Splunk for Analytics and Data Science

Document Usage Guidelines

- Should be used only for enrolled students
- Not meant to be a self-paced document, an instructor is needed
- Do not distribute

Course Prerequisites

- Splunk Fundamentals 1
- Splunk Fundamentals 2
- Advanced Searching and Reporting
- *OR equivalent Splunk experience*

Course Goals

- Describe analytics, data science and related processes and terms:
  - Data visualizations
  - Statistics
  - Machine learning
- Recognize common use cases
- Explore, implement, and evaluate solutions to common analytics challenges
- Collaborate on effective projects with both technical and non-technical stakeholders

Course Outline

Module 1: Analytics Framework

Module 2: Exploratory Data Analysis

Module 3: Machine Learning Workflow

Module 4: Algorithms, Preprocessing and Feature Extraction

Module 5: Market Segmentation & Transactional Analysis

Module 6: Anomaly Detection

Module 7: Estimation & Prediction

Module 8: Classification

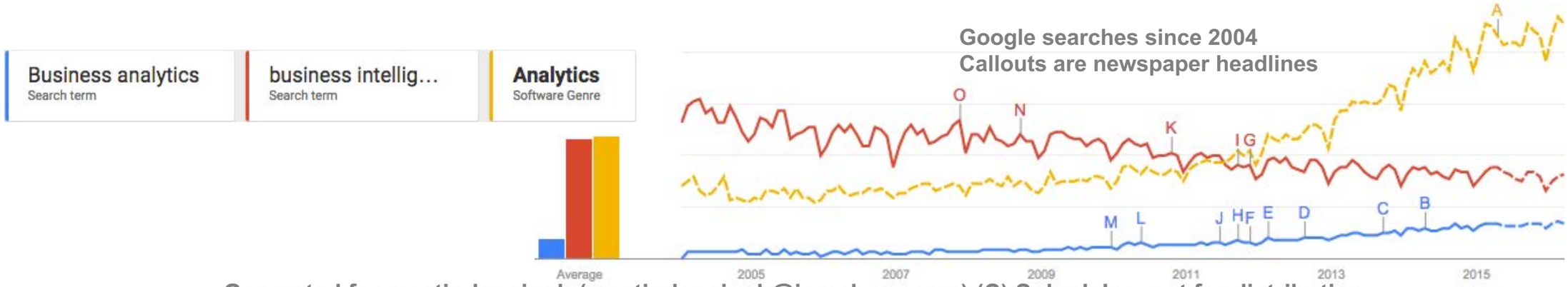
Module 1: Analytics Framework

Module Objectives

- Define analytics, data science, and related terms
- Identify roles in a data science team
- Describe the framework for analytics projects
- Identify analytics project best practices
- Identify common data science use cases for statistics and machine learning

Analytics

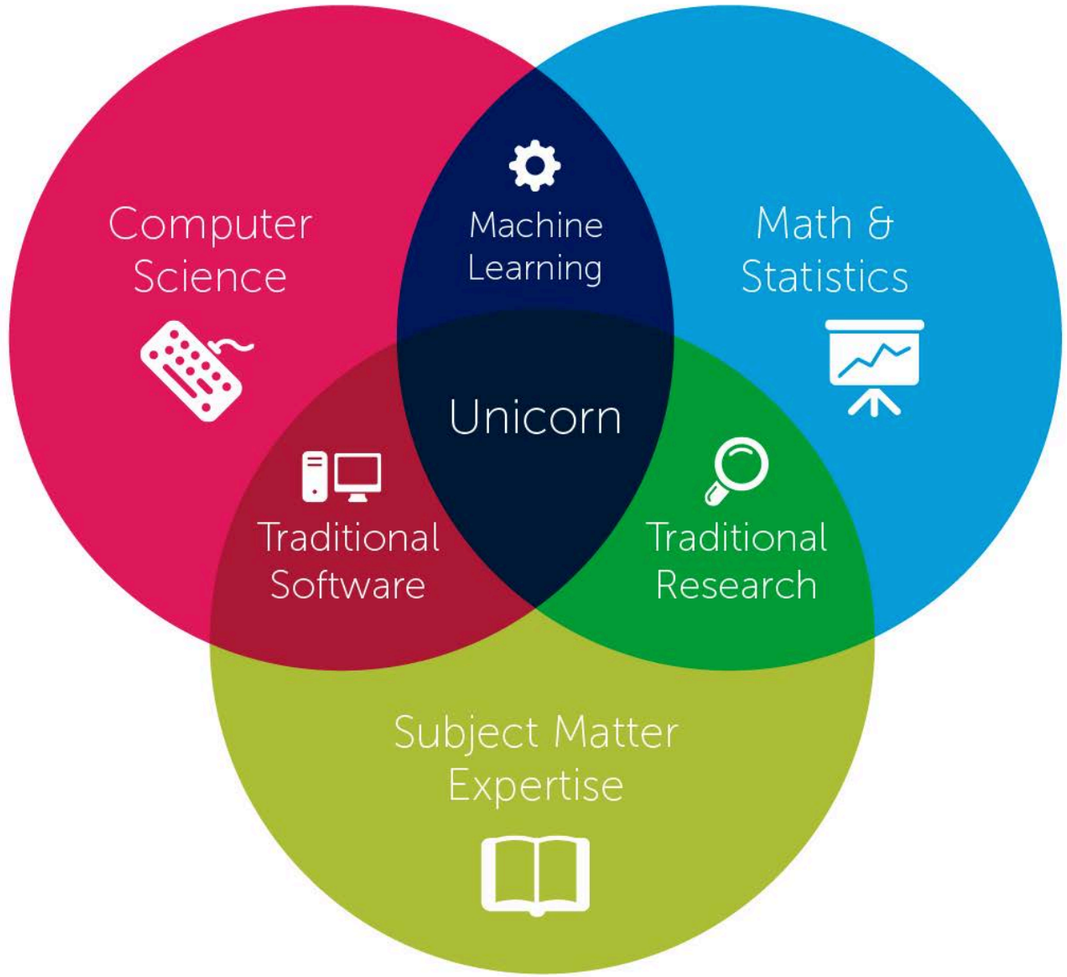
- The use of data to answer important questions:
  - Why is this happening? (root cause)
  - What will happen next? (prediction)
  - What's the best outcome that can happen? (optimization)
- Broader than Business Intelligence or Business Analytics
  - BI often measures historically structured data for business uses
  - BA is the use of real-time intelligence for operational decision-making



What is Data Science?

An emerging field that overlaps analytics: to extract **actionable insights** from data

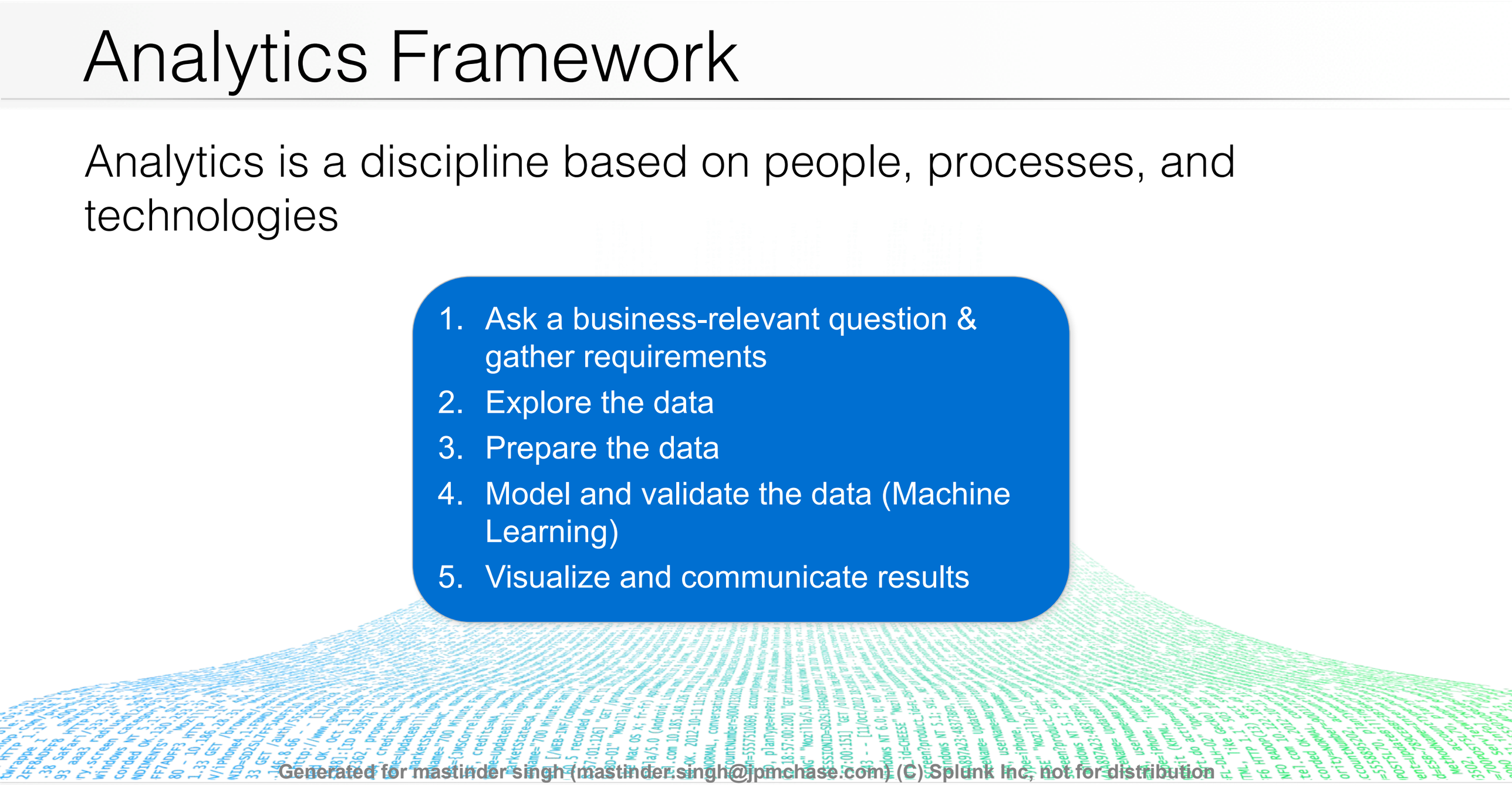
- Each data product is targeted to a specific persona
- Shifts between deductive (hypothesis-based) and inductive (pattern-based) reasoning



Analytics Framework

Analytics is a discipline based on people, processes, and technologies

1. Ask a business-relevant question & gather requirements
2. Explore the data
3. Prepare the data
4. Model and validate the data (Machine Learning)
5. Visualize and communicate results



1. Ask a Question / Gather Requirements

- Find the questions each stakeholder is trying to answer
  - Story trying to tell
  - Department
  - Fit in the organization
  - Primary contact
  - (Trained) power user
  - Engagement model
    - Self-service
    - Full change control
    - 2 hour power session

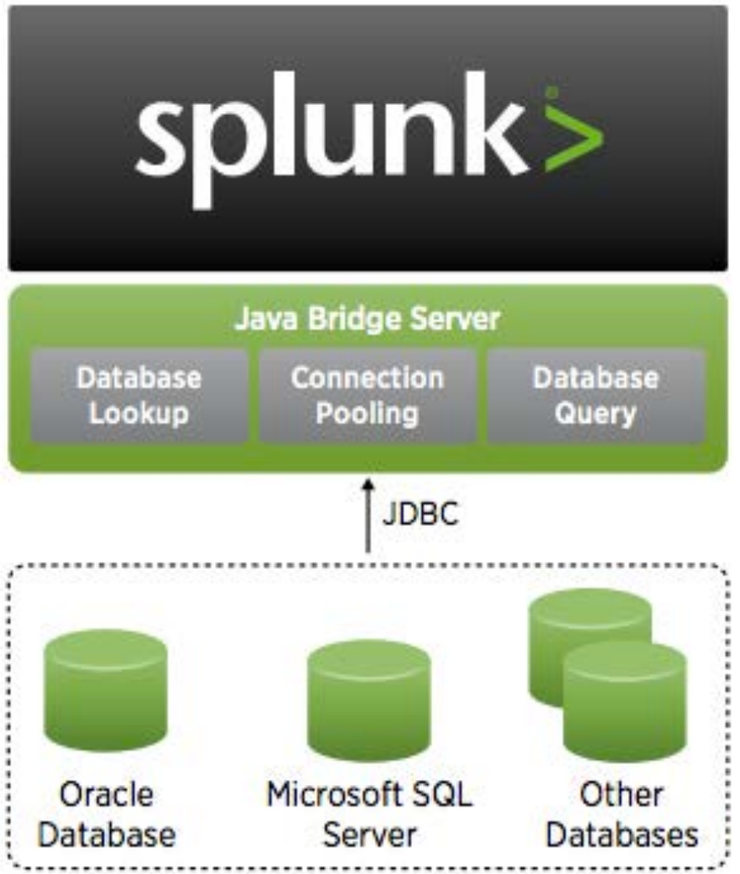
Business Problem	Search/Dashboard	Data Sources	Built By	Personas	Consumption Method
Product Adoption	B2B PM Dashboard • Usage by Location • Platform Metrics • Platform Trends	Apache Customer Info SFDC Product DB	PM Search Expert	PM	Splunk
Customer Behavior	Data Model/Pivot • KPIs • Bounce Rate • Pathing • Conversions	Apache	Self	Analyst	Splunk Pivot
Executive Dashboard	Executive Dashboard • KPIs • Revenue Trends • Uptime/Capacity	All	Splunk PS	Exec Management	PDF via email

2. Explore the Data

- **First Rule of Data Science: Know your data.** Learn the shape of the data and look for unanticipated characteristics
  - Maximize insight into a data set
  - Uncover underlying structure
  - Test assumptions
- Visualize the data in multiple ways
  - Raw data (such as histograms)
  - Simple statistics such as mean, standard deviation, box plots, etc.
  - Adjust to maximize the brain's natural pattern-recognition abilities
- Target promising potential relationships

3. Prepare the Data

- Clean & onboard data/configure
  - Are people using BI tools?
  - Who owns data onboarding /configs?
  - Who is the project manager?
- What are the data sources? IT & Business
  - Onboard & configure Splunk data sources
  - Configure all relational data sources
  - Identify & document all relevant SQL queries
- Make sure you have all needed fields
- Fix gaps and nulls, convert units, etc.



4. Model & Validate the Data (ML)

- After exploring, to determine which datasets are likely to be useful
- Make and save models to organize and frame that data over time
  - How might the fields you care about be related?
  - Can fields be derived from other fields?

```
Net_Profit = Sales_Revenue – Total_Costs
```

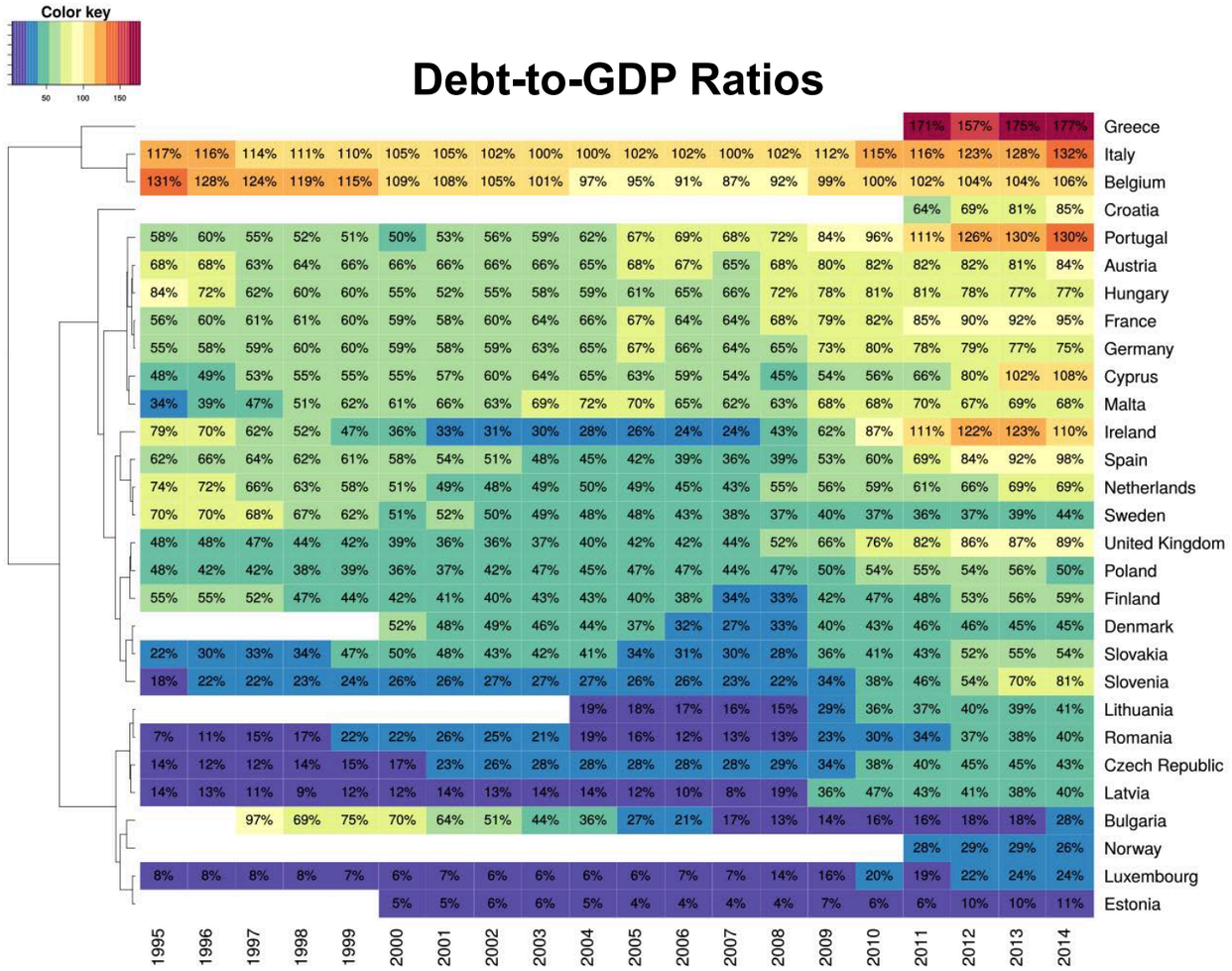
- Simple models are easy to use and understand
- Complex models could increase accuracy, but may decrease interpretability

5. Visualize & Communicate Results

- Highlight business objectives on each dashboard
- Target data to different personas
  - **Business Beneficiary:** gets value from the data
  - **Business Analyst:** deeply uses 3-5 dashboards
  - **Power User:** helps build complicated SPL & dashboards
- Explore all details of security and access control
  - Limit access to only what they need
  - Configure in-dashboard drilldowns by persona
    - Disable full search functionality
    - If somebody needs data, configure a channel specifically for that persona

Good Visualization?

Countries with similar ratio development are placed near one another



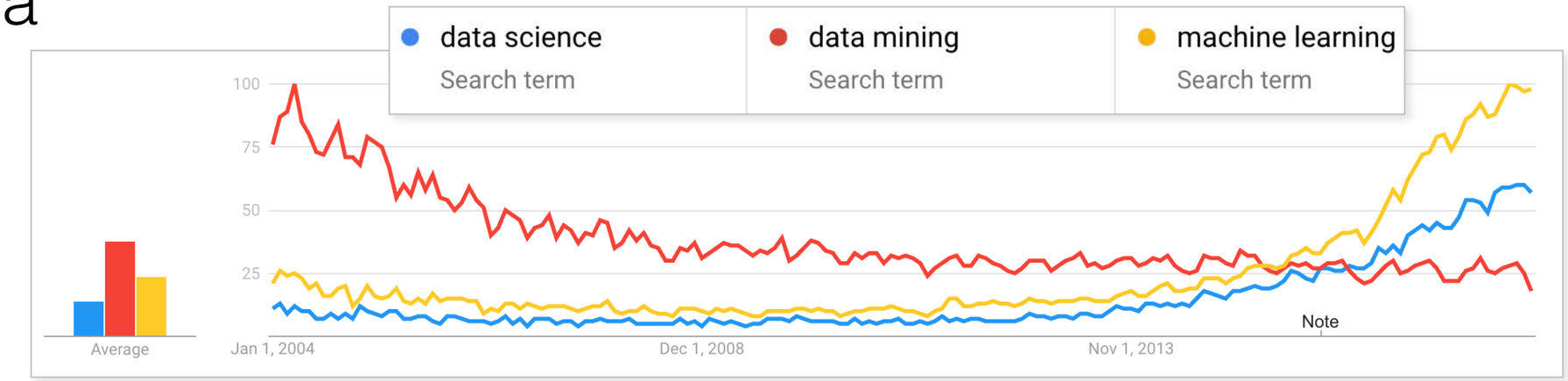
Debt-to-GDP ratio for some European countries; data are taken from Eurostat data explorer (31 may 2015, Table: gov_10dd_edpt1).

Optimizing Deployments

- Examine and improve all processes
  - Optimize searches
  - Speed up dashboards
  - Maximize efficiency
    - Watch for cases of moving large amounts of data in and out of Splunk (e.g. into lookup, via ODBC)
  - Implement dedicated user roles from the beginning
  - Verify the plan scope and the long term usage for each stakeholder
    - How do different workflows fit together? Will dashboards be used daily? Weekly?
    - How are dashboards and deliverables iterated? How often revised?

