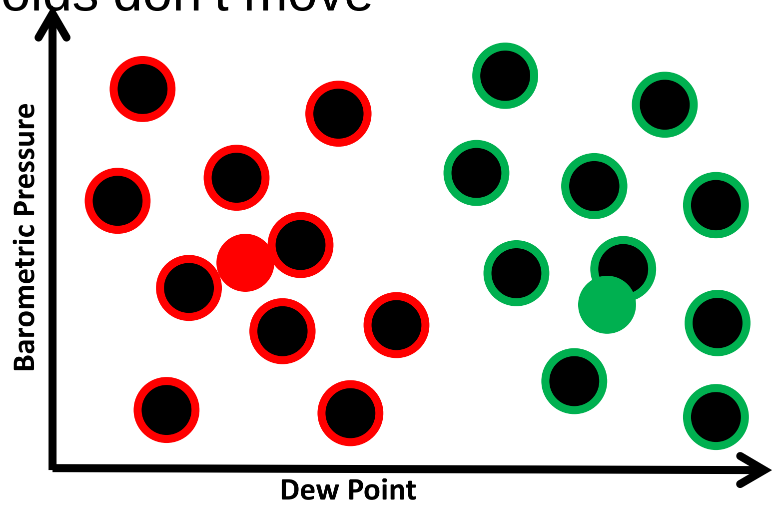


Clustering: K-means

Idea: Find the clusters by minimizing distances of cluster centers to data.

Algorithm:

- Instantiate k distinct random guesses μ_i of the cluster centers
- Each data point classifies itself as the μ_i it is closest to it
- Each μ_i finds the centroid of the points that were closest to it and jumps there
- Repeat until centroids don't move

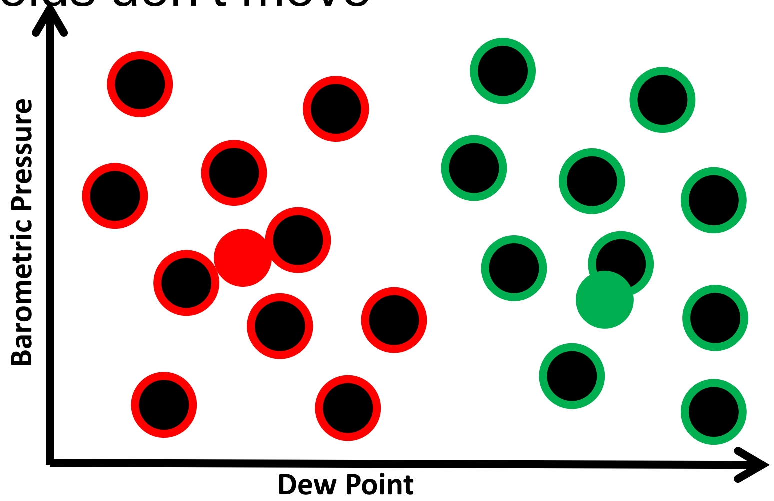


Clustering: K-means

Idea: Find the clusters by minimizing distances of cluster centers to data.

Algorithm:

- Instantiate k distinct random guesses μ_i of the cluster centers
- Each data point classifies itself as the μ_i it is closest to it
- Each μ_i finds the centroid of the points that were closest to it and jumps there
- Repeat until centroids don't move

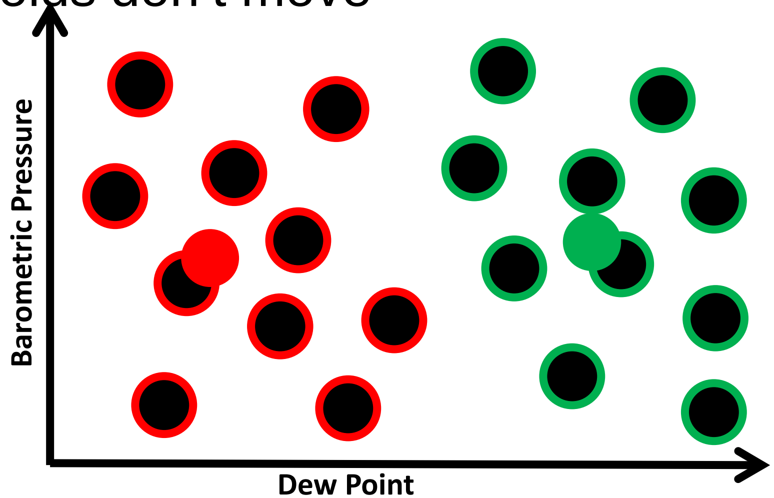


Clustering: K-means

Idea: Find the clusters by minimizing distances of cluster centers to data.