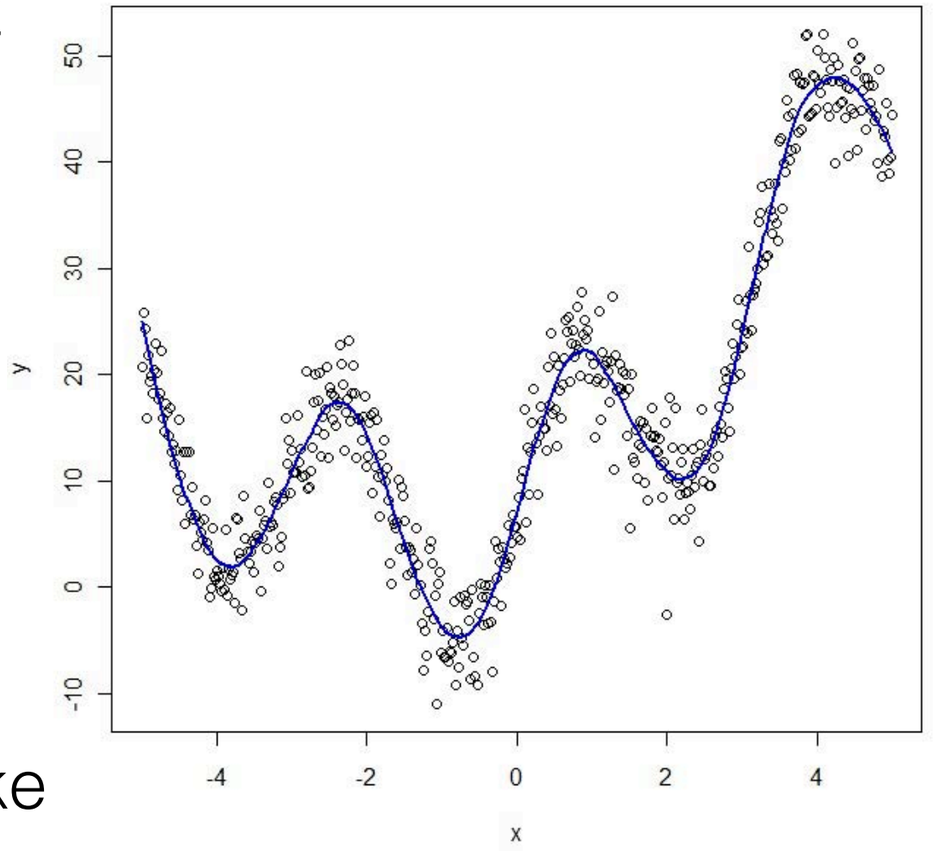
Data Science, Stats & Machine Learning

- **Statistics:** a branch of math that provides theoretical and practical support for the use of tools and processes
  - Machine Learning: a process for generalizing from examples
- **Data Science:** using all available tools and processes to provide actionable insights to stakeholders in all organizational areas
- **Data Mining:** looking for useful information in large amounts of database data



Statistical Models

- A statistical model is a formal specification of all variables of interest and a parameterized relationship between those variables
- Why care about statistical models?
  - Reasoning – simplifications of reality make it easier to reason
  - Programmable – can be converted into algorithms
  - Reusable – concisely described to make leveraging easy



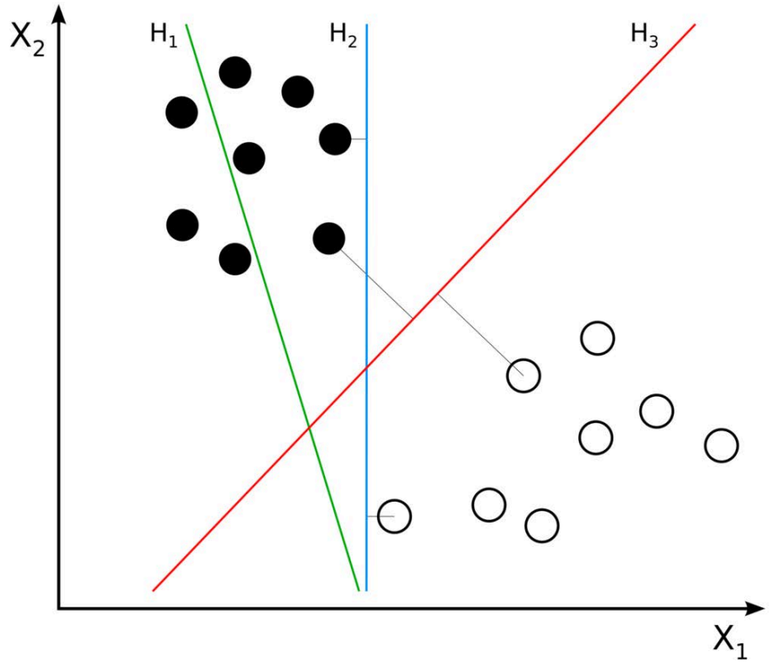
ML Toolkit: `fit` creates a model

Applied Statistics

- Explains data in terms of other data (multiple layers of data)
- Makes decisions based on these explanations
- Always has a business problem attached
- Consider some statistics use cases in Splunk
  - **Exploratory Data Analysis:** Explore data before using a model
  - **Market Segmentation:** Assign customers to clusters not known
  - **Transactional Analysis:** Group events into transactions
  - **Anomaly Detection:** Find unexpected events
  - **Estimation and Prediction:** Use data to predict unknowns
  - **Classification:** Unseen inputs into pre-defined classes: spam filter

Data Science & Machine Learning (ML)

- Data Science helps find what is interesting, relevant, and valuable
- People decide on models, variables, and acceptable margins of error
- Machine learning can help automate the decision-making process



Machine Learning Success Stories

Company	Service	Description
StitchFix	Stylists use algorithms to provide a personal shopping service	Hundreds of styling algorithms that match products to clients, pair stylists with clients, measure customer happiness, and select inventory.
SoundHound	Hound virtual assistant	Natural Language Processing (Samsung, Nvidia, Sony Xperia, Yelp, Uber)
Descartes Labs	Prevent food shortages by predicting crop yields	Applies machine learning to 3 petabytes of satellite imagery datasets
Iris AI	Speeds up scientific research	Surfaces relevant data within millions of published articles
Trademark Vision	Unique logos	Find new logos that identify a company without infringing on copyrights, and alert copyright holders of infringements
IBM	Watson applied to weather data	New analysis of weather patterns ($500 billion of annual commerce is weather-dependent)

Module 1 Lab Exercise – Review Available Data

Time: 10 minutes

Tasks:

- Log in to Splunk on the classroom server
- Review available data
- Change your account settings to reflect your name and local time zone

Module 2: Exploratory Data Analysis

Module Objectives

- Describe the purpose of data exploration
- Explore data using Splunk
  - `kmeans` command
  - `cluster` command
  - Patterns tab
  - `correlate` command
  - `associate` command

Exploratory Data Analysis

- Exploring data without first building a model
- Splunk
  - Helps you navigate and explore the data
  - Allows the relationships between the data to rise to the surface
- Once you find those relationships between the data:
  - You can tell stories about the data
  - The *stories* act as a proxy for the model

Start Simple

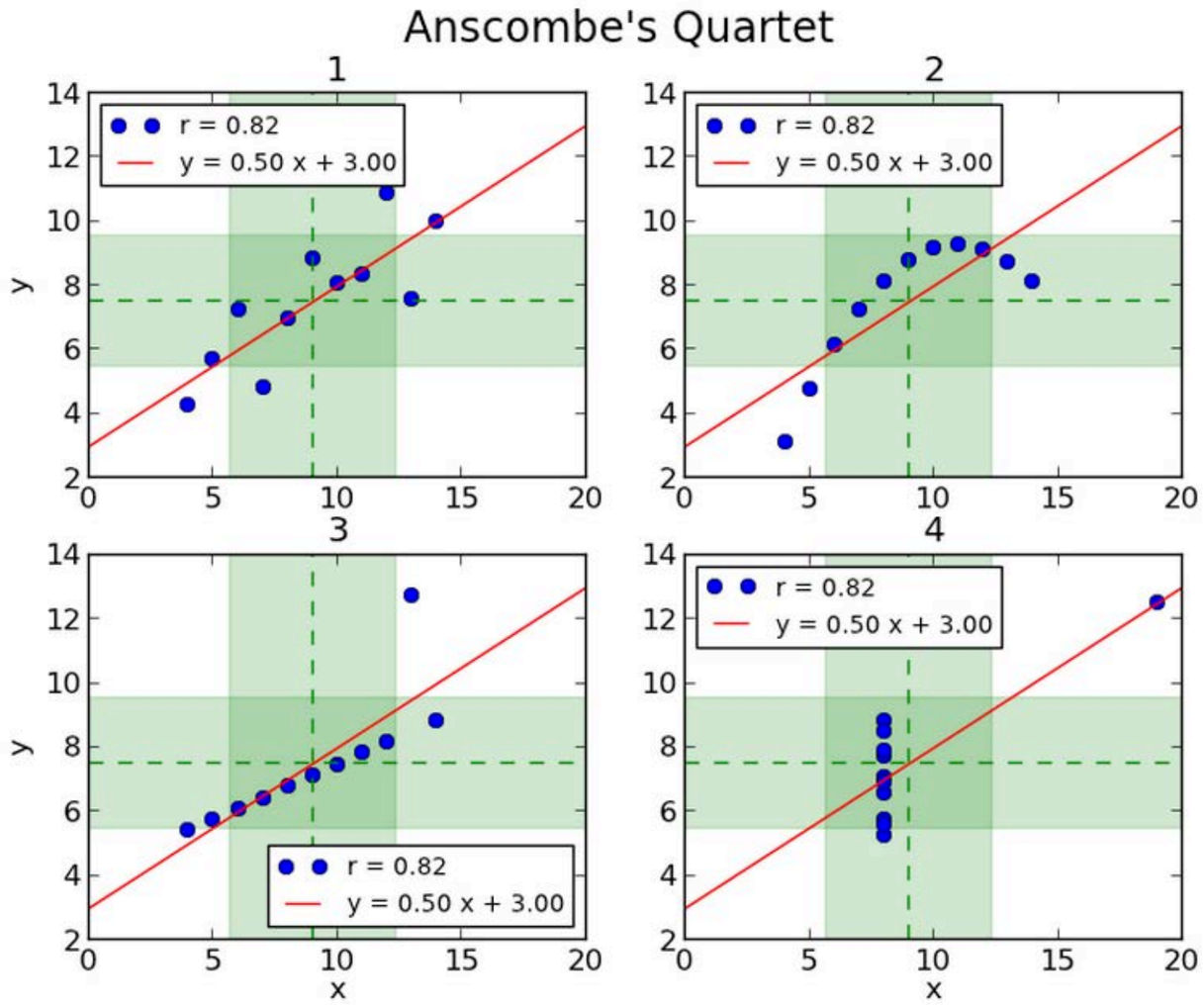
data set 1		data set 2		data set 3		data set 4	
X1	Y1	X2	Y2	X3	Y3	X4	Y4
10	8.04	10	9.14	10	7.46	8	6.58
8	6.95	8	8.14	8	6.77	8	5.76
13	7.58	13	8.74	13	12.74	8	7.71
9	8.81	9	8.77	9	7.11	8	8.84
11	8.33	11	9.26	11	7.81	8	8.47
14	9.96	14	8.1	14	8.84	8	7.04
6	7.24	6	6.13	6	6.08	8	5.25
4	4.26	4	3.1	4	5.39	19	12.5
12	10.84	12	9.13	12	8.15	8	5.56
7	4.82	7	7.26	7	6.42	8	7.91
5	5.68	5	4.74	5	5.73	8	6.89

.81 .81 .81 .81

Correlation Coefficient

Wikipedia : dataMind.co

Visualization as Exploration



data set 1		data set 2		data set 3		data set 4	
X1	Y1	X2	Y2	X3	Y3	X4	Y4
10	8.04	10	9.14	10	7.46	8	6.58
8	6.95	8	8.14	8	6.77	8	5.76
13	7.58	13	8.74	13	12.74	8	7.71
9	8.81	9	8.77	9	7.11	8	8.84
11	8.33	11	9.26	11	7.81	8	8.47
14	9.96	14	8.1	14	8.84	8	7.04
6	7.24	6	6.13	6	6.08	8	5.25
4	4.26	4	3.1	4	5.39	19	12.5
12	10.84	12	9.13	12	8.15	8	5.56
7	4.82	7	7.26	7	6.42	8	7.91
5	5.68	5	4.74	5	5.73	8	6.89
	.81		.81		.81		.81

Correlation Coefficient

Wikipedia : dataMind.co

bin command (bucket)

Note

Useful for making frequency distributions or histograms of numerical data.

- Puts continuous numerical values into discrete sets, or bins
- Adjusts value of a numerical field so all items in a set have the same value range
  - **span=** option of **timechart** is one way bins are formed

bin [<bin-options>...] <field> [AS <newfield>]

- options
  - **bins** maximum number of bins (**bins=20**)
  - **span** sets the size for each bin (**span=250**)
    - If **span** creates more sets than the max specified by **bins**, **bins** is ignored
  - **minspan** specifies the smallest span size to use for each bin
  - **<start-end>** specifies minimum and maximum for numerical bins
    - When no span is used

bin Command Examples

```
sourcetype=access_combined
| stats sum(price) as totalSales by product_name
| bin totalSales bins=10
| stats list(product_name) as product_name by totalSales
| eval totalSales = "$".totalSales
```

totalSales	product_name
$0-1000000	Curling 2014 Fire Resistance Suit of Provolone Holy Blade of Gouda Manganiello Bros. Tee Puppies vs. Zombies World of Cheese Tee
$1000000-2000000	Benign Space Debris Final Sequel Mediocre Kingdoms SIM Cubicle
$2000000-3000000	Manganiello Bros. Orvil the Wolverine World of Cheese
$3000000-4000000	Dream Crusher

```
sourcetype=access_combined
| bin _time span=1h
| stats sum(bytes) as totalBytes
by _time, host
| eval totalBytes = round(totalBytes/(1024*1024),2)." MB"
| xyseries _time, host, totalBytes
```

_time	www1	www2	www3
2018-06-08 04:00	0.20 MB	0.18 MB	0.17 MB
2018-06-08 05:00	0.18 MB	0.31 MB	0.12 MB
2018-06-08 06:00	0.21 MB	0.14 MB	0.26 MB
2018-06-08 07:00	0.14 MB	0.21 MB	0.14 MB
2018-06-08 08:00	0.11 MB	0.20 MB	0.18 MB
2018-06-08 09:00	0.10 MB	0.16 MB	0.21 MB

makecontinuous

- Makes a field on the x-axis numerically continuous
  - Where no data exists, adds empty bins
  - Where there is data, quantifies the periods
  - Use chart or timechart to invoke this new x-axis value

```
makecontinuous [<field>]
<bins-options>...
```


fieldsummary

- Calculates summary statistics for all or a subset of fields in your search results
- Summary information is displayed as a results table including:

`field`	field name in the event
`count`	number of events/results with that field
`distinct_count`	number of unique values in the field*
`is_exact`	whether or not the field is exact
`max`	if numeric, the maximum of its value
`mean`	if numeric, the mean of its values
`min`	if numeric, the minimum of its values
`numeric_count`	count of numeric values in the field (excludes NULL values)
`stdev`	if numeric, the standard deviation of its values
`values`	distinct values of the field and count of each value

* Related to `distinc_count`. If the number of values of the field exceeds `maxvals`, then `fieldsummary` will stop retaining all the values and compute an approximate distinct count instead of an exact one. 1 means it is exact; 0 means it is not.

`fieldsummary` Syntax & Example

```
fieldsummary [maxvals=<num>] [<wc-field-list>]
```

- Optional arguments
  - **`maxvals`** max distinct values to return for each field (default: 100)
  - **`wc-field-list`** field(s) that can include fields with wildcards

```
sourcetype=sendmail_syslog | fieldsummary maxvals=3
```

field	count	distinct_count	is_exact	max	mean	min	numeric_count	stdev	values
change_type	0	0	1				0		[]
class	676	1	1	0	0	0	676	0	[{"value":"0","count":676}]
ctladdr	618	8	0				0		[{"value":"britany@mailsv1.splunk.com (665/666)","count":178}, {"value":"hammer@mailsv1.splunk.com (967/967)","count":126}, {"value":"madonna@mailsv1.splunk.com (662/663)","count":124}]
daemon	672	1	1				0		[{"value":"MTA","count":672}]

Preparing and Cleaning Data

Cleaning data is modifying the raw data to make it consistent and organized enough to input to your analytical algorithm

- Also called munging or wrangling
  - Sometimes creates some derived data, such as creating unique identifiers
  - Sometimes fills null or missing values
- Making data “tidy” makes it possible to fit a model
  - Variables or features should be the columns
  - Samples or data points should be the rows
- 80% is a common estimate of data science time spent munging

Data Munging Commands

`trendline`	moving averages of fields
`erex`	extract data from a field when regex isn't known
`filldown` or `autoregress`	fills in blanks with previous values
`replace`	replaces first string in a specified field or all fields
`search field=*`	filter out missing values
`fillnull`	fill in missing values
`rex mode=sed "s/\W//g"`	remove non-word characters
`eval{foo}=1|fillnull`	dummy encoding
`xyseries / untable`	pivots

```
| inputlookup airline_tweets.csv
| fields text|rex field=text
   mode=sed "s/\W/ /g"
| eval text = lower(text)
```

```
sourcetype=access_combined action=*
|eval is_{action}=1
|fillnull|table is_*
```

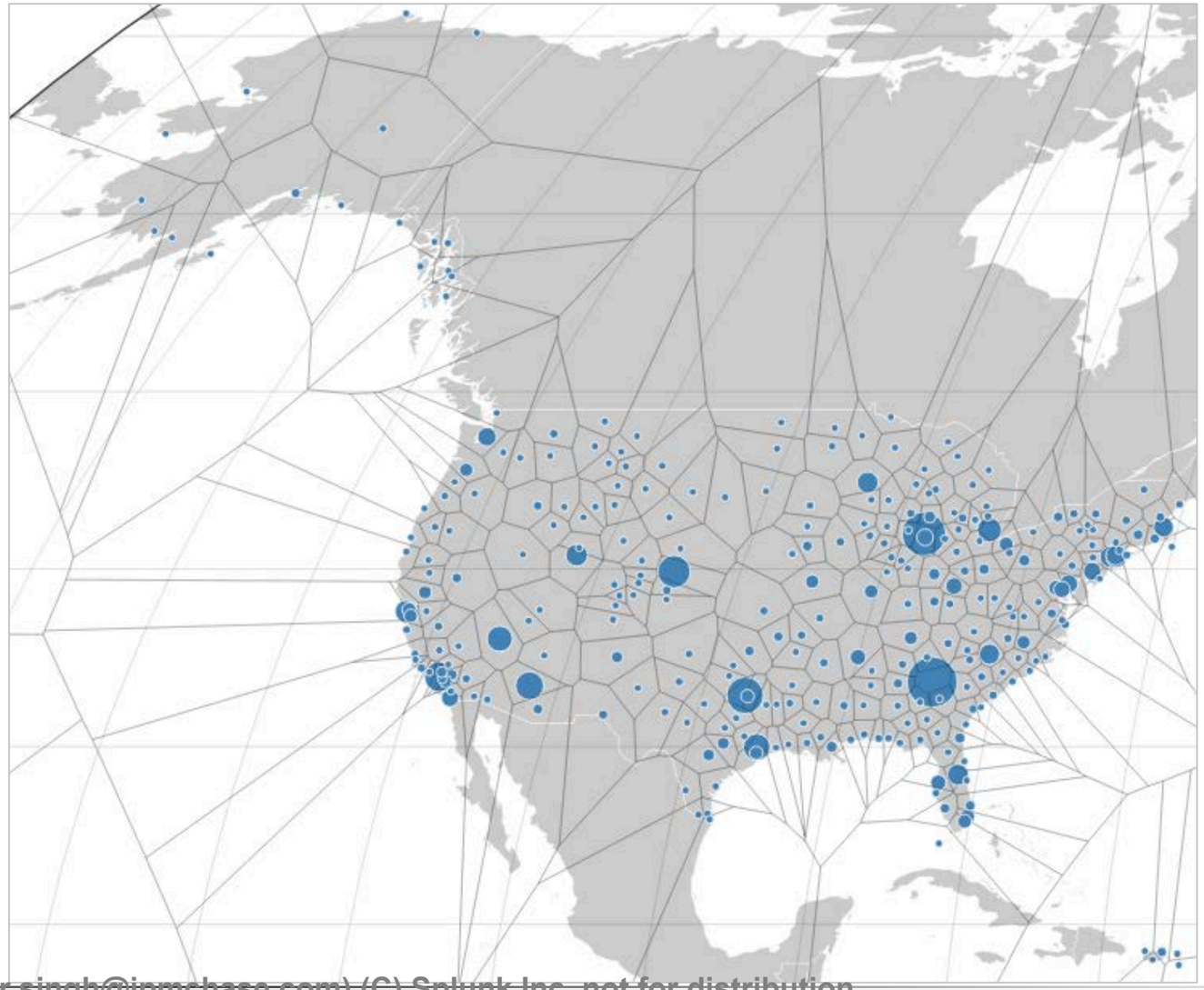

Built-in Commands for EDA

Exploratory Data Analysis Concept	Splunk Command
Find **common and/or rare events** in your data	`cluster` (Patterns tab) `kmeans`
Examine the **relationship among all the *fields*** in a set of search results (percentage of times the two fields co-occur in the same events)	`correlate`
Analyze all **numerical fields** to see how well each might predict the value of a field you choose: the target classified; see appendix	`analyzefields`
Find changes in **entropy*** between pairs of fields based on their **values** *The more evenly a field's values are distributed, the higher its entropy	`associate`

`kmeans` Command for Numerical Fields

- Divides data into k clusters
- Each data point belongs to the cluster with the nearest mean (centroid)
- Partitioned into Voronoi cells
- **Numerical** fields common to both results sets are used

```
<reps>|<iters>|<tol>|<k>
|<cnumfield>|<distype>
```



`kmeans` Command Options

- `reps`: repeats of `kmeans` using random starting clusters (default:10)
- `iters`: maximum number of iterations allowed (default: 10,000)
- `t`: algorithm convergence tolerance (default: 0)
- `k`: number of clusters to use or range of numbers (default: 2)
  - `range` clusters for each cluster count in the range, include the size of the clusters and a `'distortion'` field: how well the data fits those clusters
- `cnumfield`: names the field to annotate results (default: `CLUSTERNUM`)
- `dt` is the distance metric to use (default: `sqeuclidean`)
  - l1, l1norm, and cb: cityblock
  - l2, l2norm, and sq: sqeuclidean
  - cos: cosine

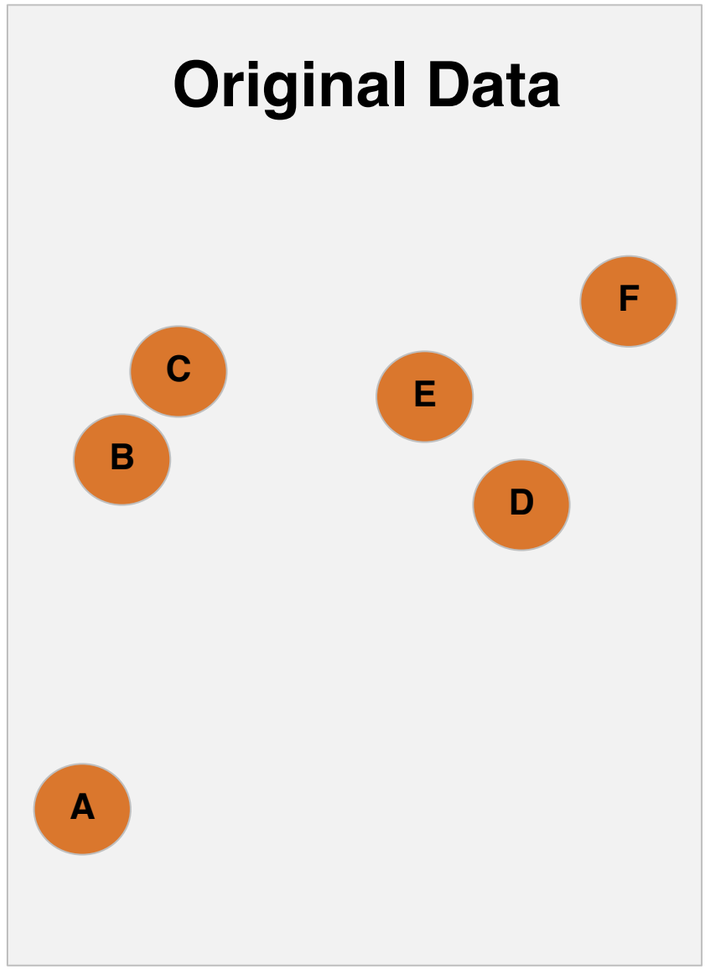
kmeans Example

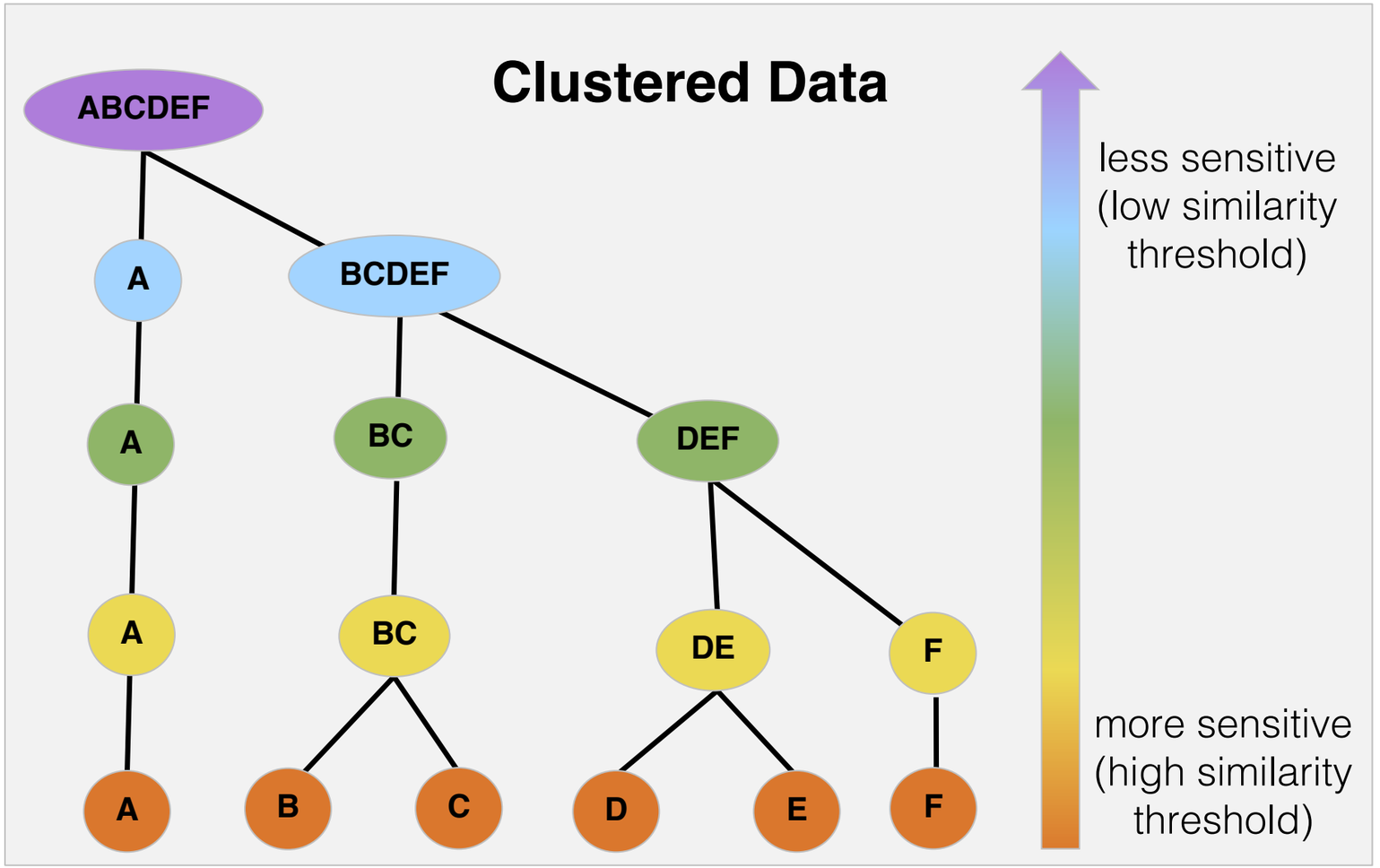
```
sourcetype=access_combined action=purchase
| kmeans k=3 price
| stats values(product_name) as product, avg(price) as average_price by CLUSTERNUM
```

CLUSTERNUM	product	average_price
1	Dream Crusher Manganiello Bros. Orvil the Wolverine	39.98999999999985
2	Benign Space Debris Curling 2014 Final Sequel Mediocre Kingdoms SIM Cubicle World of Cheese	23.406886543535453
3	Fire Resistance Suit of Provolone Holy Blade of Gouda Manganiello Bros. Tee Puppies vs. Zombies World of Cheese Tee	6.876287625418069

`cluster` Command (Agglomerative)

Iteratively groups events based on a minimum similarity threshold





`cluster` Command

Groups events based on patterns it detects in event text streams

- Based on cosine similarity of the **`_raw`** field by default
  - Can be changed using **`field=<field>`** (i.e. **`field=action`**)
- Breaks fields into terms, computes the vector between events
- Creates 2 new fields named **`cluster_label, cluster_count`**
  - Default field names can be changed using **`labelfield=<field>`** and **`countfield=<field>`**

How `cluster` Breaks Fields into Terms

- Breaks fields into terms, computes the vector between events
  - `match=termlist` (default) breaks fields into words
    - Requires identical order of terms
  - `match=termset` breaks fields into terms in *any order*
  - `match=ngramset` compares sets of trigrams (3-character substrings)
    - Most useful for short non-textual fields, like punct

`cluster` Command Options

`labelonly=false` (default)
one event from each cluster is returned

- `labelonly=true` all events

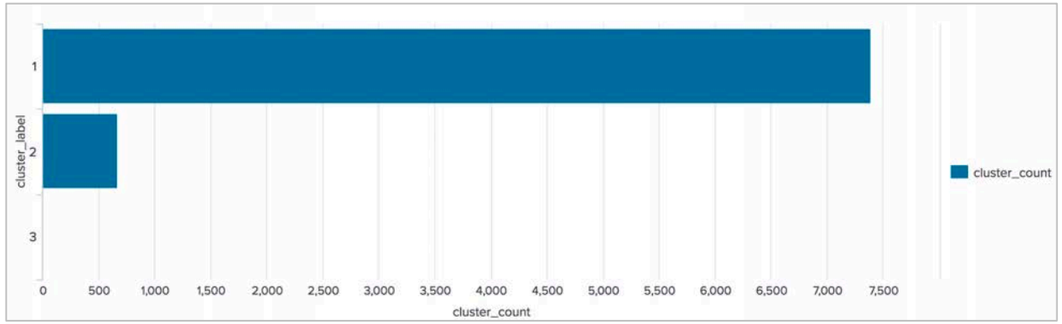
`showcount=true` all events get `cluster_count` field

`t=.8` (threshold) adjusts cluster sensitivity

- lower `t` = fewer clusters

`| dedup 5 cluster_label`
returns the 5 most recent grouped events in each cluster

```
sourcetype=cisco_esa
| cluster t=0.2 showcount=t
| table cluster_label cluster_count _raw
| sort -cluster_count
```

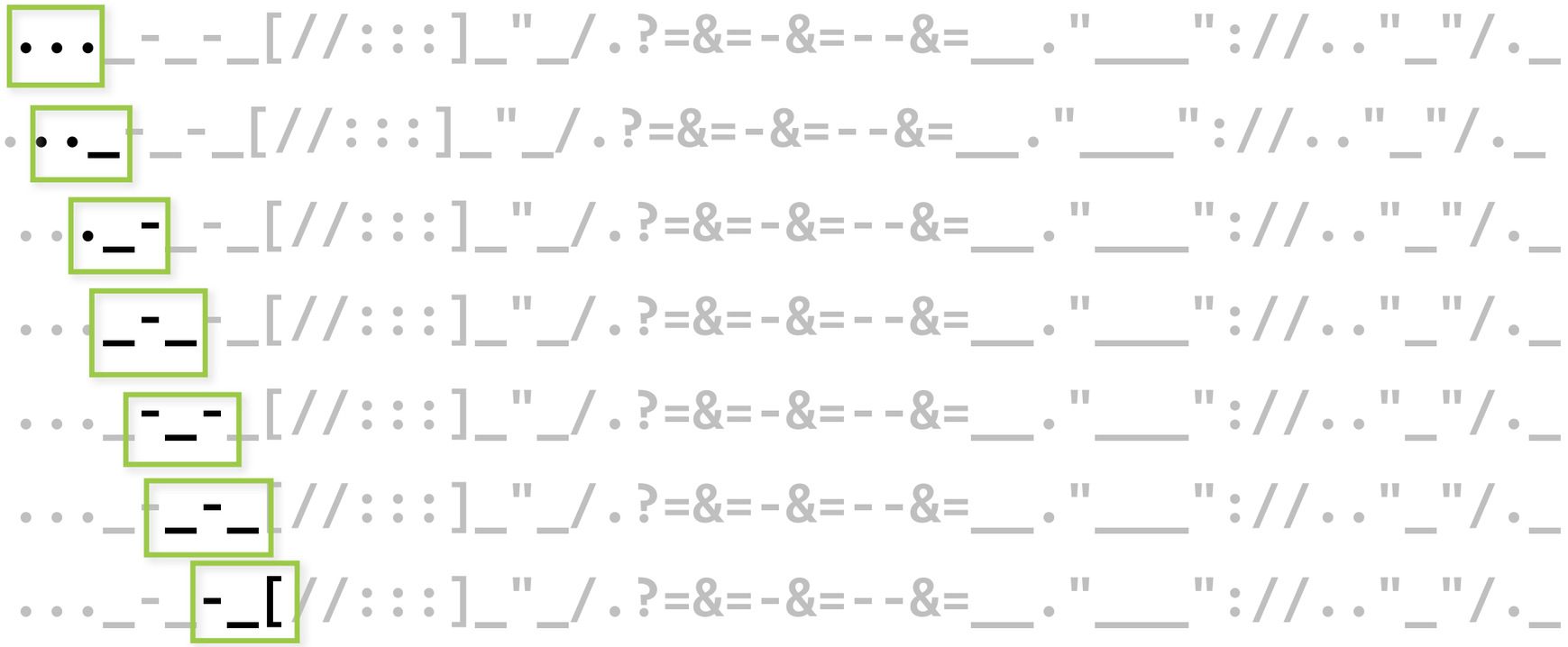


cluster_label	cluster_count	_raw
1	7338	Sat Jun 09 17:06:41 2018 Info: MID 245078 queued for delivery
2	700	Fri Jun 08 23:58:43 2018 Info: New SMTP ICID 743953 interface Management (192.168.3.120) address 206.176.229.254 reverse dns host ironport.mineralore.com verified yes
3	2	Fri Jun 08 13:42:29 2018 Warning: Received an invalid DNS Response: rcode=ServFail data="'@\\xba \\x80\\x02\\x00\\x01\\x00\\x00\\x00\\x00\\x00\\x00\\x0278\\x0269\\x03152\\x0285\\x07in addr\\x04arpa \\x00\\x00\\x0c\\x00\\x01'" to IP 193.0.0.193 looking up 78.69.152.85.in addr.arpa

n-gram

Moving window of character strings of length n (below, n=3)

Original data: `..._-_-_[//:::]_"_/.?=&=-&=--&=__."___"://.."_"/._`



Cosine Similarity

Count of matching terms in A and B (Intersection)

$$\text{cosine}_{AB} = \frac{n(A \cap B)}{\sqrt{n(A) \times n(B)}}$$

Number of terms in A Number of terms in B

EVENT	TEXT	# OF TERMS
A	INFO user generated content blocked by filter	7
B	WARN system generated alert from filter	6

Cosine Similarity Termlist Term Count

Intersection of A and B

$$\text{cosine}_{AB} = \frac{n(A \cap B)}{\sqrt{n(A) \times n(B)}}$$

Number of terms in A Number of terms in B

EVENT	TEXT	# OF TERMS
A	INFO user generated content blocked by filter	7
B	WARN system generated alert from filter	6

Cosine Similarity Termlist Intersection

Intersection of A and B = 1

$$\text{cosine}_{AB} = \frac{n(A \cap B)}{\sqrt{n(A) \times n(B)}}$$

Number of terms in A Number of terms in B

EVENT	TEXT	# OF TERMS
A	INFO user generated content blocked by filter	7
B	WARN system generated alert from filter	6

Cosine Similarity Termlist Calculation

Intersection of A and B = 1

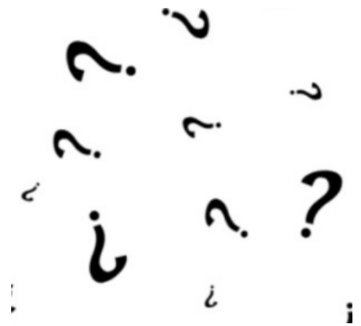
$$\text{cosine}_{AB} = \frac{n(A \cap B)}{\sqrt{n(A) \times n(B)}}$$

Number of terms in A Number of terms in B

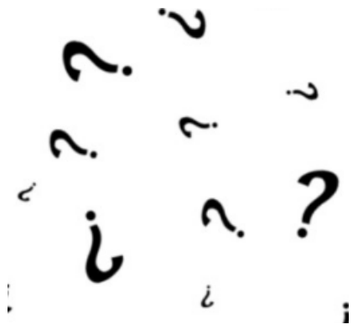
EVENT	TEXT	# OF TERMS
A	INFO user generated content blocked by filter	7
B	WARN system generated alert from filter	6

$$= \frac{1}{\sqrt{7 \times 6}} \approx 0.1543$$

Cosine Similarity – Termset



How might it differ?



Intersection of A and B

$$\text{cosine}_{AB} = \frac{n(A \cap B)}{\sqrt{n(A) \times n(B)}}$$

Number of terms in A Number of terms in B

EVENT	TEXT	# OF TERMS
A	INFO user generated content blocked by filter	7
B	WARN system generated alert from filter	6

Cosine Similarity – Termset Term Count

Intersection of A and B

$$\text{cosine}_{AB} = \frac{n(A \cap B)}{\sqrt{n(A) \times n(B)}}$$

Number of terms in A Number of terms in B

EVENT	TEXT	# OF TERMS
A	INFO user generated content blocked by filter	7
B	WARN system generated alert from filter	6

Cosine Similarity –Termset Intersection

Intersection of A and B = 2

$$\text{cosine}_{AB} = \frac{n(A \cap B)}{\sqrt{n(A) \times n(B)}}$$

Number of terms in A Number of terms in B

EVENT	TEXT	# OF TERMS
A	INFO user generated content blocked by filter	7
B	WARN system generated alert from filter	6

Cosine Similarity –Termset Calculation

Intersection of A and B = 2

$$\text{cosine}_{AB} = \frac{n(A \cap B)}{\sqrt{n(A) \times n(B)}}$$

Number of terms in A Number of terms in B

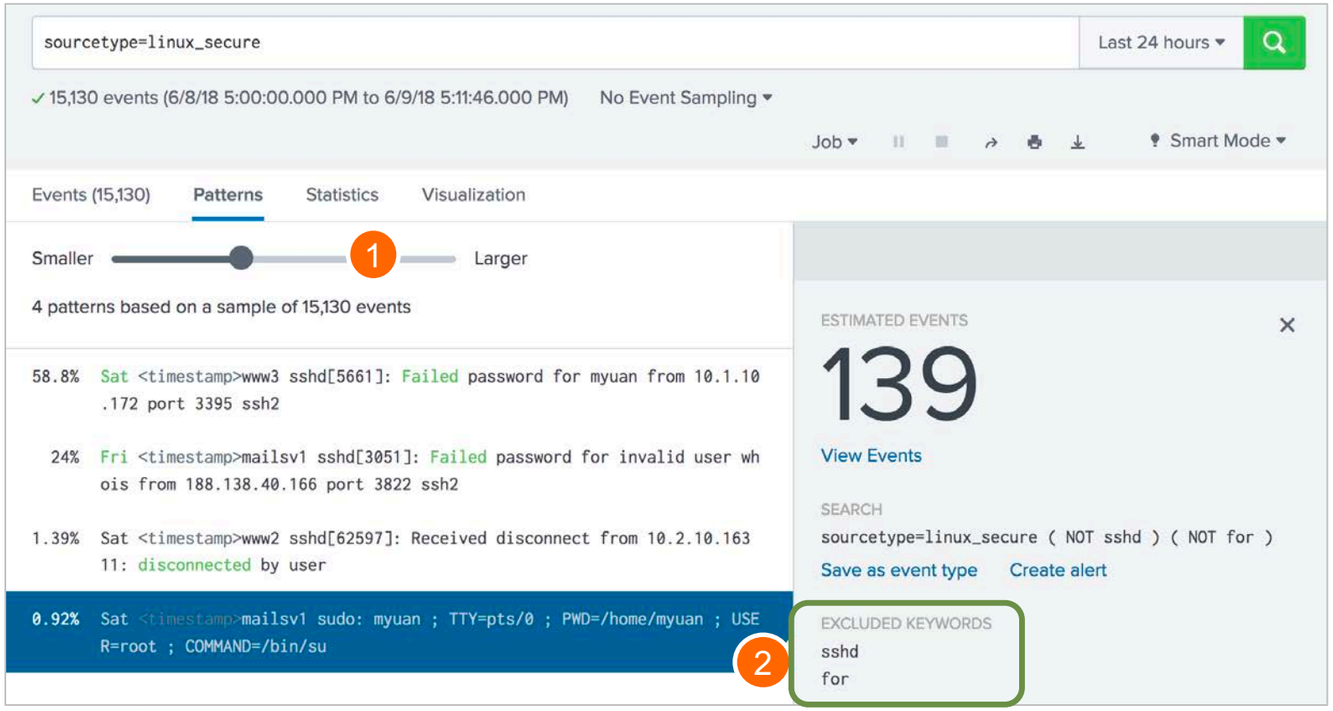
EVENT	TEXT	# OF TERMS
A	INFO user generated content blocked by filter	7
B	WARN system generated alert from filter	6

$$= \frac{2}{\sqrt{7 \times 6}} \approx 0.3086$$

Event Pattern Detection

- Some patterns are rare and difficult to find in the Events tab
- Event pattern detection is the **`cluster`** command in Splunk UI
- Use the Pattern tab as a first step to exploring your data

1. Drag the slider to view events
   More generically (Larger)
   or
   More specifically (Smaller)
2. Click the smallest cluster to view which keywords were used



`table` Command: Event Pattern Detection

3. Add a `cluster` with `showcount`, `t` value, `labelonly` and `table` with `cluster_label`, `cluster_count` and `_raw` to your search
4. Examine the Statistics tab

```
sourcetype=linux_secure
3 | cluster showcount=t t=0.5 labelonly=t
| table cluster_label, cluster_count, _raw
```