Algorithm:

- Instantiate k distinct random guesses μ_i of the cluster centers
- Each data point classifies itself as the μ_i it is closest to it
- Each μ_i finds the centroid of the points that were closest to it and jumps there
- Repeat until centroids don't move

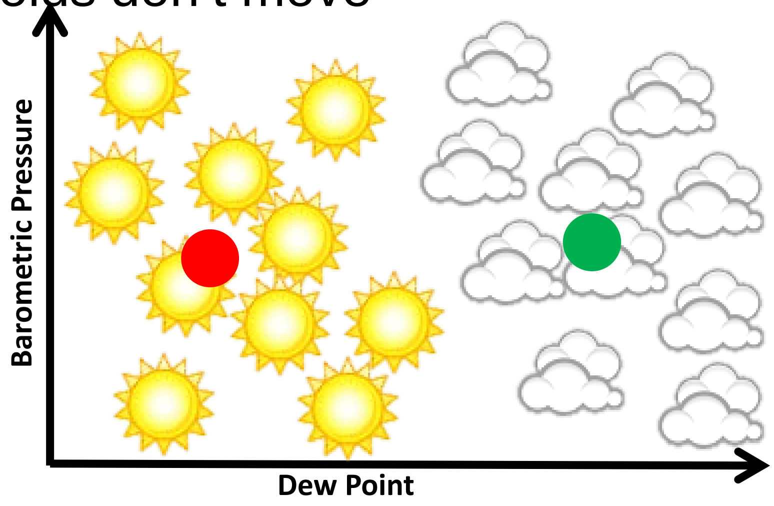


Clustering: K-means

Idea: Find the clusters by minimizing distances of cluster centers to data.

Algorithm:

- Instantiate k random guesses μ_i of the clusters
- Each data point classifies itself as the μ_i it is closest to it
- Each μ_i finds the centroid of the points that were closest to it and jumps there
- Repeat until centroids don't move



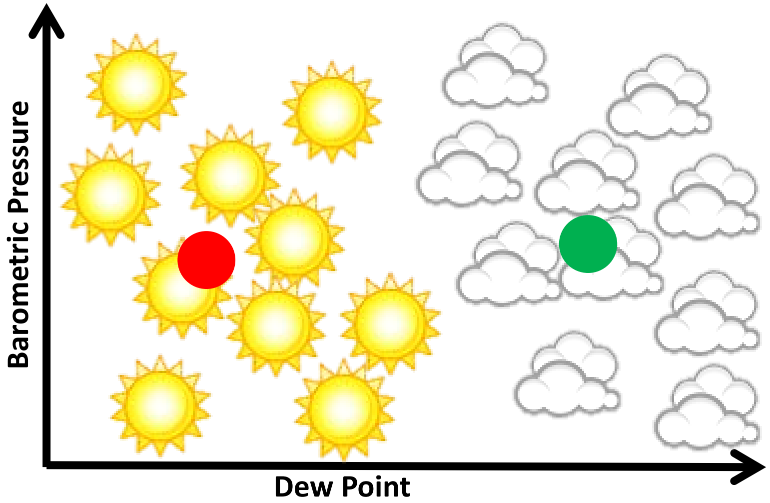
Clustering: K-means

Works well when

- there is a good distance metric between the points
- the number of clusters is known in advance

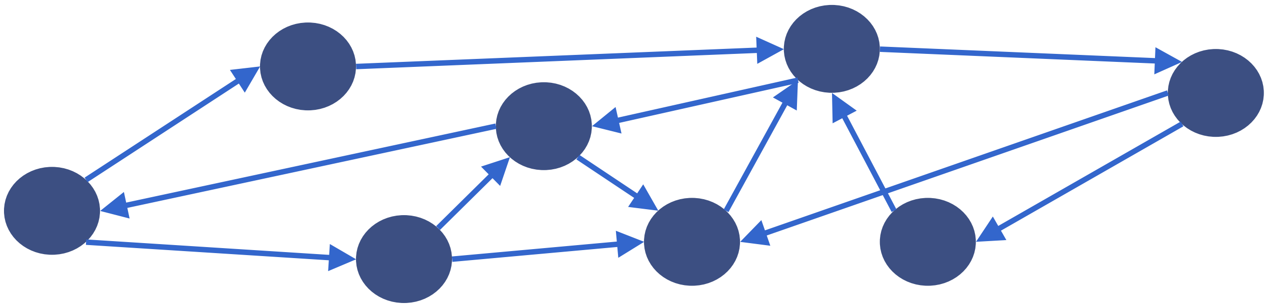
Challenges:

- Clusters that overlap or are not separable are difficult to cluster correctly.



Influencers

Goal: Detect the people who control or distribute information through a network.

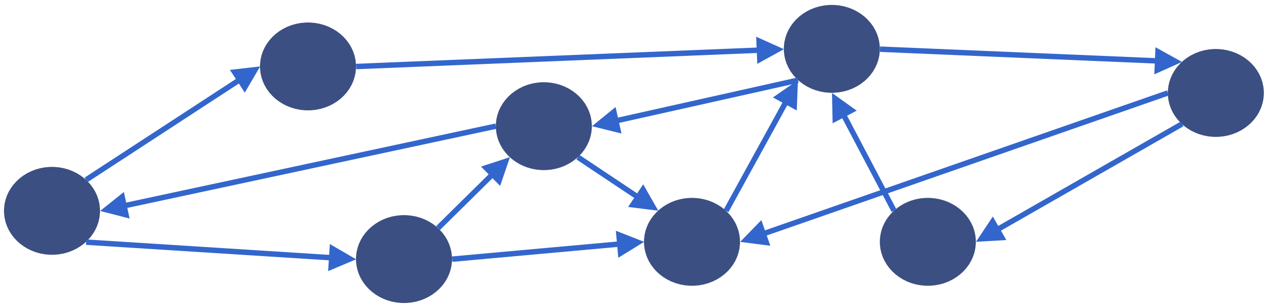


Influencers: Degree Centrality

Idea: Influential people have a lot of people watching them.

Equation

- Degree centrality = number of directed edges to the node
  - High degree centrality people are those with large numbers of followers.
- If undirected graph, transform to bi-directional and compute

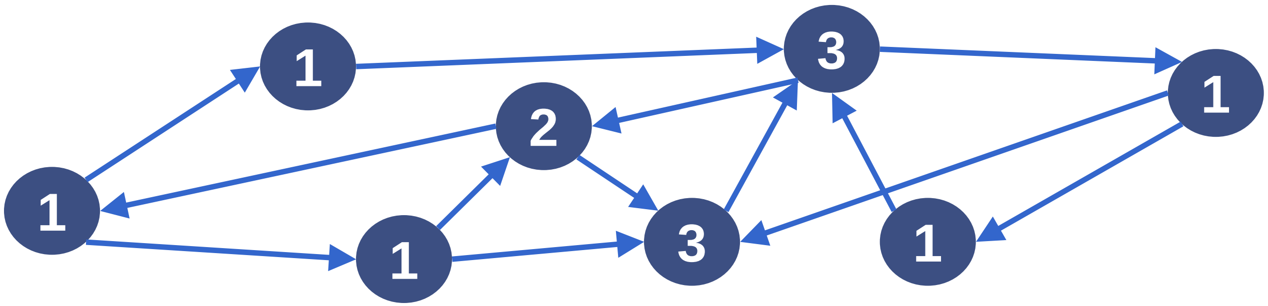


Influencers: Degree Centrality

Idea: Influential people have a lot of people watching them.