4

cluster_label	cluster_count	_raw
2	212	Sat Jun 09 2018 17:13:05 www3 sshd[68990]: Received disconnect from 10.1.10.172 11: disconnected by user
1	4551	Sat Jun 09 2018 17:12:58 www3 sshd[3107]: Failed password for root from 110.138.30.229 port 2114 ssh2
1	4551	Sat Jun 09 2018 17:12:56 www3 sshd[2107]: Failed password for invalid user helpdesk from 10.1.10.172 port 3352 ssh2
1	4551	Sat Jun 09 2018 17:12:50 www3 sshd[3892]: Failed password for myuan from 10.1.10.172 port 4781 ssh2
1	4551	Sat Jun 09 2018 17:12:42 www3 sshd[1449]: Failed password for invalid user sapadmin from 175.44.1.122 port 4406 ssh2
1	4551	Sat Jun 09 2018 17:12:41 www3 sshd[5227]: Failed password for invalid user hsqldb from 110.138.30.229 port 2074 ssh2
1	4551	Sat Jun 09 2018 17:12:28 www2 sshd[4564]: Failed password for invalid user angel from 10.1.10.172 port 3874 ssh2
1	4551	Sat Jun 09 2018 17:12:24 www3 sshd[4143]: Failed password for invalid user cyrus from 110.138.30.229 port 4008 ssh2
3	1298	Sat Jun 09 2018 17:12:16 www2 sshd[28641]: pam_unix(sshd:session): session closed for user djohnson by (uid=0)

`findkeywords` Command: `Patterns` Tab

5. Replace table with **`findkeywords`**
6. View details of how Splunk clustered
   - Included keywords and their values
   - Excluded keywords and their values

```
sourcetype=linux_secure
| cluster showcount=t t=0.5 labelonly=t
| findkeywords labelfield=cluster_label
```

5

6

confidence	eventTypeable	excludeKeywords	groupID	includeKeywords	numInInputGroup	numMatched	percentInInputGroup	percentMatched	sampleEvent	search
1	1		1	invalid	9083	9066	0.5980379246773769	0.5969186199631288	Sat Jun 09 2018 17:17:10 www1 sshd[4145]: Failed password for invalid user tavi from 121.254.179.199 port 4653 ssh2	search sourcetype=linux_ invalid
0	1		5	ssh2	4101	13184	0.270015801948907	0.8680537266262839	Sat Jun 09 2018 17:13:08 www3 sshd[88015]: Accepted password for djohnson from 10.3.10.46 port 8305 ssh2	search sourcetype=linux_ ssh2
1	1		2	session	1308	1373	0.08612062154332367	0.09040031603897813	Sat Jun 09 2018 17:16:50 www2	search sourcetype=linux_

`correlate` Command

View co-occurrence between fields (not values) in a matrix format

- 1.0: in event(s) that contain one field, the other is always present
- Processes data only for the first N (default 1000)
  - **`Maxfields`** – set in **limits.conf**
  - Increasing default may have memory / CPU costs

```
sourcetype=sendmail_syslog
| correlate
```

- Ex: find the co-occurrence between all fields in e-commerce

RowField	class	ctladdr	daemon	date_hour	date_mday	date_minute	date_month	date_second	date_wday	date_year	date_zone	delay	dsn	eventtype	from
class	1.00	0.00	0.99	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.00	0.00	0.29	1.00
ctladdr	0.00	1.00	0.00	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.85	0.85	0.27	0.00
daemon	0.99	0.00	1.00	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.00	0.00	0.29	0.99
date_hour	0.29	0.27	0.29	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.31	0.31	1.00	0.29
date_mday	0.29	0.27	0.29	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.31	0.31	1.00	0.29
date_minute	0.29	0.27	0.29	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.31	0.31	1.00	0.29

`associate` Command

- Can knowing the value of a field predict the value of another?
  - Calculates change in entropy (unpredictability) based on field values
- Fields outputted to a table:
  - Analyzed fields
    - `Reference_Key, Reference_Value, and Target_Key`
  - Calculated fields
    - `Unconditional_Entropy, Conditional_Entropy, and Entropy_Improvement`
  - Summary fields
    - `Description, Support`

`associate` Command Options

If you specify fields, only specified fields are used; others are not

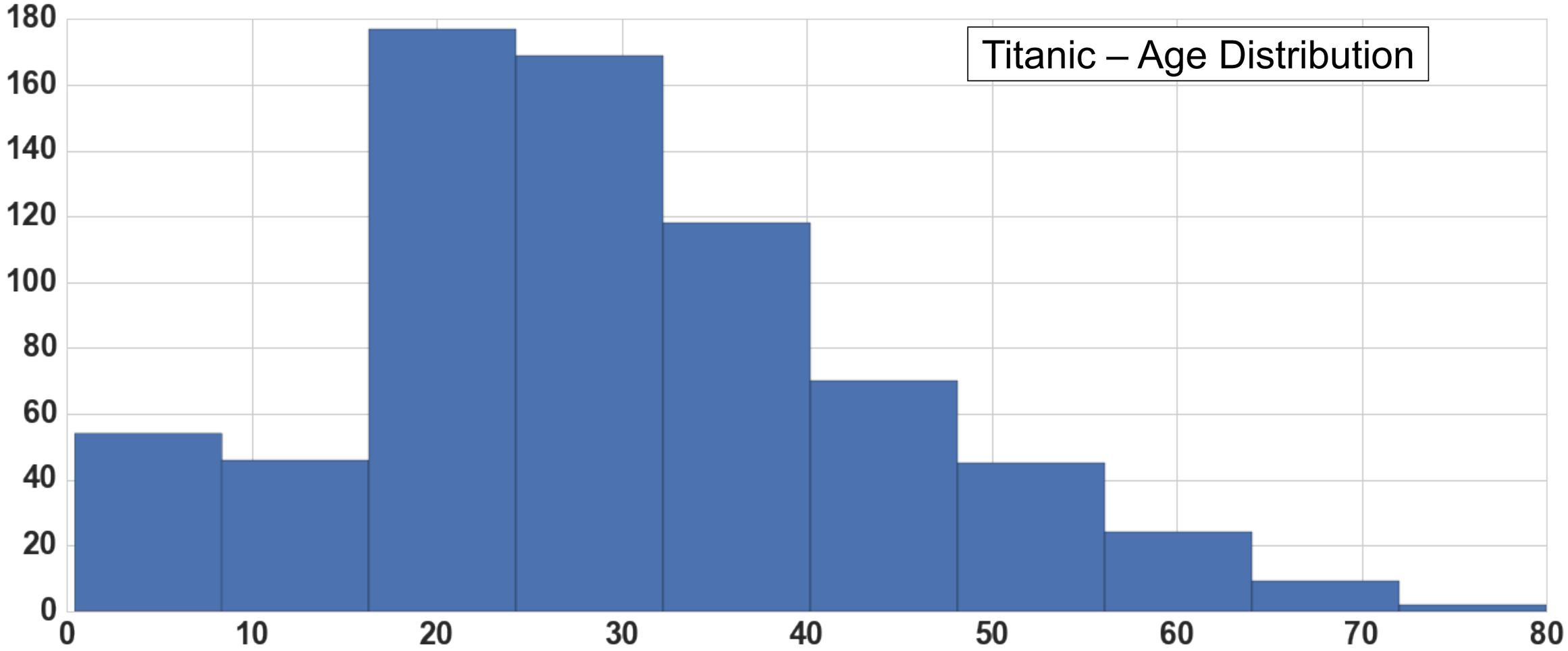
- **`supcnt`** minimum number of times that the "reference key=reference value" combination must appear
  - Default is 100
- **`supfreq`** minimum frequency of "reference key=reference value" combination as a fraction of the number of total events
  - Default is 0.1
- **`improv`** minimum entropy improvement for the "target key" outputted field
  - Default is 0.5

`associate` Command Example

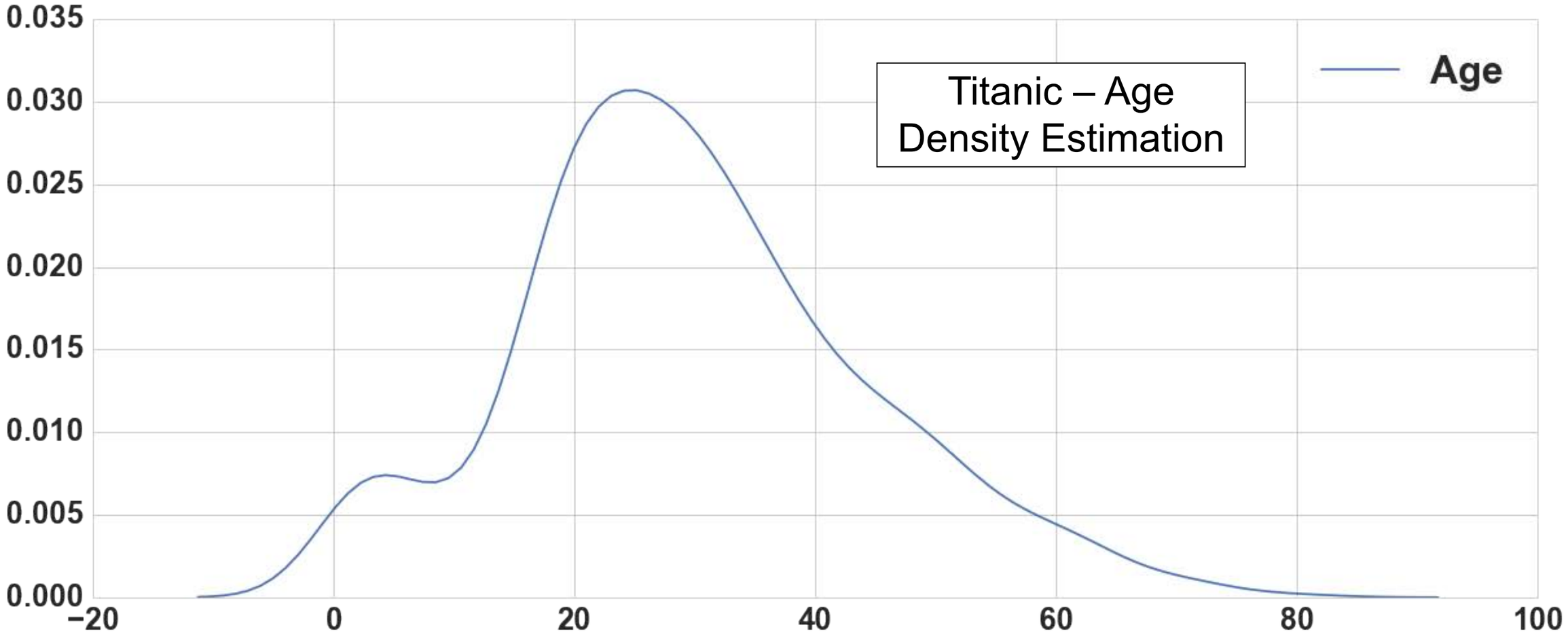
```
| inputlookup phishing.csv
| associate
```

Reference_Key	Reference_Value	Target_Key	Support	Unconditional_Entropy	Conditional_Entropy	Entropy_Improvement	Top_Conditional_Value	Description
Favicon	-1	popUpWidnow	18.57%	0.708	0.193	0.515388	-1 (19.33% -> 97.03%)	When 'Favicon' has the value '-1', the entropy of 'popUpWidnow' decreases from 0.708 to 0.193.
Favicon	1	popUpWidnow	81.43%	0.708	0.119	0.589348	1 (80.67% -> 98.39%)	When 'Favicon' has the value '1', the entropy of 'popUpWidnow' decreases from 0.708 to 0.119.
Favicon	1	port	81.43%	0.573	0.040	0.533061	1 (86.41% -> 99.57%)	When 'Favicon' has the value '1', the entropy of 'port' decreases from 0.573 to 0.040.
Prefix_Suffix	1	Result	13.25%	0.991	0.000	0.990624	1 (55.69% -> 100.00%)	When 'Prefix_Suffix' has the value '1', the entropy of 'Result' decreases from 0.991 to 0.000.
Prefix_Suffix	1	SSLfinal_State	13.25%	1.329	0.365	0.964834	1 (57.27% -> 93.04%)	When 'Prefix_Suffix' has the value '1', the entropy of 'SSLfinal_State' decreases from 1.329 to 0.365.
Result	-1	Prefix_Suffix	44.31%	0.564	0.000	0.564307	-1 (86.75% -> 100.00%)	When 'Result' has the value '-1', the entropy of 'Prefix_Suffix' decreases from 0.564 to 0.000.
Result	1	SSLfinal_State	55.69%	1.329	0.442	0.887088	1 (57.27% -> 91.44%)	When 'Result' has the value '1', the entropy of 'SSLfinal_State' decreases from 1.329 to 0.442.
Result	1	URL_of_Anchor	55.69%	1.508	1.000	0.508669	0 (48.28% -> 62.29%)	When 'Result' has the value '1', the entropy of 'URL_of_Anchor' decreases from 1.508 to 1.000.

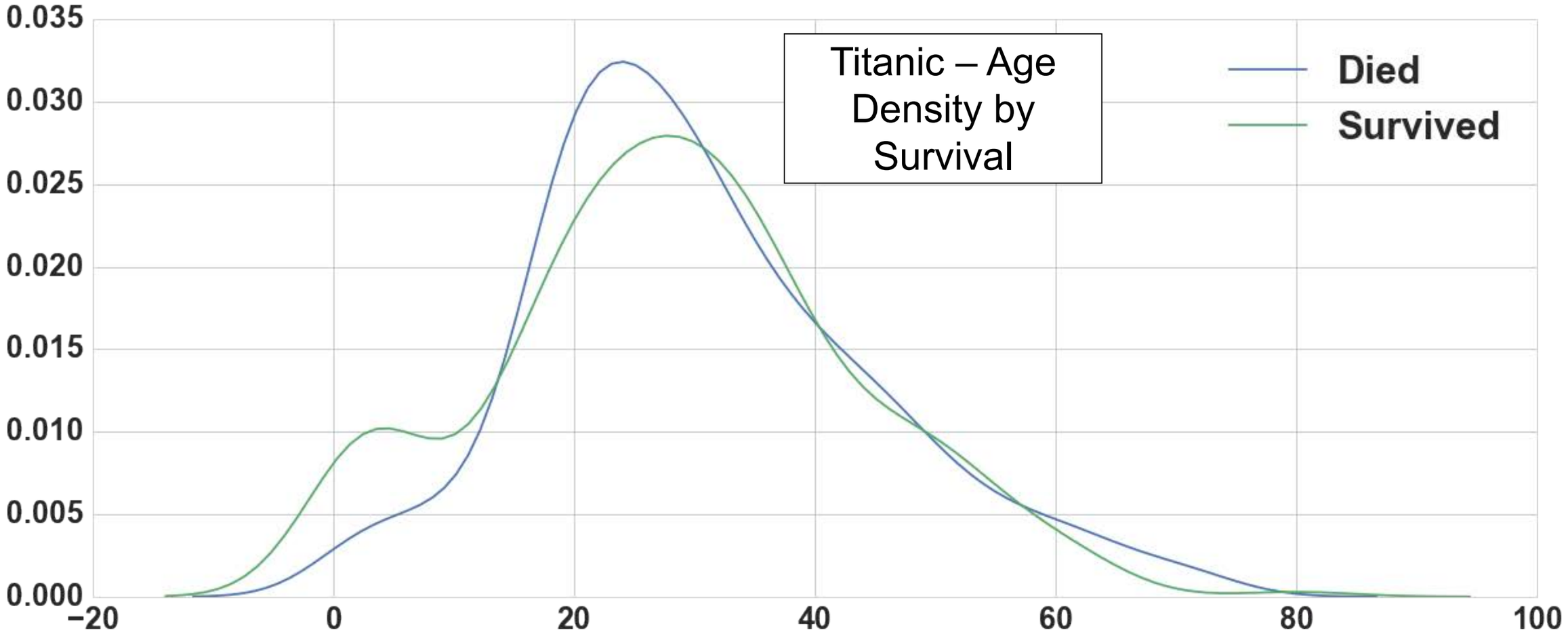
Normal Approximations - Age Distribution



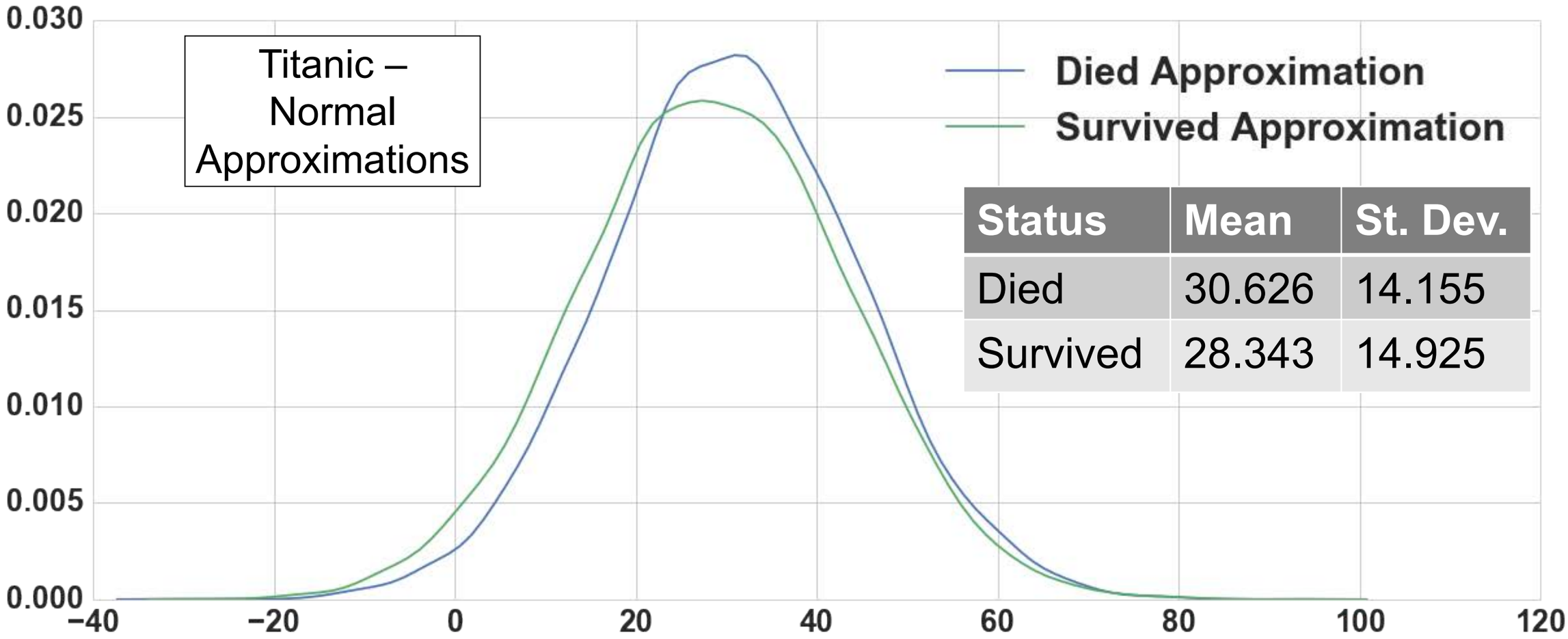
Normal Approx. - Age Density Estimation



Normal Approx. - Age Density by Survival



Normal Approximation – Titanic Data



Status	Mean	St. Dev.
Died	30.626	14.155
Survived	28.343	14.925

Filtering Results

- `search` command
  - May be easier because you are familiar with basic search syntax
  - Treats field values in a case-insensitive manner
  - Can use the * (asterisk) as wildcard
  - Allows searching on keyword
- `where` command
  - Can compare values from two different fields
  - Can do a wildcard search on multiple characters (`%`) or simply on one character (`_`); must use the like operator with wildcards
  - Functions are available, example `isnotnull()`
  - Field values are case-sensitive
- `REGEX`

eventstats & streamstats

Command	Generates	for
eventstats	summary statistics	global, all events
streamstats	summary statistics	specifiable window of each event at the time it is seen by Splunk

bytes	SumOfBytes	clientip
1354	1354	217.23.14.61
1391	4840	211.166.11.101
972	4840	211.166.11.101
2477	4840	211.166.11.101

```
sourcetype=access_combined*
| head 4
| sort _time
| eventstats sum(bytes)
as SumOfBytes by clientip
| table bytes, SumOfBytes, clientip
```

_time	bytes	SumOfBytes	clientip
2018-06-09 17:31:50	3848	3848	200.6.134.23
2018-06-09 17:32:07	604	4452	200.6.134.23
2018-06-09 17:32:09	3651	3651	193.33.170.23
2018-06-09 17:32:15	2333	5984	193.33.170.23

```
sourcetype=access_combined*
| head 4
| sort _time
| streamstats sum(bytes)
as SumOfBytes by clientip
| table bytes, SumOfBytes, clientip
```

foreach Command

For each field in a wild-carded field list * (www1, www2, www3), **foreach** runs a templatized streaming sub-search

Scenario

Display yesterday's hourly volume in MB for each web server.