Equation

- Degree centrality = number of directed edges to the node
  - High degree centrality people are those with large numbers of followers.
- If undirected graph, transform to bi-directional and compute

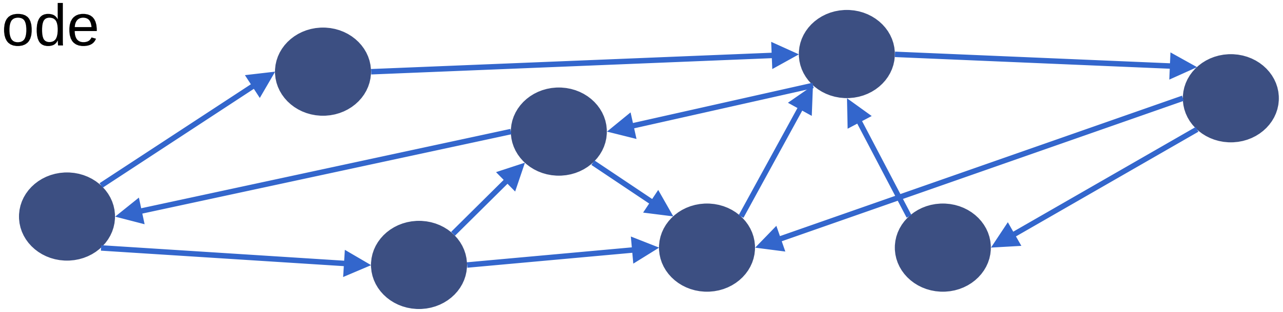


Influencers: Betweenness Centrality

Idea: Influential people are "information brokers" who connect different groups of people.

Algorithm

- Find all shortest paths from all nodes to all other nodes in the graph.
- Betweenness centrality for a node = sum over all start and end nodes of the number of shortest paths in the graph that include the node

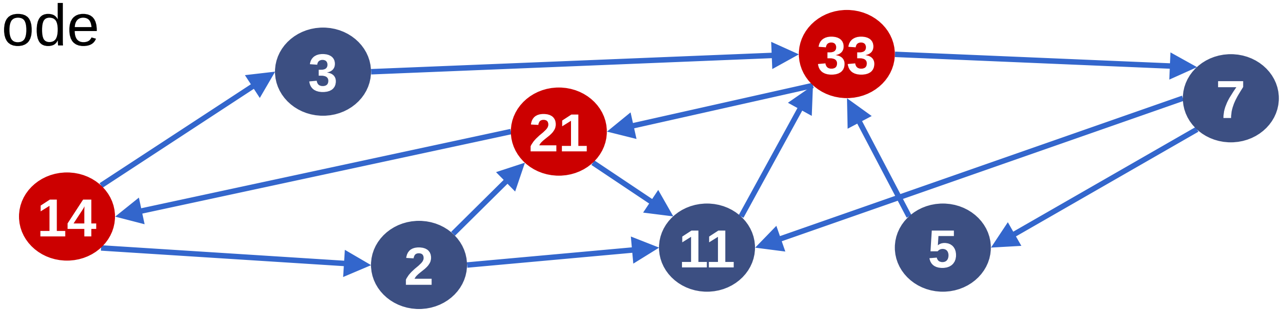


Influencers: Betweenness Centrality

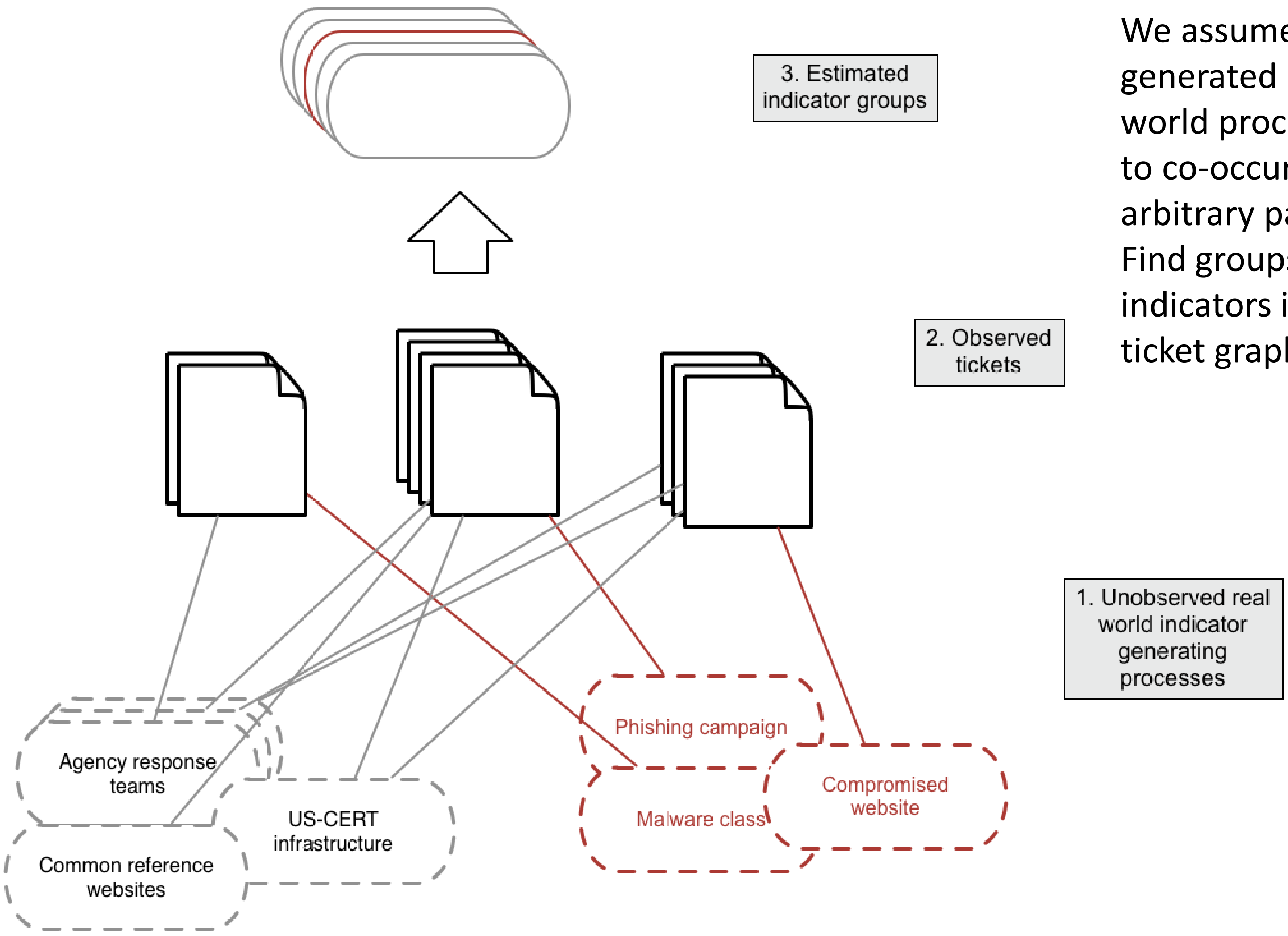
Idea: Influential people are “information brokers” who connect different groups of people.

Algorithm

- Find all shortest paths from all nodes to all other nodes in the graph.
- Betweenness centrality for a node = sum over all start and end nodes of the number of shortest paths in the graph that include the node