_time	www1	www2	www3
2018-06-08 17:00	0.23 MB	0.25 MB	0.16 MB
2018-06-08 18:00	0.17 MB	0.24 MB	0.23 MB
2018-06-08 19:00	0.20 MB	0.27 MB	0.24 MB
2018-06-08 20:00	0.23 MB	0.27 MB	0.18 MB
2018-06-08 21:00		0.25 MB	0.26 MB
2018-06-08 22:00		0.19 MB	0.30 MB
2018-06-08 23:00		0.26 MB	0.20 MB
2018-06-09 00:00	0.24 MB	0.16 MB	0.40 MB

```
sourcetype=access_combined
| timechart span=1h A sum(bytes) by host B
| foreach * B [eval <<FIELD>> =
  round(<<FIELD>>/(1024*1024),2)." MB"] C
```


`xyseries` Command

```
xyseries <x-field> <y-name-field> <y-data-field>
```

- `<x-field>` is the field to use as the x-axis
- `<y-name-field>` is the field that contains the values to be used as labels for the data series
- `<y-data-field>` is the field(s) containing the data to be charted
- Generally, instead of **`xyseries`**, use **`chart a over b by c`**
- Which is equivalent to **`stats a by b,c | xyseries b c a`**
- For processing after **`chart`**, use **`stats`** then **`xyseries`**
- **`untable`** is the opposite of **`xyseries`**

EDA Commands Quick Ref

- `kmeans` clusters numeric fields
- `cluster` command/Patterns tab clusters textual data (by patterns)
- `correlate` co-occurrence ratios between fields in a matrix format
- `associate` : determine if one field's value informs us of another field's values
- `analyzefields` finds fields most predictive of the value of a specified field (ML Tooklit has `FieldSelector` algorithm for this)
  - `bin`
  - `makecontinuous`
  - `fieldsummary`
  - `foreach`
  - `findkeywords`
  - `xyseries`

Module 2 Lab Exercise – Begin Analyzing Data

Time: 15 – 20 minutes

Tasks:

- Use the `Patterns` tab and the `cluster` command to create groups of events
- Use the `associate` command to find relationships among fields' values in your data

Module 3: Machine Learning Workflow

Module Objectives

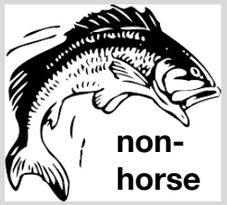
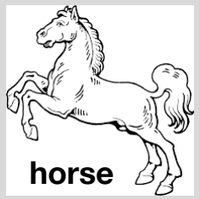
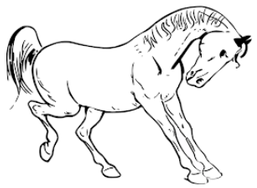
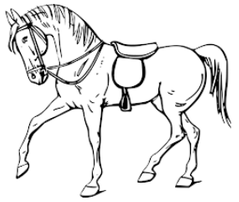
- Define some concepts and terms associated with machine learning
- Model data using machine learning
- Split data to train and test models
- Use Machine Learning Toolkit commands

Features and Samples

Splunk Term	ML Term
row, event, line	sample
column, field, attribute	feature, parameter

id	feature1	feature2	feature3
1	244	1	16
2	2093	1	2
3	254	1	51
4	1988	1	30
5	2080	1	25
6	1851	1	11
7	3060	1	54
8	3065	1	44
9	3699	1	24
10	882	1	4

Supervised & Unsupervised ML

	Supervised Learning	Unsupervised Learning
When provided with	labeled examples   	nothing
After many trial-and-error loops with feedback, the model learns to classify	 horse  horse  non-horse  non-horse  non-horse  horse	

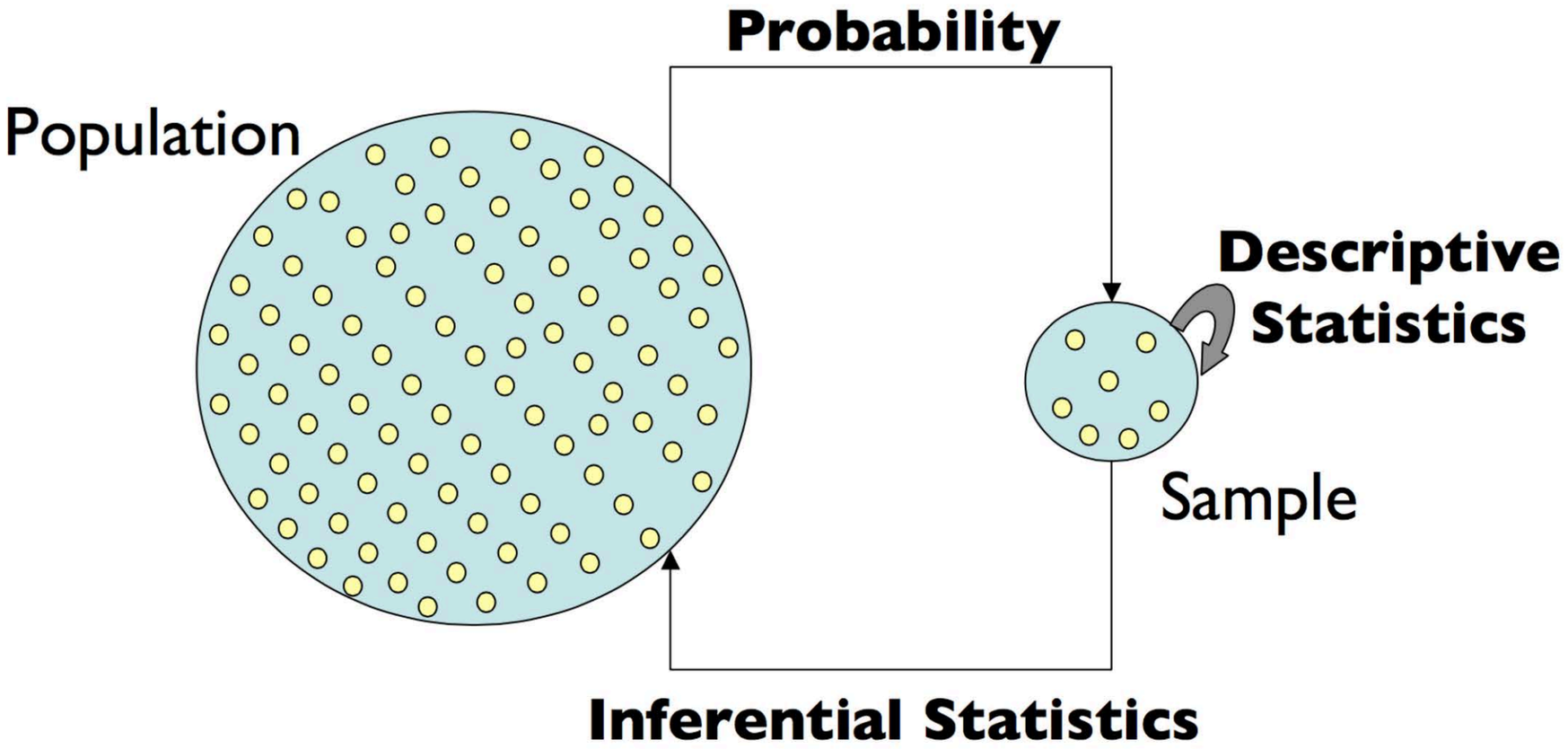
Machine Learning Workflow

1. Prepare (clean) the data
2. Split data (set aside some to train the model, some to test its performance)
3. Fit the model on some of the data (training data)
4. Apply the model on data the model hasn't seen (test data)
5. Validate the performance of the model on the test data
6. Refine the model based on sources of error
7. Repeat as needed until the margin of error is low enough

BUILD MODEL

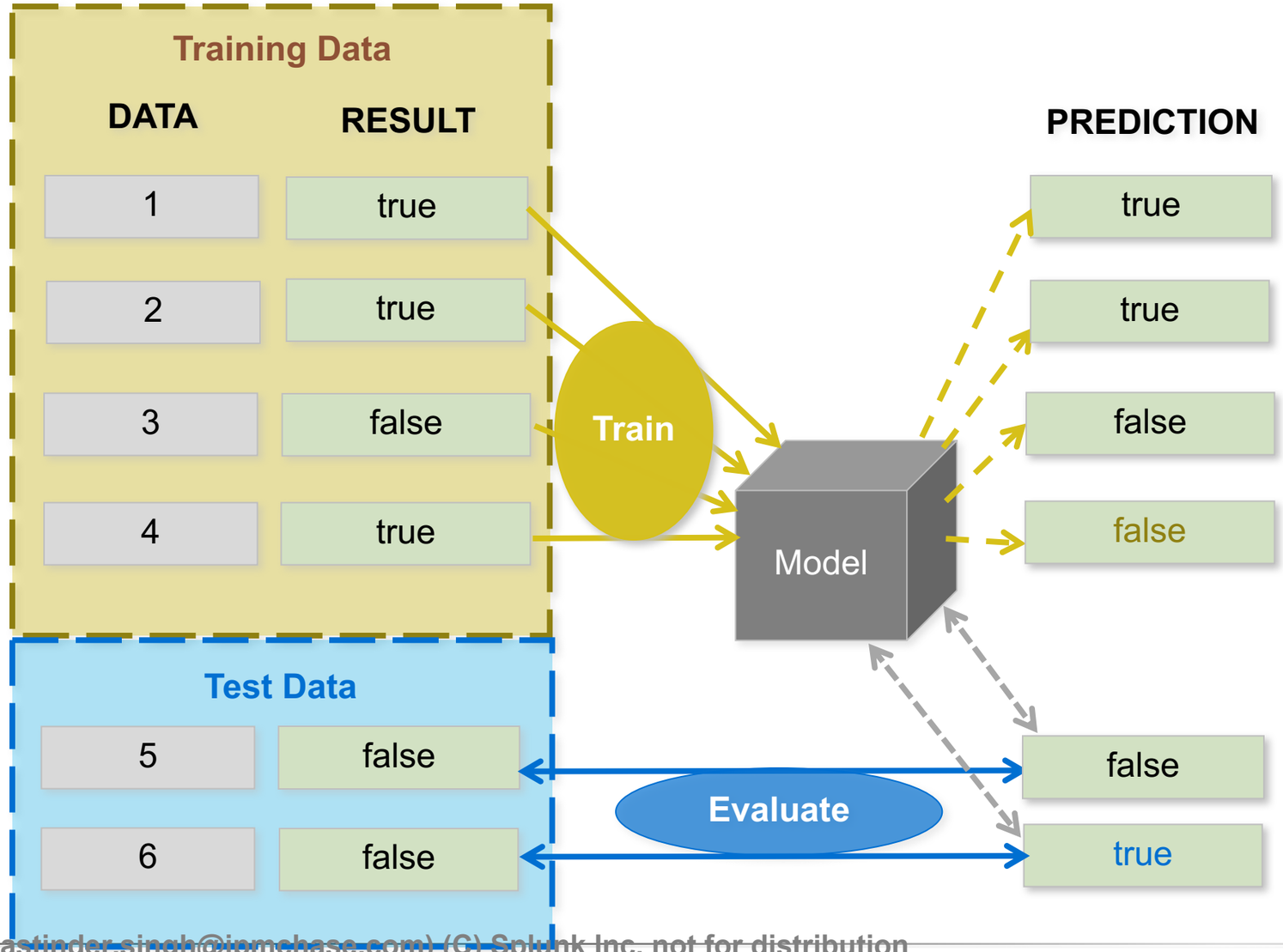
8. Deploy (use) the model to answer real world business questions

Machine Learning ~ Statistical Inference



Train / Test Split

- Data subset to test model
  - Random sample
  - Not the data it was trained on
  - There is no one rule on which percentages to use for splitting
- Splitting with Splunk
  - Use `sample`
  - Use time to split data
  - Try a combination



`sample` Command vs. Event Sampling

Sample Command	Event Sampling
Samples after results are collected from the indexes	Samples before results are collected from the indexes
Uses the entire search pipeline	Used only at the beginning of the search pipeline
Includes modes Event Sampling does not: - partitioning - biased sampling - can specify an exact number of results - reproducible results	faster
Not native to Splunk Enterprise	Native to Splunk Enterprise

`sample` Command Sampling Modes

- Ratio: when you want an approximation; a float between 0 and 1
  - The probability each event has of being included in the result set
  - 0.01 ratio: % probability for each event of being included in the results
- Count: exact number of randomly-chosen events to return
  - If sample count > total events in the search, all events are returned
- Proportional: Each event is sampled with a probability specified by this field value; yields a biased sampling

```
sourcetype=access_combined price=*
| eventstats max(price) as max min(price) as min range(price) as range
| eval price_proportion = if(min=max, price, (price - min) / range)
| top price_proportion limit=0
| addtotals col=t row=f percent label=proportion labelfield=price_proportion
```


`sample` Command Partitioning Mode

Partitions can divide your results into groups for different purposes

- Such as using some results for testing and some for training
- You decide the number of partitions in which to randomly divide events
  - The split is approximate
- For **`ratio`** and **`count`**, **`keywords`** can be omitted
  - `| sample 0.01` is equivalent to `| sample ratio=0.01`
  - `| sample 10` can be used in place of `| sample count=10`

`sample` Command Syntax

```
sample (ratio=<float between 0 and 1>)?
(count=<positive integer>)?
(proportional=<name of numeric field> (inverse)?)?
(partitions=<natural number greater than 1>
(fieldname=<string>)?)?
seed=<number> choose a number to ensure reproducible results
by <split_by_field>)?
```


`sample` Command Examples

- Partition events into 7 groups, with the chosen group returned in a field called "partition_number"

```
... | sample partitions=7 fieldname="partition_number"
```

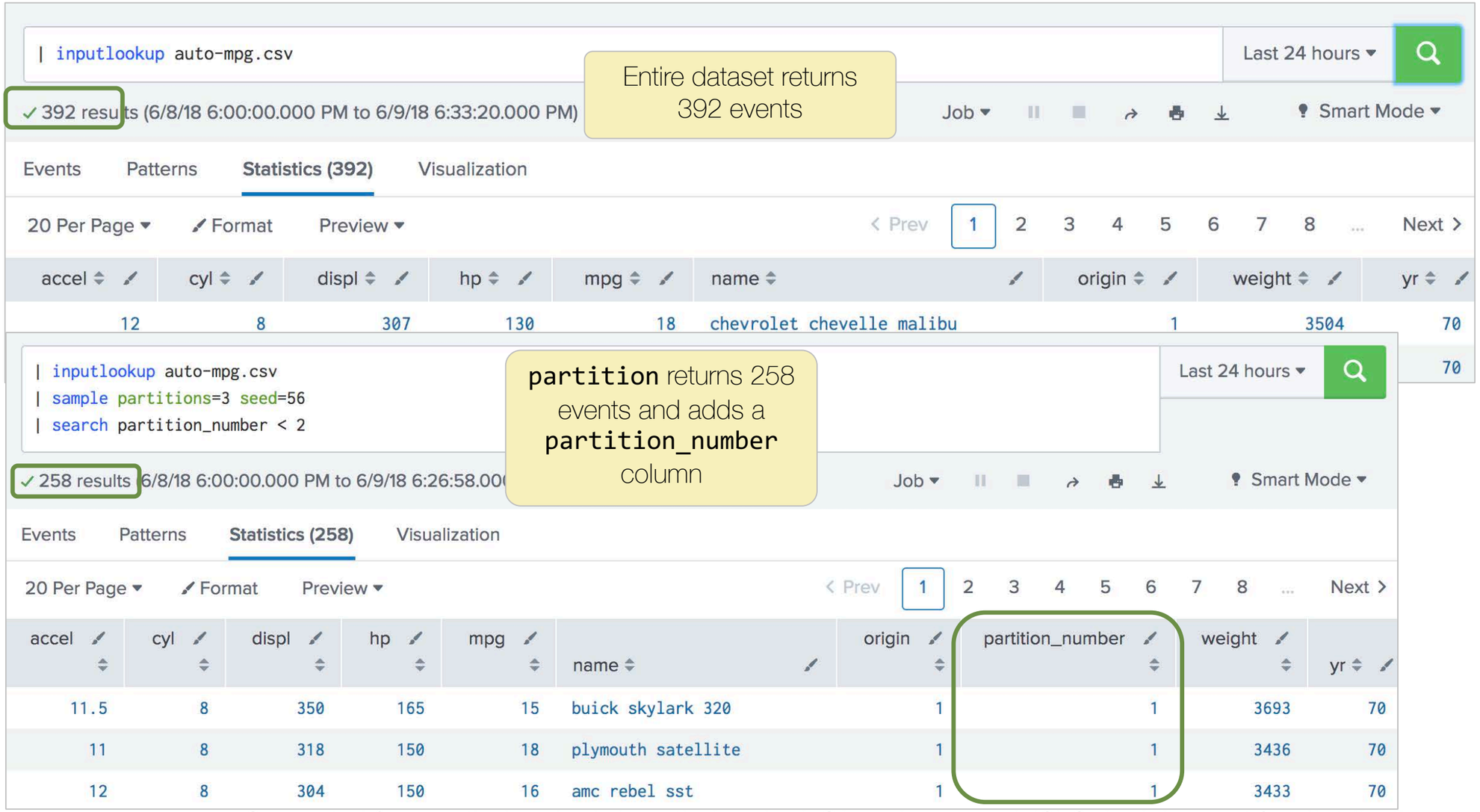
- Retrieve exactly 20 events at random from each host

```
... | sample count=20 by host
```

- Return each event with a probability determined by the value of "some_field"

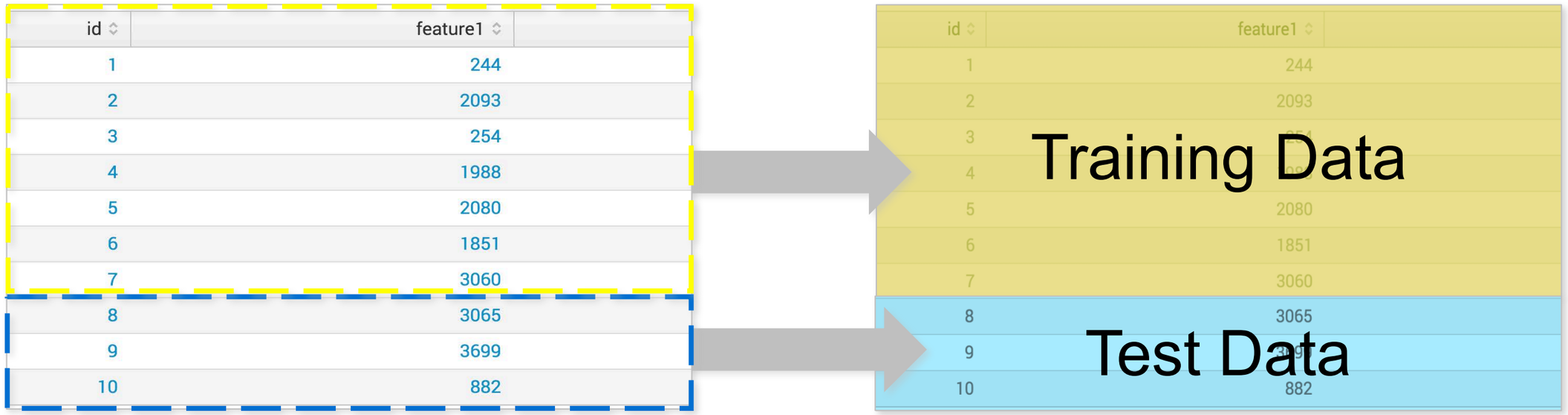
```
... | sample proportional="some_field"
```


seed and partition Example



Splitting Data by Time

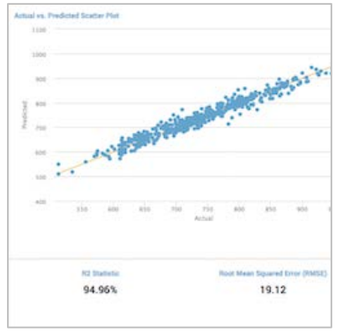
- Here is an example of how it might look to split data by time in Splunk
- Data split randomly would show many small arrows crossing each other



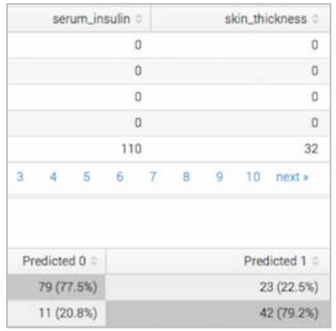
Splunk Machine Learning Toolkit

- Commands, visualizations, experiments and examples
- Also called “ML Toolkit” or ”ML App”
- Also, ML-SPL Performance App https://splunkbase.splunk.com/app/3289/

1	Clean the data
2	Fit the model
3	Validate the model
4	Refine the model
5	Deploy the model



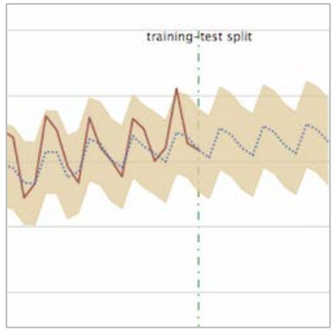
Predict Numeric Fields
Server Power Consumption
VPN Usage
Median House Value
Power Plant Energy Output



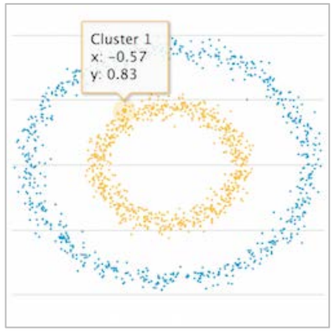
Predict Categorical Fields
Hard Drive Failure
Presence of Malware
Telecom Customer Churn
Presence of Diabetes
Vehicle Make and Model

Detect Numeric Outliers
Server Response Time
Number of Logins vs. Predicted
Supermarket Purchases
Power Plant Humidity

Detect Categorical Outliers
Disk Failures
Bitcoin Transactions
Supermarket Purchases
Mortgage Contracts
Diabetes Patient Records
Mobile Phone Activity