Indicator communities

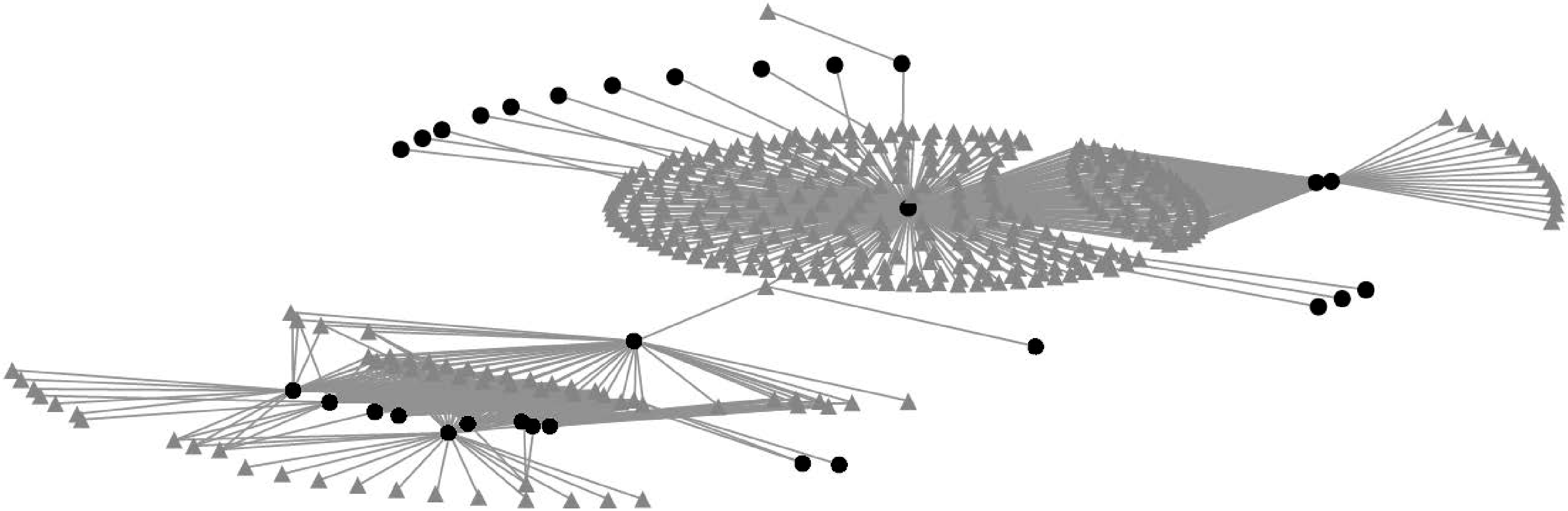


But what if we aren't starting with a reference indicator?

We assume that indicators generated by a coherent real world process will be more likely to co-occur in tickets than arbitrary pairs of indicators.

Find groups of highly similar indicators in complete indicator-ticket graph.

Indicator-ticket graph



A subset of the ticket-indicator graph
(for a small set of selected indicators)

- Tickets are grey triangles
- Indicators are black circles
- Edges connect tickets to the indicators they contain

Machine Learning Is Growing

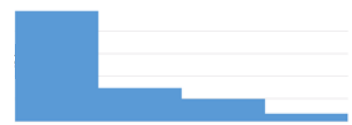
Preferred approach for many problems

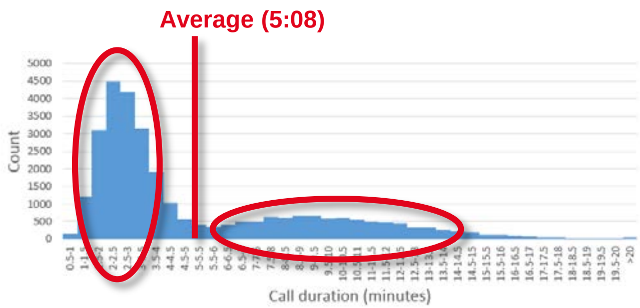
- Speech recognition
- Natural language processing
- Medical diagnosis
- Robot control
- Sensor networks
- Computer vision
- Weather prediction
- Social network analysis
- AlphaGO, Watson Jeopardy!

This slide also intentionally left blank, just like the earlier one

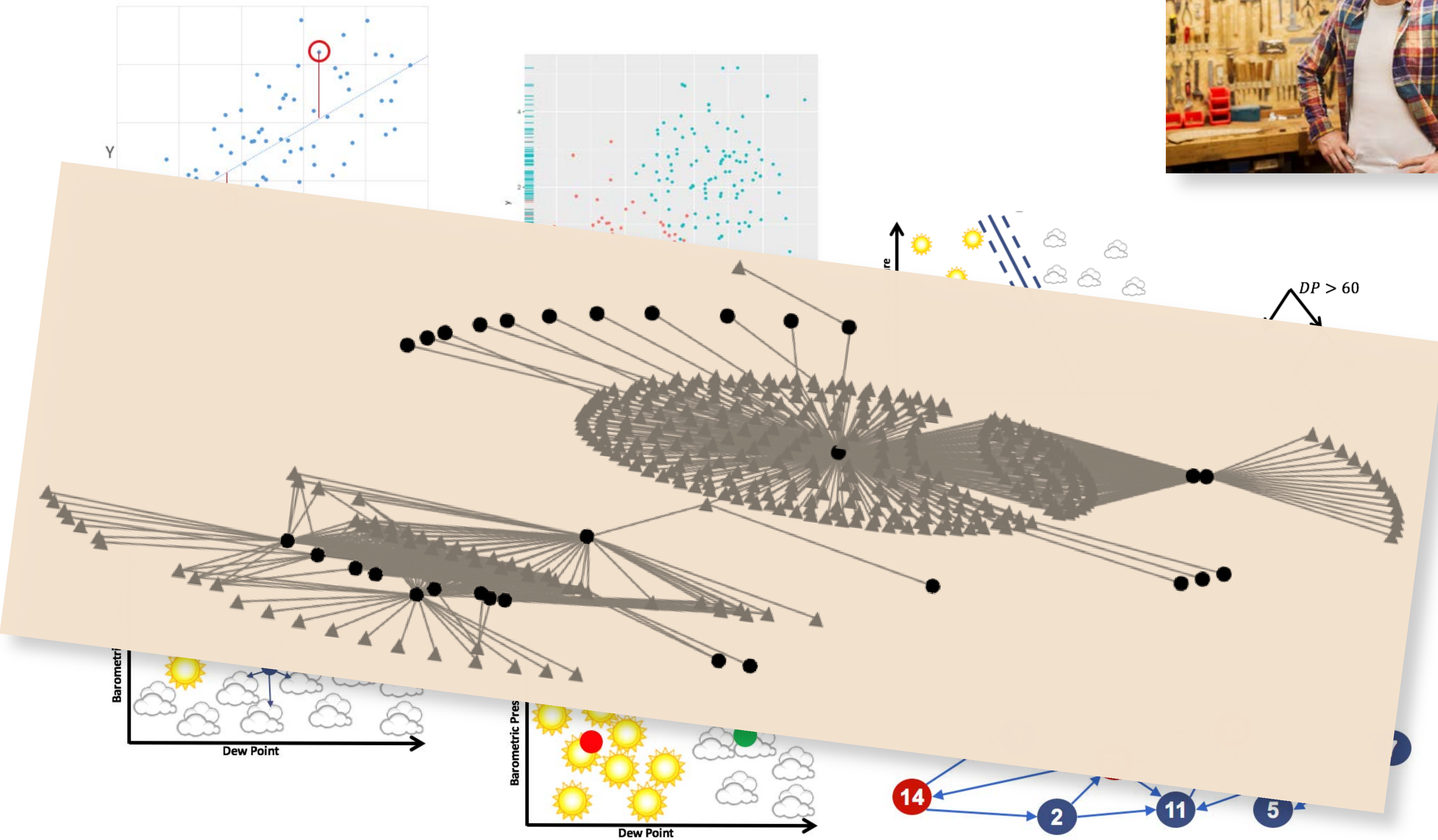
What we did today

Name	Mgr	Dir	Length	Line	Solved?	Comment
Beth Jones	Dan Thomas	Anne Kim	1:30	1	Y	...
Beth Jones	Dan Thomas	Anne Kim	1:52	3	Y	...
Jones, Beth	Dan Thomas	Anne Kim	90	2	Y	...
Tom Keane	Mark Ryan	Tim Pike	88	2	N	...





What we did today



Data Science helps you use data to get results.

Eliezer Kanal
Technical Manager
(412) 268-5204
ekanal@sei.cmu.edu

Daniel DeCapria
Data Scientist
(412) 268-2457
djdecapria@sei.cmu.edu