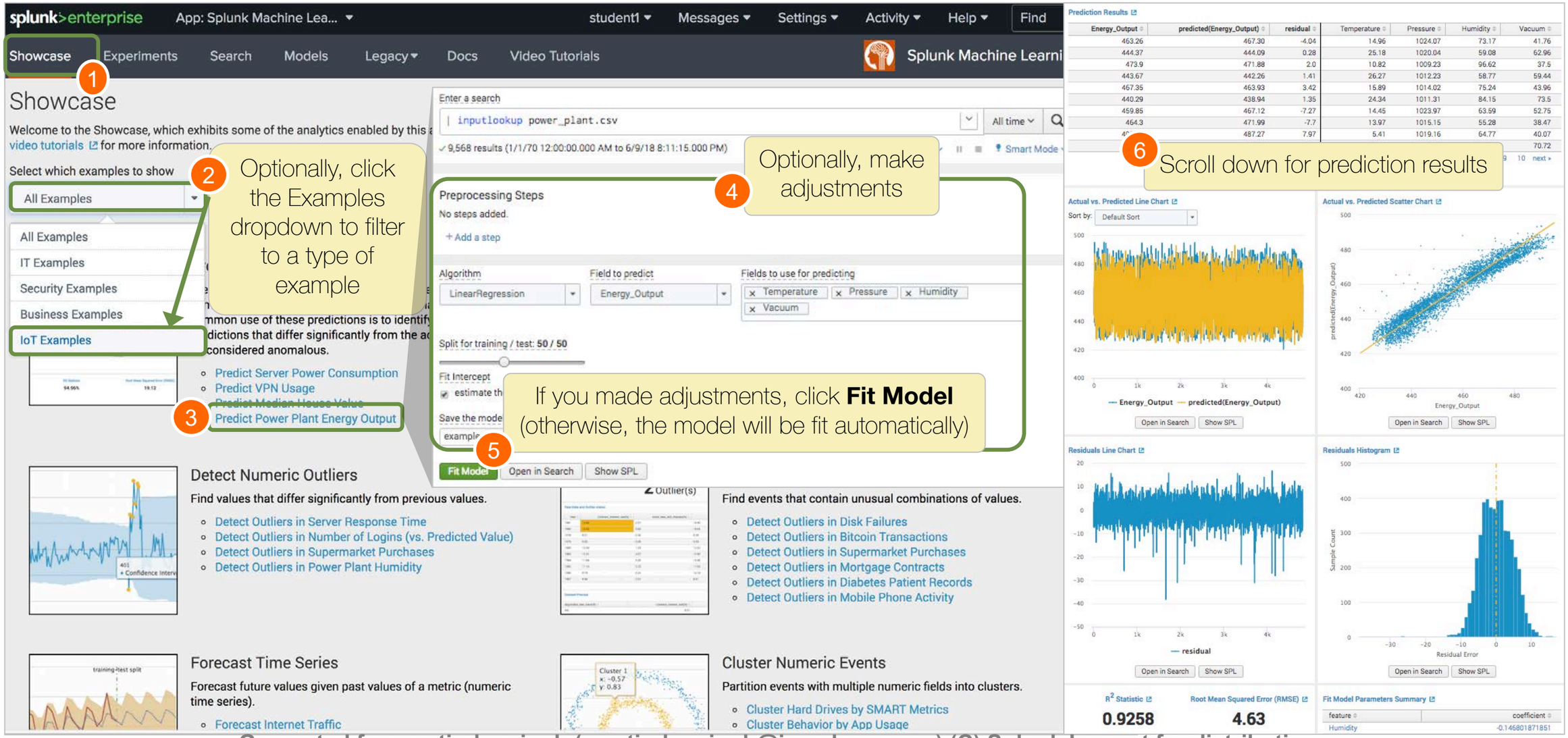


Forecast Time Series
Internet Traffic
Number of Employee Logins
Monthly Sales
Number of Bluetooth Devices
Exchange Rate TWI (ARIMA)



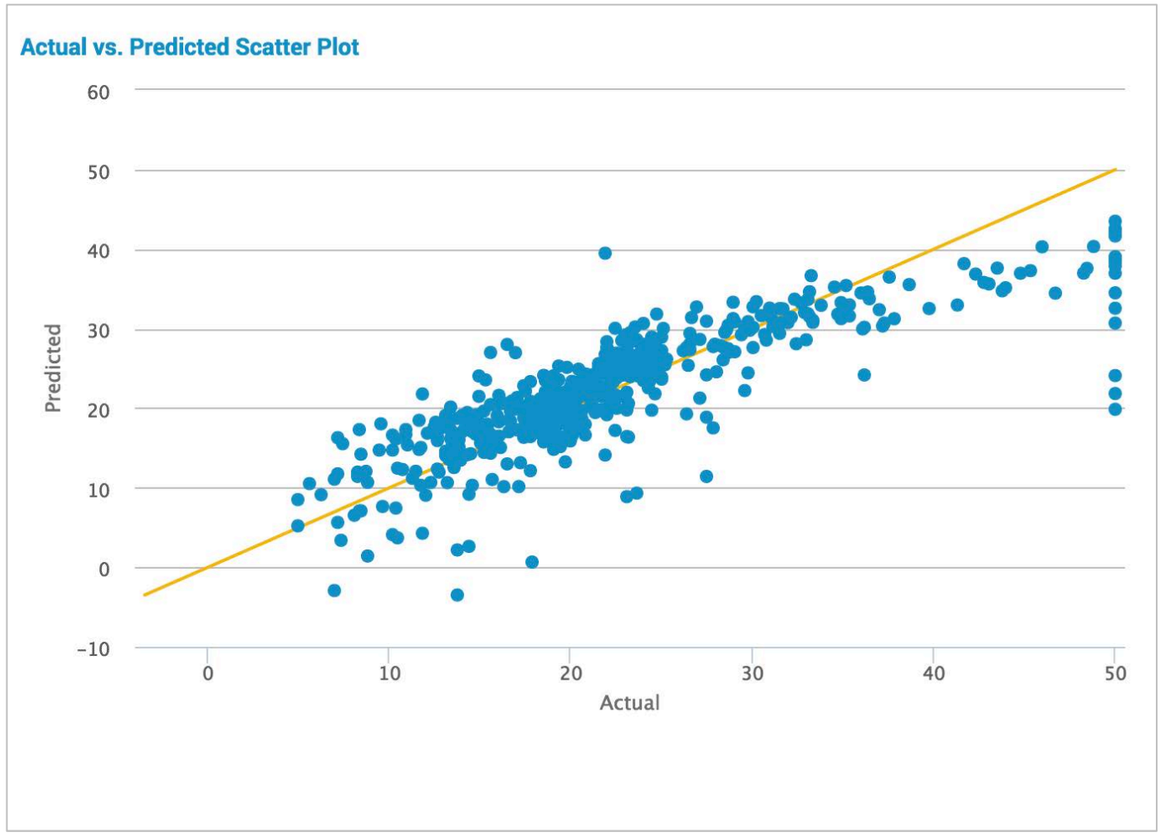
Cluster Numeric Events
Hard Drives by SMART Metrics
Behavior by App Usage
Neighborhoods by Properties
Vehicles by Onboard Metrics
Power Plant Operating Regimes

ML Toolkit Showcase: Pre-built Examples



`fit` Command

1. Start with an algorithm that:
   - Takes in a set of parameters
   - Returns a predicted data set
2. Use an error function to provide a number representing the difference between your data and the model's prediction for any given set of model parameters
3. Find the parameters that minimize this difference



`fit` Requirements

```
| fit <algorithm> <field to predict>
from <feature1> <feature2> into <model>
```

- `<algorithm>` **required** and case sensitive
- `<field to predict>` only required by supervised learning algorithms, such as `LinearRegression, LogisticRegression, SVM`
- `<feature>` fields (at least one is required)
- `<model>` model name, only required if you want to save the model

```
| fit Kmeans <fields> [into <model name>] [k=<int>]
[random_state=<int>]
```


`fit` Examples

```
| inputlookup auto-mpg.csv | fit LinearRegression mpg as mpg(predicted) from hp
```

- **`from`** clause separates predicted field from its features
- Creates a new field called **`predicted(mpg)`**

```
| inputlookup auto-mpg.csv | fit LinearRegression mpg from hp into mpg_model
```

- **`into`** saves the results into a model
- You specify the name of the model, in this case **`mpg_model`**

```
| inputlookup auto-mpg.csv | fit KMeans k=4 hp
```

- In this example, a value for **`k`** is specified
- Creates a new field called **`cluster`**

`fit` Example with `LinearRegression`

```
| inputlookup auto-mpg.csv
| sample partitions=3 seed=71
| search partition_number < 2
| fit LinearRegression mpg from hp weight
into mpg_model_LinearRegression
```

accel	cyl	displ	hp	mpg	name	origin	partition_number	predicted(mpg)	weight	yr
12.0	8	307.0	130	18.0	chevrolet chevelle malibu	1	0	19.1456945109	3504	70
11.5	8	350.0	165	15.0	buick skylark 320	1	1	16.5970055946	3693	70
11.0	8	318.0	150	18.0	plymouth satellite	1	0	18.7547142219	3436	70
9.0	8	454.0	220	14.0	chevrolet impala	1	0	10.3884544357	4354	70
8.5	8	440.0	215	14.0	plymouth fury iii	1	0	10.8433551288	4312	70
10.0	8	455.0	225	14.0	pontiac catalina	1	0	9.75800267476	4425	70
8.5	8	390.0	190	15.0	amc ambassador dpl	1	0	14.6433368394	3850	70
10.0	8	383.0	170	15.0	dodge challenger	1	0	17.1833043386	3563	70

`fit` Example with `KMeans`

```
| inputlookup auto-mpg.csv
| sample partitions=3 seed=56
| search partition_number < 2
| fit KMeans mpg hp weight
into auto_model_KMeans
```

accel	cluster	cluster_distance	cyl	displ	hp	mpg	name	origin	partition_number	weight	yr
11.5	0	2582.79471111	8	350.0	165	15.0	buick skylark 320	1	1	3693	70
11.0	6	25801.8847222	8	318.0	150	18.0	plymouth satellite	1	1	3436	70
12.0	6	24896.0847222	8	304.0	150	16.0	amc rebel sst	1	1	3433	70
9.0	3	12771.0004844	8	454.0	220	14.0	chevrolet impala	1	1	4354	70
8.5	3	5834.17695502	8	440.0	215	14.0	plymouth fury iii	1	0	4312	70

`apply` Command

- Use the apply command to
  - Use a saved model
  - Compute predictions for the current search results
    - Based on a model learned by the **`fit`** command
- Used on different search results than those used when fitting
  - However, results should have an identical list of fields
- Available on all algorithms, except:
  - **`DBSCAN`**
  - **`SpectralClustering`**

```
| apply  <model_name> (as output_field)
```


`summary` Command

```
| summary <model_name>
```

Returns a summary of a machine learning model learned using `fit`

- The summary it generates is algorithm specific
- Some algorithms don't support `summary`

Algorithm	Summary
Linear regression	List of coefficients
Logistic regression	List of coefficients for each class

coefficient	feature
-0.0401308637844	hp
-0.00605348509944	weight
45.5741185913	_intercept

`listmodels` & `deletemodel` Commands

- Returns a list of machine learning models learned using `fit`
- The algorithm and arguments given when fit was invoked are displayed for each model
- Models may be deleted with | `deletemodel <model_name>`
- Deletes a machine learning model learned using fit
- It's a good idea to verify before deleting
  - To get a list of all models, use | `listmodels`
- For more on ML commands

  http://docs.splunk.com/Documentation/MLApp/latest/User/Customsearchcommands

Saving Models

- Models are not automatically saved (save them with **`fit into`**)
  - It often takes several versions before finding a model to use going forward
  - Models also take up space in Splunk
- If you adjust your model after using **`apply`**
  - Remember to use the into clause to save the updated version
  - Use the same name if you want to overwrite the old version
- Standardize on version naming conventions for your organization
- To check to prevent overwriting, use | **`listmodels`** to verify

`fit` & `apply` Configurations

$SPLUNK_HOME/etc/apps/Splunk_ML_Toolkit/default/mlspl.conf	Upgrading - save a copy of mlspl.conf with only the modified stanzas and settings in $SPLUNK_HOME/etc/apps/Splunk_ML_Toolkit/local/
`max_inputs`	(default: `100,000`) maximum number of events used in `fit`
max_distinct_cat_values	(default: `100`) distinct values in a field used in one-hot encoding
max_distinct_cat_values_for_classifiers	(default: `100`) distinct values in a categorical field that's the target (output)
`use_sampling`	(default: `true`)when `max_inputs` is exceeded, downsample? `true: fit` downsamples its input using Reservoir Sampling `false: fit` returns an error message
`max_fit_time`	(default: `600`) maximum time, in seconds, to spend in the "fit" phase Does not relate to the other phases of a search
`max_memory_usage_mb`	(default: `1000`) maximum allowed memory usage, in megabytes, by the `fit` command
`max_model_size_mb`	(default: `15`) max. model size, in megabytes, created by `fit` `SVM` & `RandomForest` may create models large enough to impact performance with bundle replication

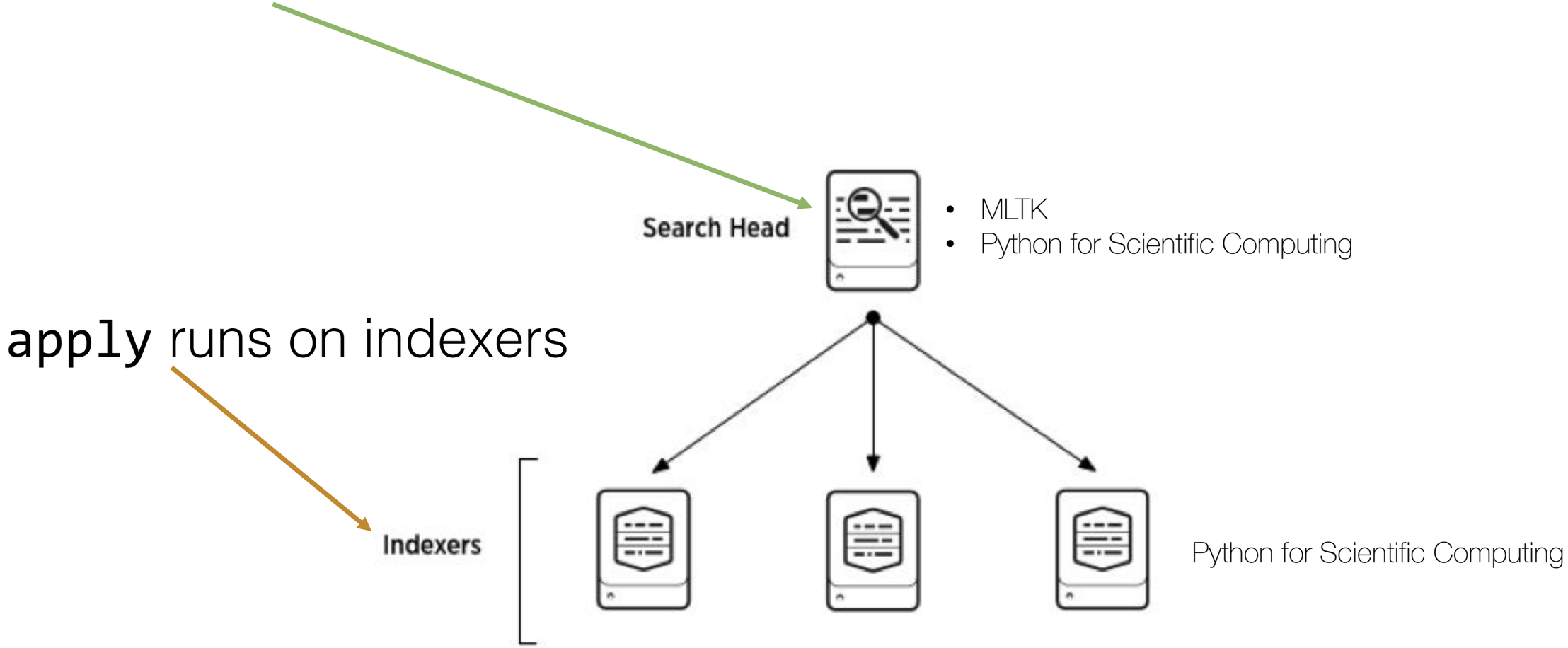
Distributed Option: `streaming_apply`

- Configure indexers to run the apply command
- Install Python for Scientific Computing add-on on all indexers in the indexing cluster
- On each search head in your deployment:
  - Open `$SPLUNK_HOME/etc/apps/Splunk_ML_Toolkit/local/mlspl.conf` configuration file in a text editor
  - Create the `mlspl.conf` in the local directory if one does not exist
- Copy the [default] stanza from `$SPLUNK_HOME/etc/apps/Splunk_ML_Toolkit/default/mlspl.conf` to the local version of the configuration file if this stanza is not present.
- Change streaming_apply to true: `streaming_apply = true`

Architecture: `streaming_apply`

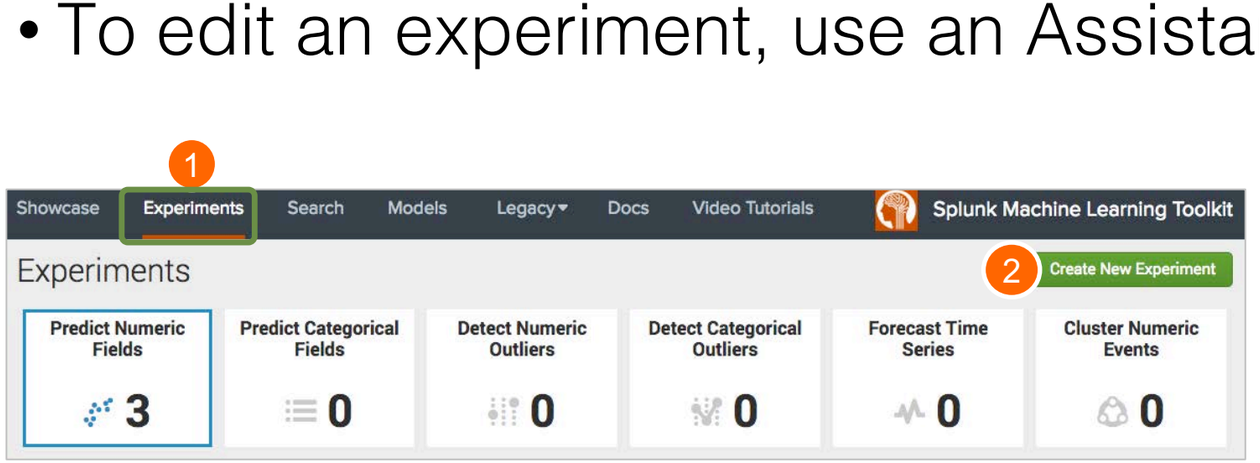
`fit` runs on the search head

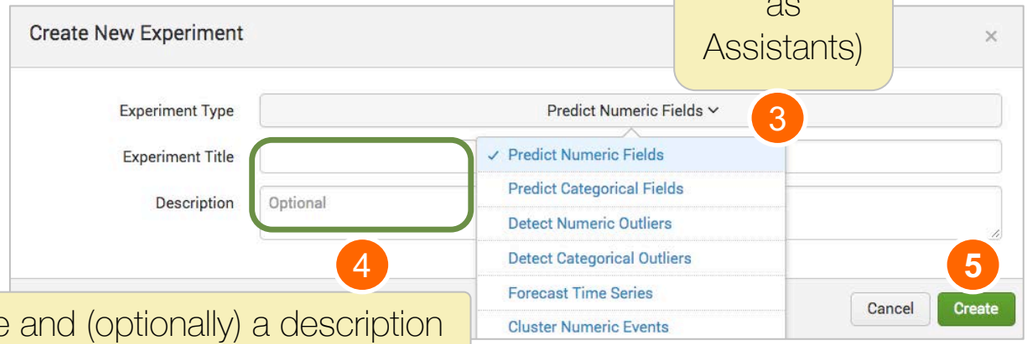
`apply` runs on indexers



MLTK Experiments

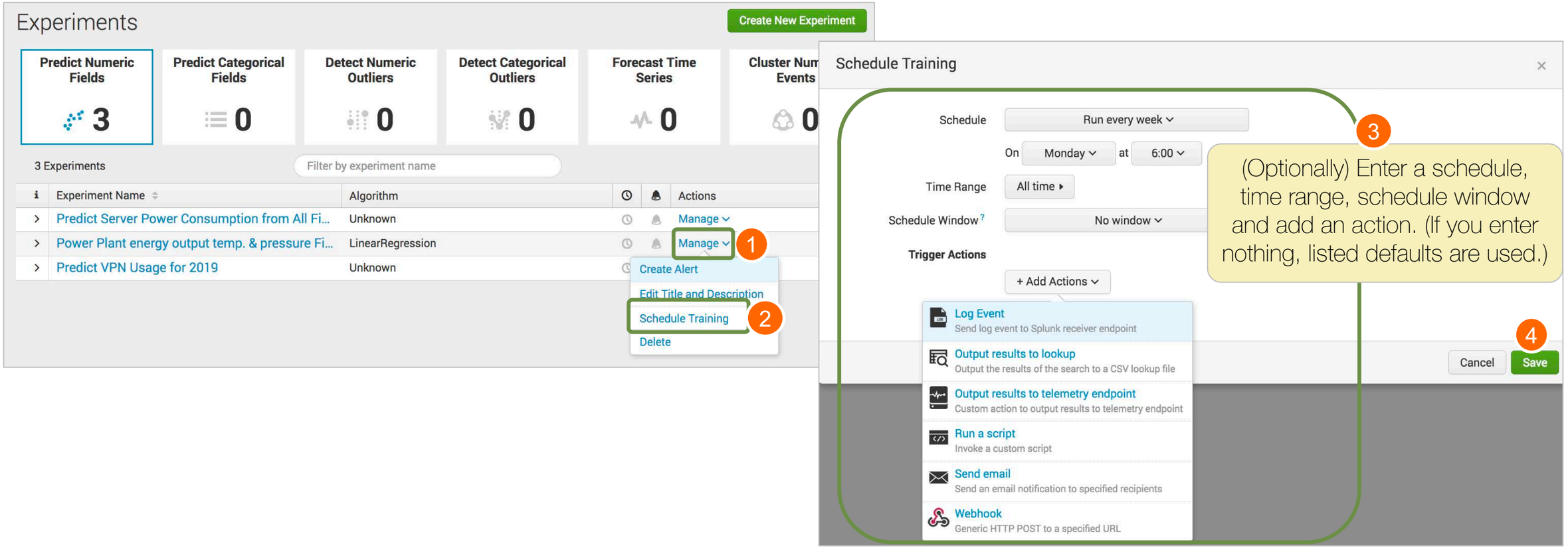
- A workspace in the MLTK that tracks its settings, history, alerts and scheduled trainings
- Each experiment is a single framework that manages its own:
  - Data source
  - Algorithm used
  - Additional parameters to configure that algorithm
- To edit an experiment, use an Assistant



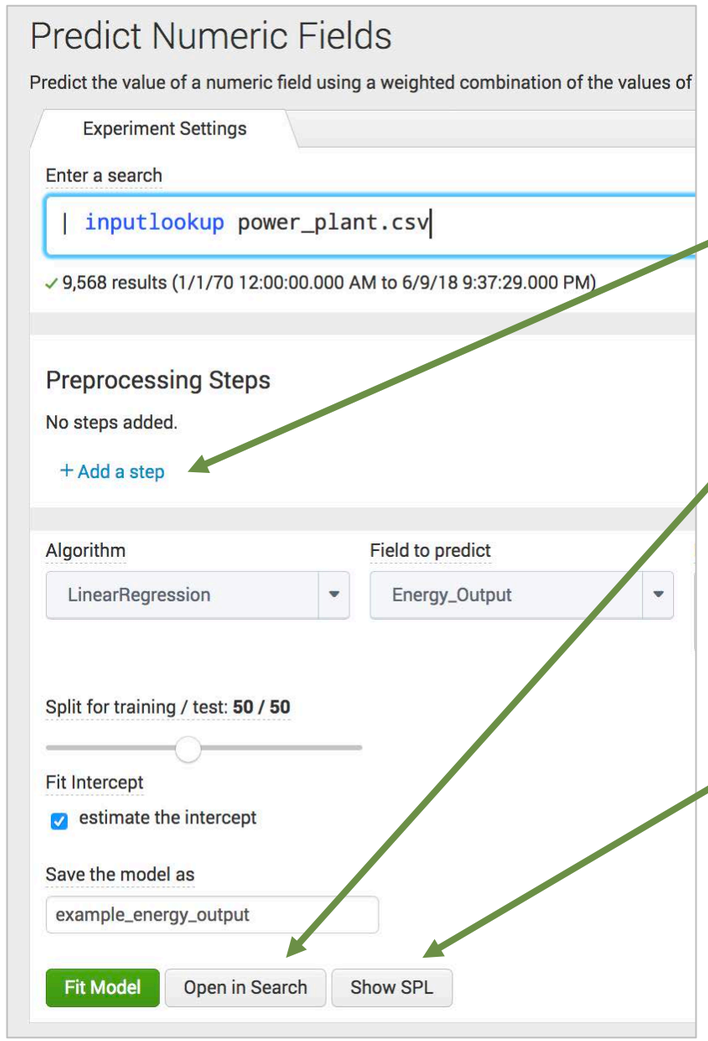


Scheduled `fit`

After you validated and refined an experiment, you can set it to fit (train) at set intervals to keep it current as new data arrives



Working with Searches



To work with the search under the model

- Preprocess the data if needed
- Click **Open in Search** to open a new Search tab populated with the same search, using all data (instead of just the training set)
- Click **Show SPL** to see the search used to fit the model

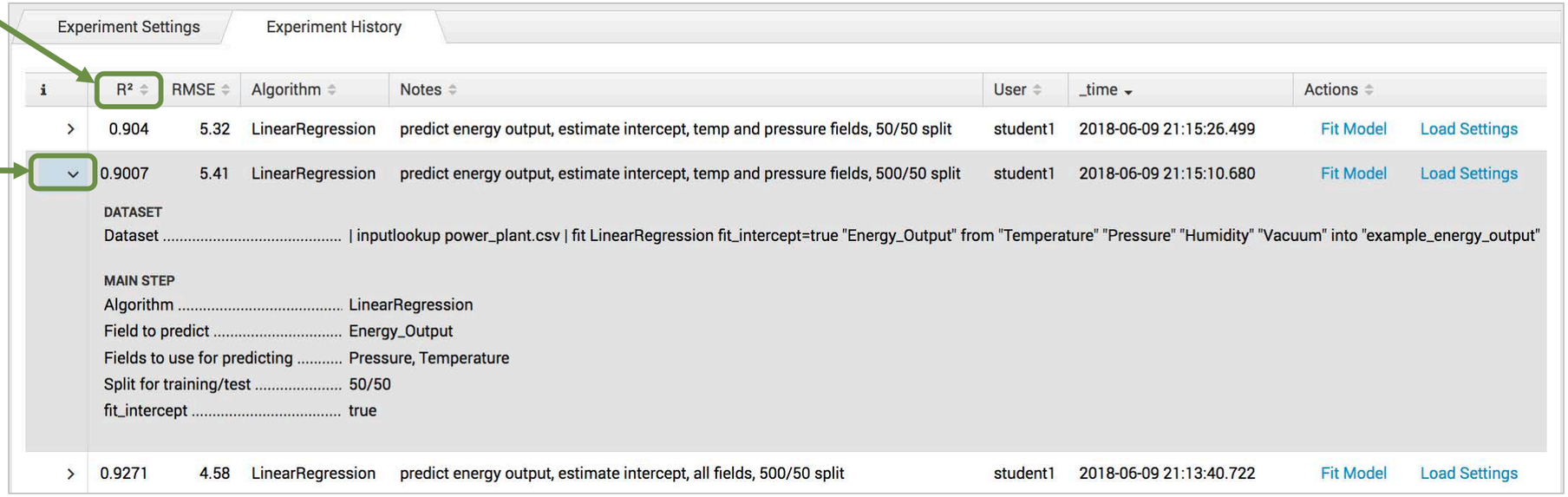
Refining Models: Load Existing Settings

Click the **Experiment History** tab and click **Load Settings** to open your experiment's search and settings so you can optimize:

- Remove less helpful fields and / or add more fields
- **Fit** the model again and, when you're satisfied, **Save** it

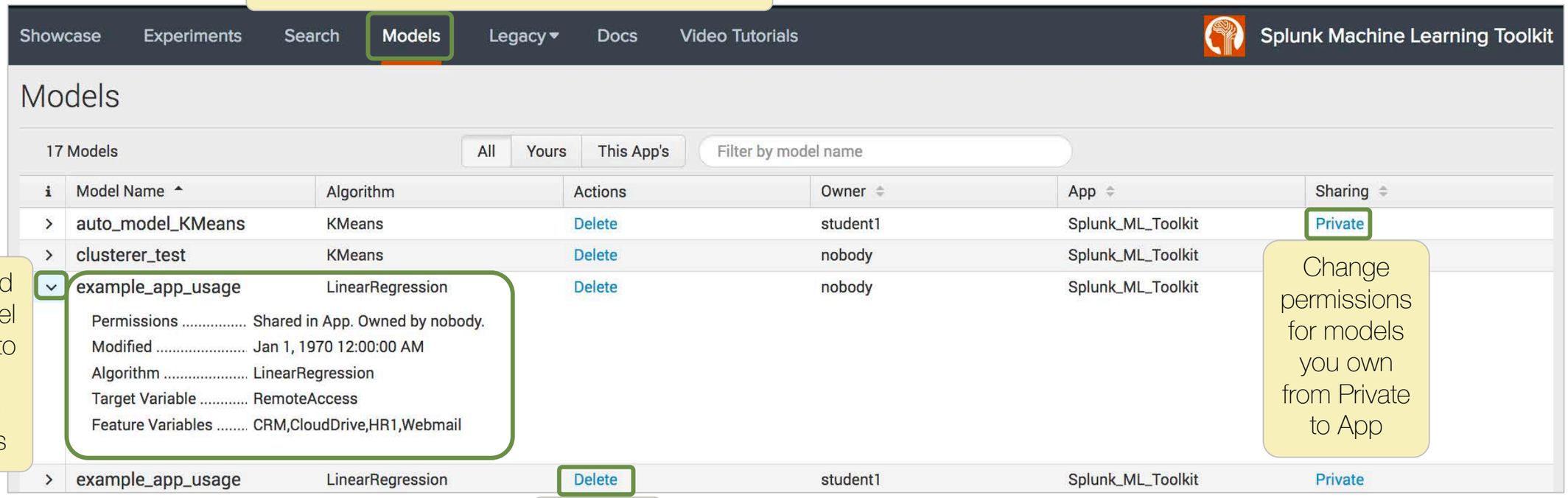
Click to sort by R^2 to see which Experiment worked best

Click > to view details about an Experiment



Managing Models

Click **Models** in the menu bar to view a list of models that you created or were shared



Expand a model listing to view more details

Change permissions for models you own from Private to App

Delete models you own

Module 3 Lab Exercises – Split Data & Explore ML

Time: 20 – 25 minutes

Tasks:

- Split a data set into a training set and a test set using the `sample` command
- Fit, apply, and save a model
- Identify elements of the machine learning workflow
  - Explore the Machine Learning Toolkit Showcases
  - Explore the Machine Learning Toolkit Experiments

Module 4: Algorithms, Preprocessing & Feature Extraction

Module Objectives

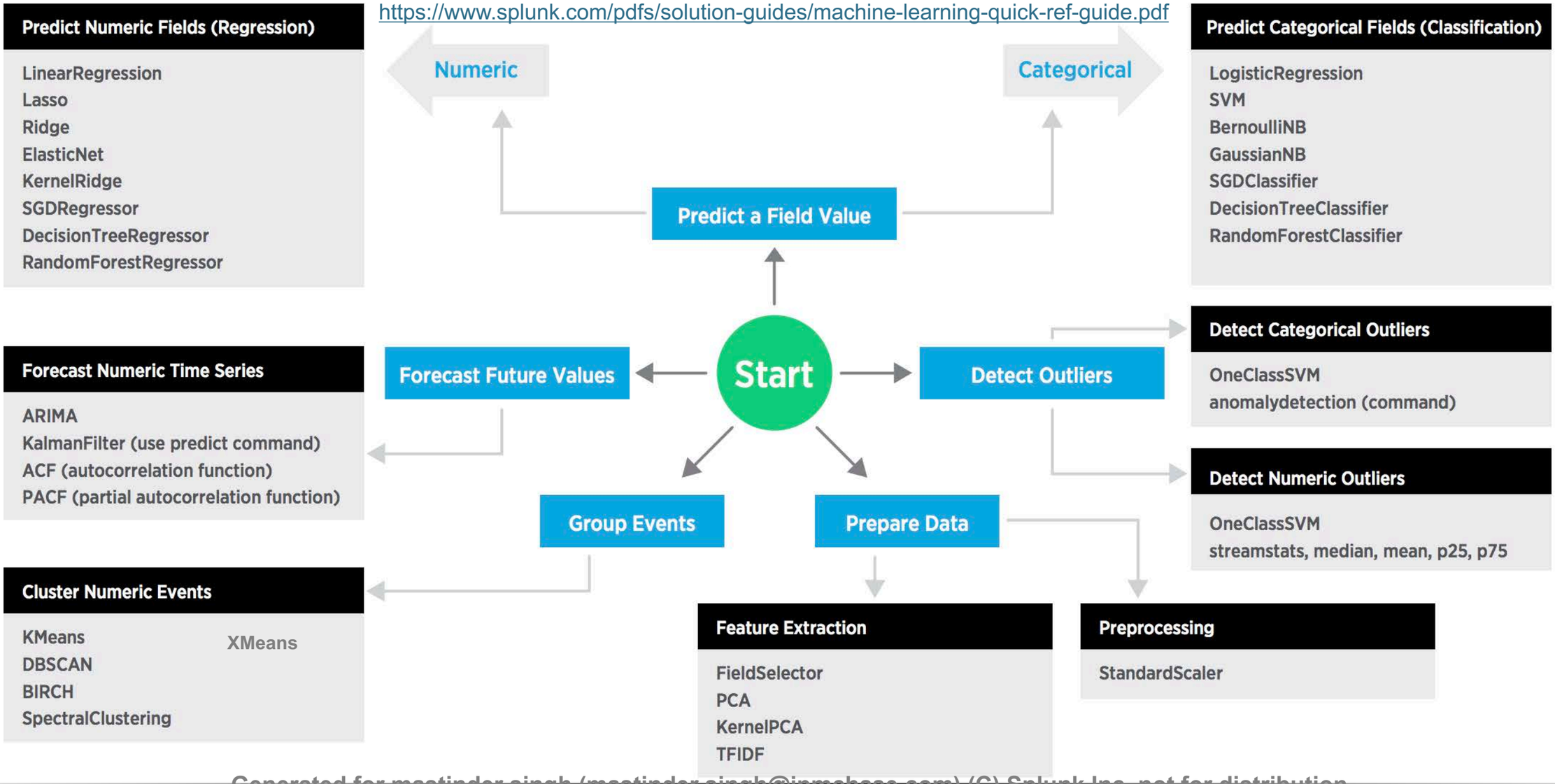
- Choose an algorithm
- Use `FieldSelector` to choose relevant fields
- Use PCA to reduce dimensionality of data
- Scale features
- Normalize data
- Use `StandardScaler` to preprocess data
- Describe a kernel
- Extract features with TF-IDF
- Use partial fit

Choosing an Algorithm

- An algorithm is a procedure or formula for solving a problem
  - Results in a predictable end-state from a known beginning
- Define model objectives, select data, and try some algorithms
  - Clustering: market segmentation or anomaly detection
  - Classification: customer retention or recommender systems
  - Regression: credit scoring or predicting the next outcome of timed events
  - May provide more detailed insight into the business problem
  - Helps identify which variables have predictive power

Algorithm Cheat Sheet

https://www.splunk.com/pdfs/solution-guides/machine-learning-quick-ref-guide.pdf



Algorithm Requirements

Purpose	Algorithm	Field(s) Created

Classification	`LogisticRegression`
	`DecisionTreeClassifier`
	`RandomForestClassifier`
	`SVM`
	`BernoulliNB`
	`GaussianNB`
	`GradientBoostingClassifier`
		`predicted(field)`
		the predicted categorical value e.g. true, false

Regression	`DecisionTreeRegressor`
	`KernelRidge`
	`LinearRegression`
	`RandomForestRegressor`
	`Lasso`
	`ElasticNet`
	`Ridge`
	`GradientBoostingRegressor`
	`SGDRegressor`
		`as`
		the predicted continuous value e.g. 107.8, 42

Time Series Analysis	`ARIMA`	**`predicted(field)`** **`lower95(predicted(field))`** **`upper95(predicted(field))`**

Algorithm Requirements (cont.)

Purpose	Algorithm	Field(s) Created
Anomaly Detection	`OneClassSVM`	**`isNormal`** -1 (anomalous) 1 (normal)

Clustering	`KMeans`
	`DBSCAN`
	`Birch`
	`SpectralClustering`
	`XMeans`
		`cluster`
		A number represents the `clusterID`

Preprocessing	`StandardScaler`	Scaled versions of each field Scaled field names begin with **`SS_`**

Feature Extraction	`FieldSelector`
	`TFIDF`
	`PCA`
	`KernelPCA`
		`PC_1, PC_2, PC_3 … PC_k`
		the projection of the inputs onto their principal components

FieldSelector

- Meta estimator
  - Decides which features are useful (like **analyzefields**)
  - Based on F-tests
  - Estimator agnostic
  - Returns a subset of fields that score well
  - All univariate (one field at a time)
- You specify:
  - Types (cat or num)
  - Modes
  - Parameters (float or int)

```
fit FieldSelector <field_to_predict>
from <explanatory_fields>
[into <model name>]
[type=<'categorical', 'numerical'>]
[mode=<'k_best', 'fpr', 'fdr', 'fwe',
'percentile'>]
[param=<N>]
```


FieldSelector Modes

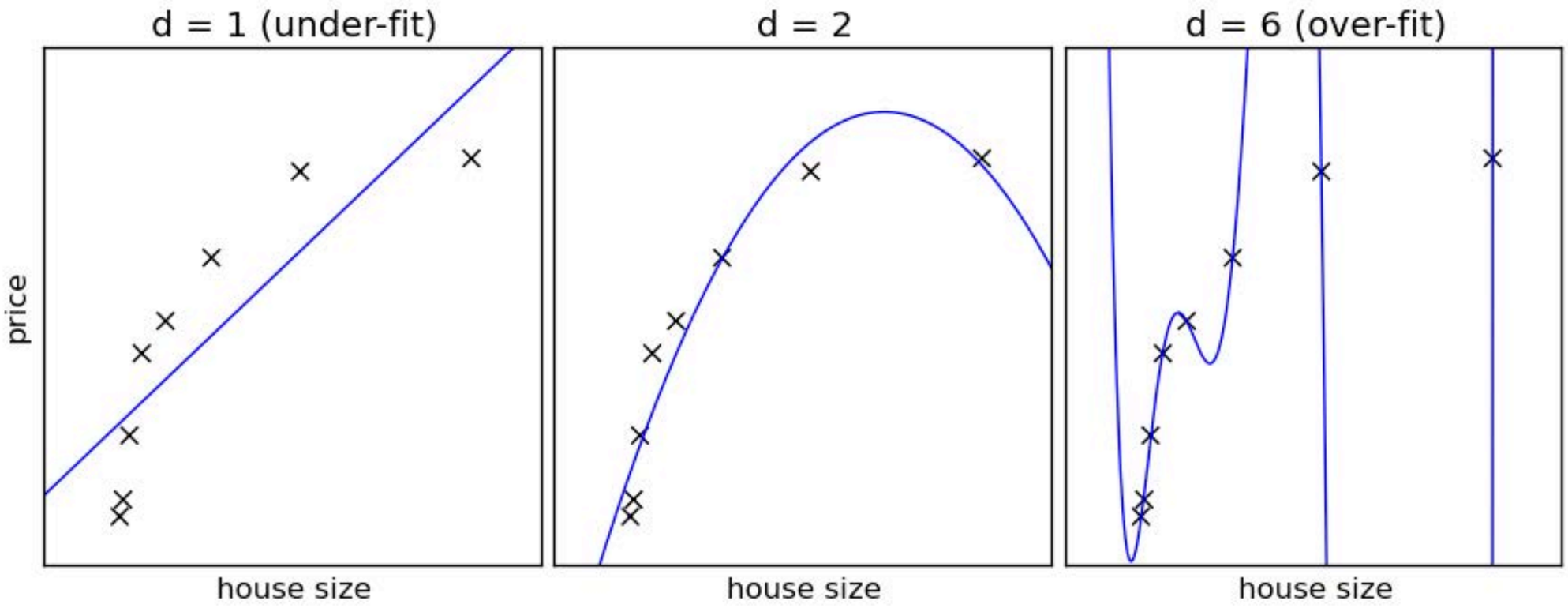
- `mode` - determines parameters for feature selection

`k_best`	removes all but the k highest scores
`fpr`	false positive rate for a single measurement
`fdr`	false discovery rate (false positives) for multiple measurements
`fwe`	family-wise error rate, the probability of making at least one error among all hypotheses (particularly stringent)
`percentile`	removes all but a percentile of the highest scores

- `param`
  - Dependent on feature selection mode
  - When mode is `percentile`, parameter default is `0.05`

Overfitting

An algorithm can under-perform from high bias (underfitting) or high variance (overfitting)



Solutions to Overfitting / Underfitting

Possible solutions to overfitting	Possible solutions to underfitting
Use fewer features	Add more features
Use more training samples	Use fewer training samples (improves speed)
Increase regularization* • lasso (L1) , • ridge (L2), • elasticnet (combo) • alpha	Decrease regularization*
	Use a more sophisticated model
	Add complexity to the model

*Regularization is a technique used to impose simplicity

Preprocessing

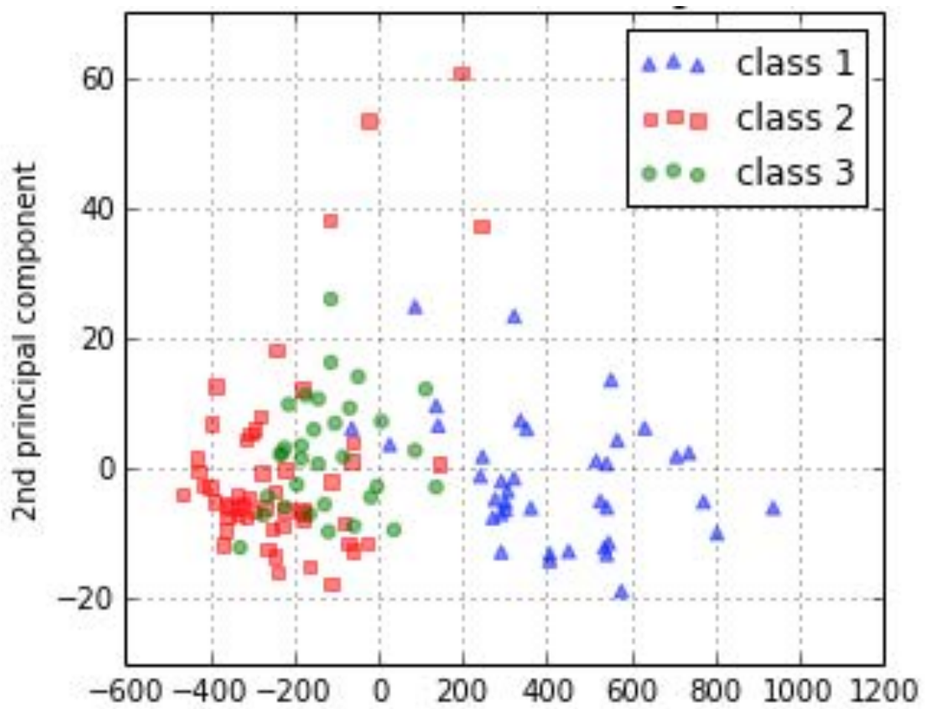
Note

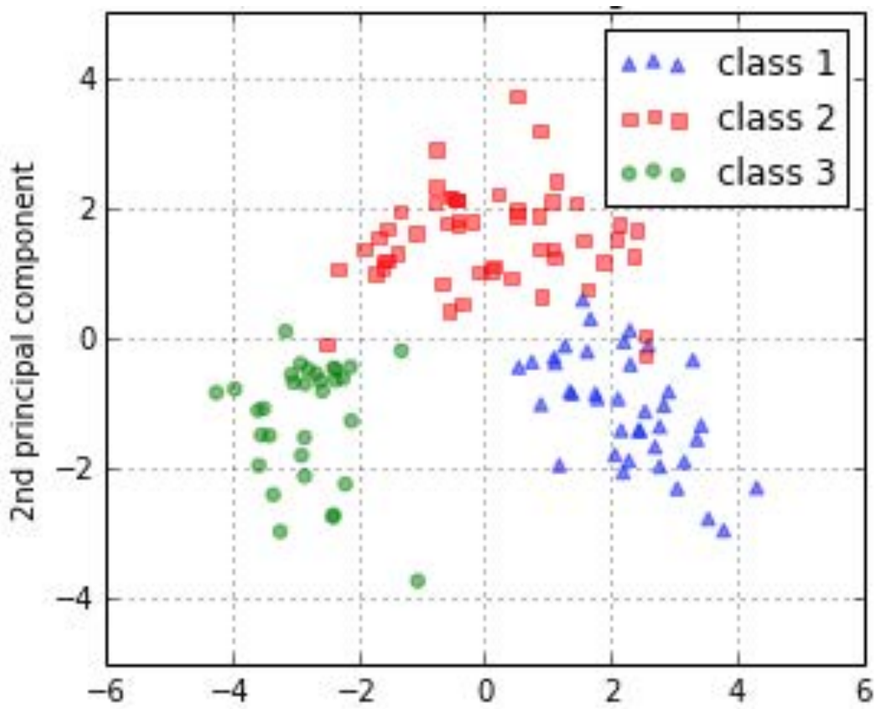
Use StandardScaler before using the KernelPCA method.

- In an Assistant, enter a search, and under **Preprocessing Steps** click **+ Add a step**
- Under **New Preprocess Step**, select the preprocessing method
  - StandardScaler is when the fields have very different scales
  - PCA or KernelPCA to reduce the number of dimensions (fields) for algorithm performance or for visualizations (scatterplot chart)
- Specify one or more fields to preprocess
  - Click for a dropdown or enter field names and use wildcards (*).
- Select any options to use with the selected preprocess method
- Click **Apply** to perform the specified preprocessing

Preparing Data

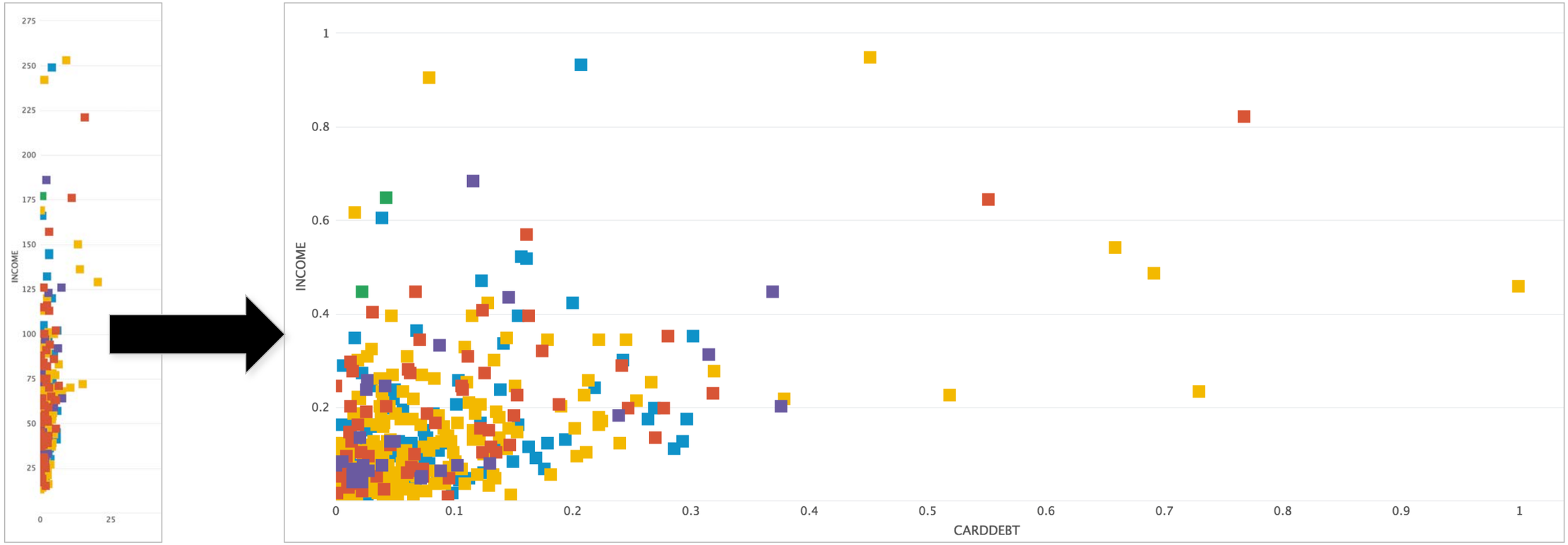
What was changed with this dataset?





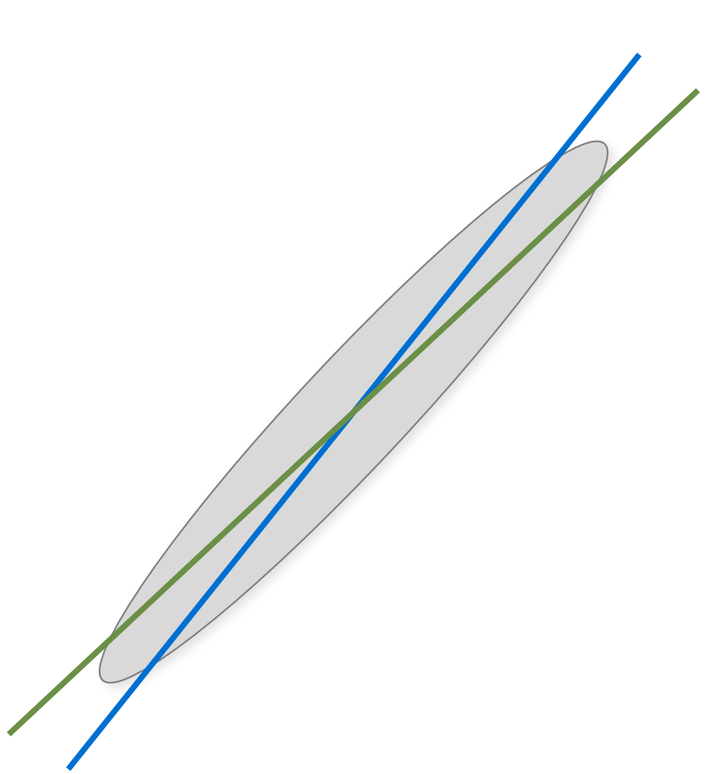
Left example was not normalized before Kmeans + PCA.
Kmeans differentiated the red dots from the green dots

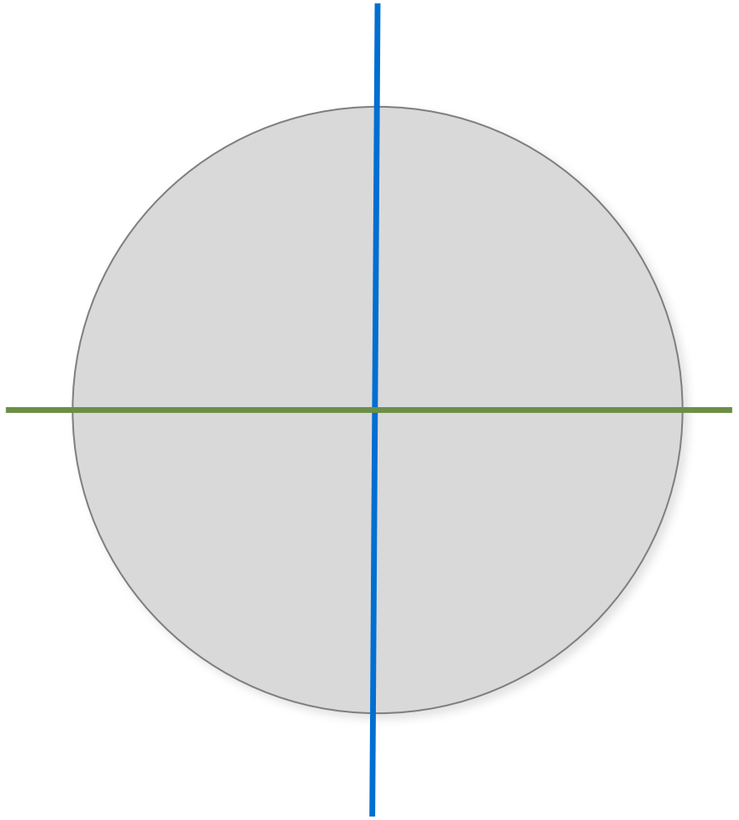
Feature Scaling



Normalization

- Transforming data to render as normal
- `StandardScaler` has a method to normalize data





Normalization Calculations

There are many ways to normalize

Standardization

- Z score is good for kmeans

$$x_z = \frac{x - \mu}{\sigma}$$

x Value
μ Mean of all values
σ Standard deviation from the mean

Rescaling

- 0 mean and unit scaling

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

x' Scaled value
x Original value

`StandardScaler` for Pre-processing

- Both standardizes (with respect to std dev) and scales (with respect to mean)
  - Prevents dominance (biasing) of field(s) over others in subsequent machine learning algorithms
  - Practically required for some algorithms (SVM and clustering algos)
- Adjusts data fields:
  - std. deviations to 1
  - scales means to 0

```
fit StandardScaler <fields> [into <model name>]
[with_mean=<bool>] [with_std=<bool>]
```


`StandardScaler` Preprocessing Options

- **with_mean** to standardize the fields with respect to their mean
- **with_std** to standardize the fields with respect to their standard deviation
- Default: true for both options (if you don't specify anything)
  - To stop one of them from being used, specify it as =false
  - (At least one needs to be *not* set to false, or nothing will happen)

Working with `StandardScaler` Models

- Save `StandardScaler` models using the `into` keyword
- Apply new data later using the `apply` command

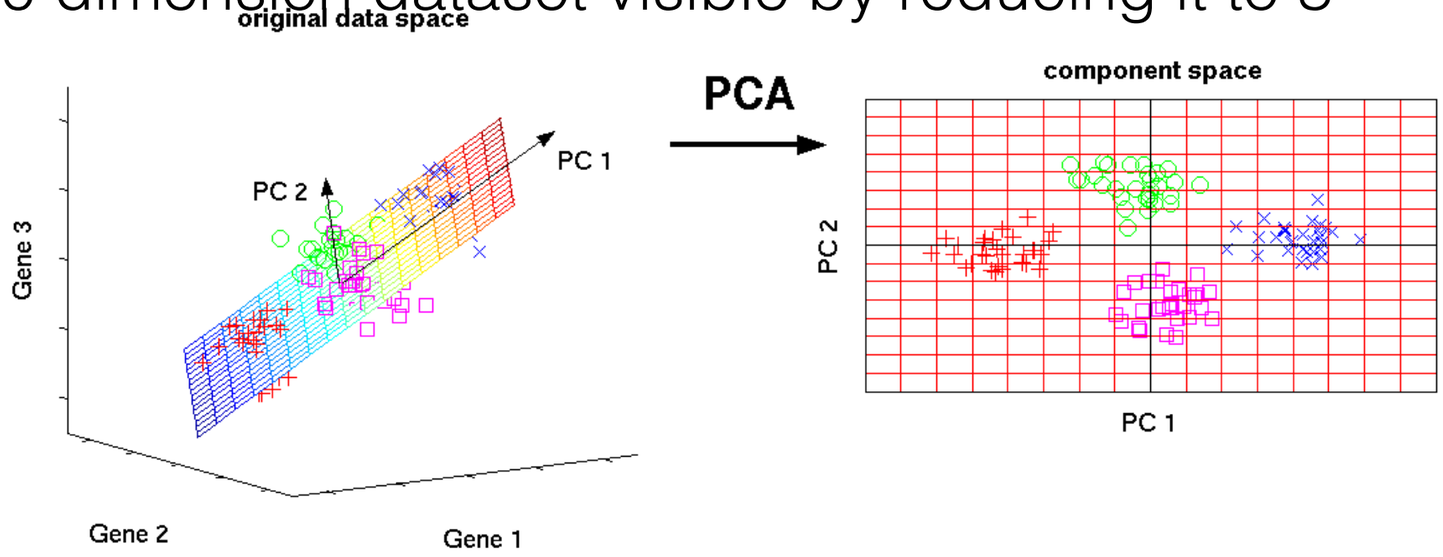
```
... apply scaling_model
```

- View the statistics extracted by `StandardScaler` with the `summary` command

```
| summary scaling_model
```


Principal Component Analysis (PCA)

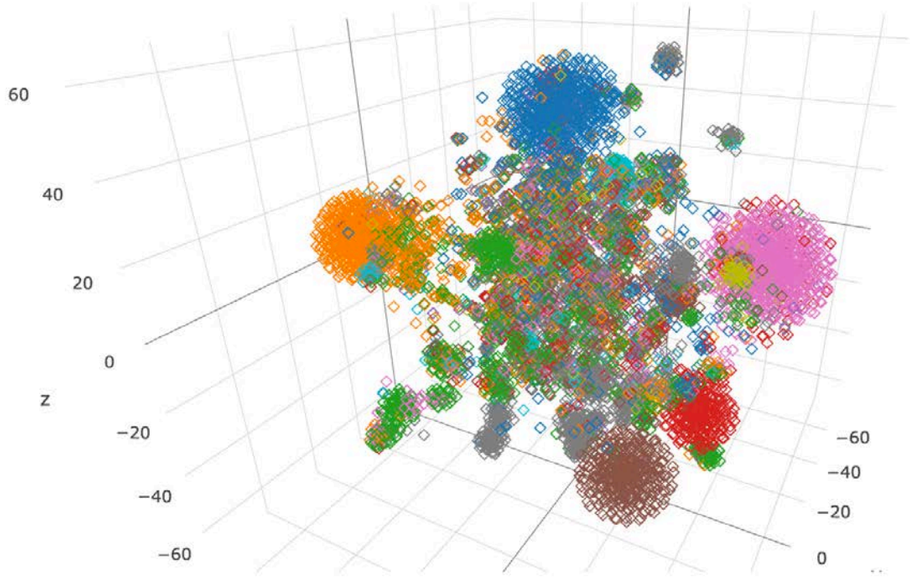
- PCA uses all numeric fields to identify the principal directions in which the data varies to represent data or to reduce its dimensionality
  - For example, make a 1000 dimension dataset visible by reducing it to 3 dimensions (x, y, and z)
  - All features of original dataset are used
  - New features are principal components
  - PCs account for as much of the variability in the data as possible



DOC493: Intelligent Data Analysis and Probabilistic Inference Lecture 15
http://www.doc.ic.ac.uk/~dfg/ProbabilisticInference/IDAPILecture15.pdf

Why Dimensionality Reduction?

- Computational: compress data to provide time/space efficiency
- Statistical: fewer dimensions mean better generalization
- Visualization: grasp structure of data
- Anomaly detection: describe normal data, detect outliers
- **Not** if all components have high variance
- **Not** in classification with high variance within a class (noise variance is higher than the variance between classes)



PCA 17-Dimension Example

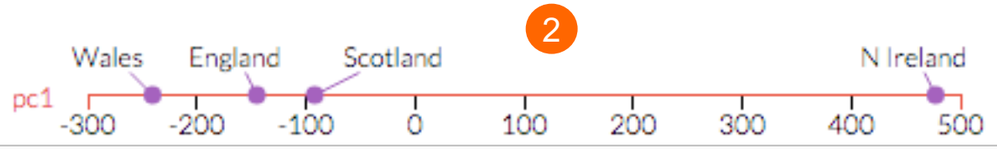
Avg consumption in g/person/ wk of 17 types of food for UK countries

1

	England	N Ireland	Scotland	Wales
Alcoholic drinks	375	135	458	475
Beverages	57	47	53	73
Carcase meat	245	267	242	227
Cereals	1472	1494	1462	1582
Cheese	105	66	103	103
Confectionery	54	41	62	64
Fats and oils	193	209	184	235
Fish	147	93	122	160
Fresh fruit	1102	674	957	1137
Fresh potatoes	720	1033	566	874
Fresh Veg	253	143	171	265
Other meat	685	586	750	803
Other Veg	488	355	418	570
Processed potatoes	198	187	220	203
Processed Veg	360	334	337	365
Soft drinks	1374	1506	1572	1256
Sugars	156	139	147	175

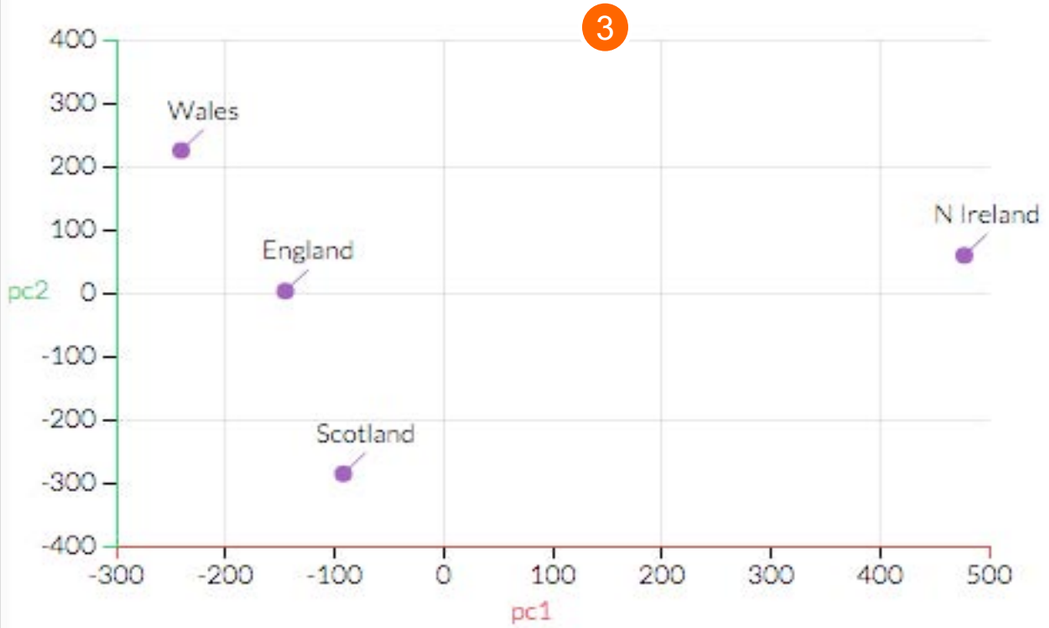
Plot along 1st principal component

2



Plot along 1st and 2nd principal components

3



Data from the Department for Environment, Food and Rural Affairs
Example from Principal Component Analysis Mark Richardson May 2009
Principal Component Analysis Explained Visually By Victor Powellwith text by Lewis Lehe

What is a Kernel?

- A similarity function: takes two inputs, returns how similar they are
- Instead of: Data > Features > Algorithm, define a kernel function to compute similarity between classes
  - Provide this kernel, and labels, to the algorithm and get a classifier
    1. Express an algorithm in terms of dot products
    2. Replace the dot product by a kernel expressed in dot products
- Kernel is often easier to compute vs. feature vector (which can be infinitely dimensional)
  - Without the "kernel trick," you would be stuck with low dimensional, low-performance feature vectors

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

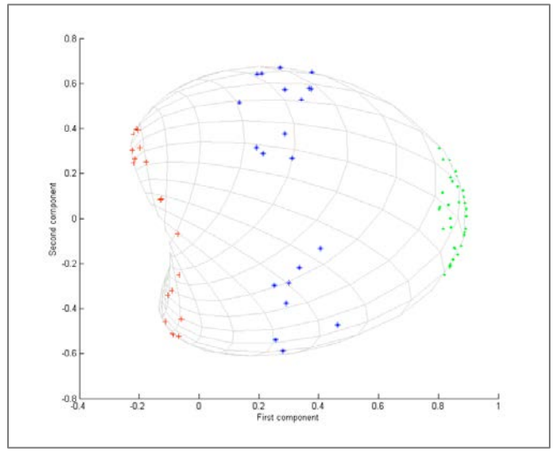
PCA Preprocessing Options

Optionally:

- Number of features to extract from the data in the **K (# of Components)** field
- Must be less than or equal to the number of fields selected and determines the number of new fields created
- If blank, the number of fields selected will be used

KernelPCA Preprocessing Options

- Extends PCA to a high dimensional feature space
  - Good for non linear data
  - Computationally expensive with a very high number of data points
- Options: specify the number of features to extract from the data in the any field
  - Must be < or = the number of fields selected and determines the number of new fields created
- If blank, the number of fields created is two
- Other parameters are optional to finetune the kernel



Previewing and Modifying Preprocessing

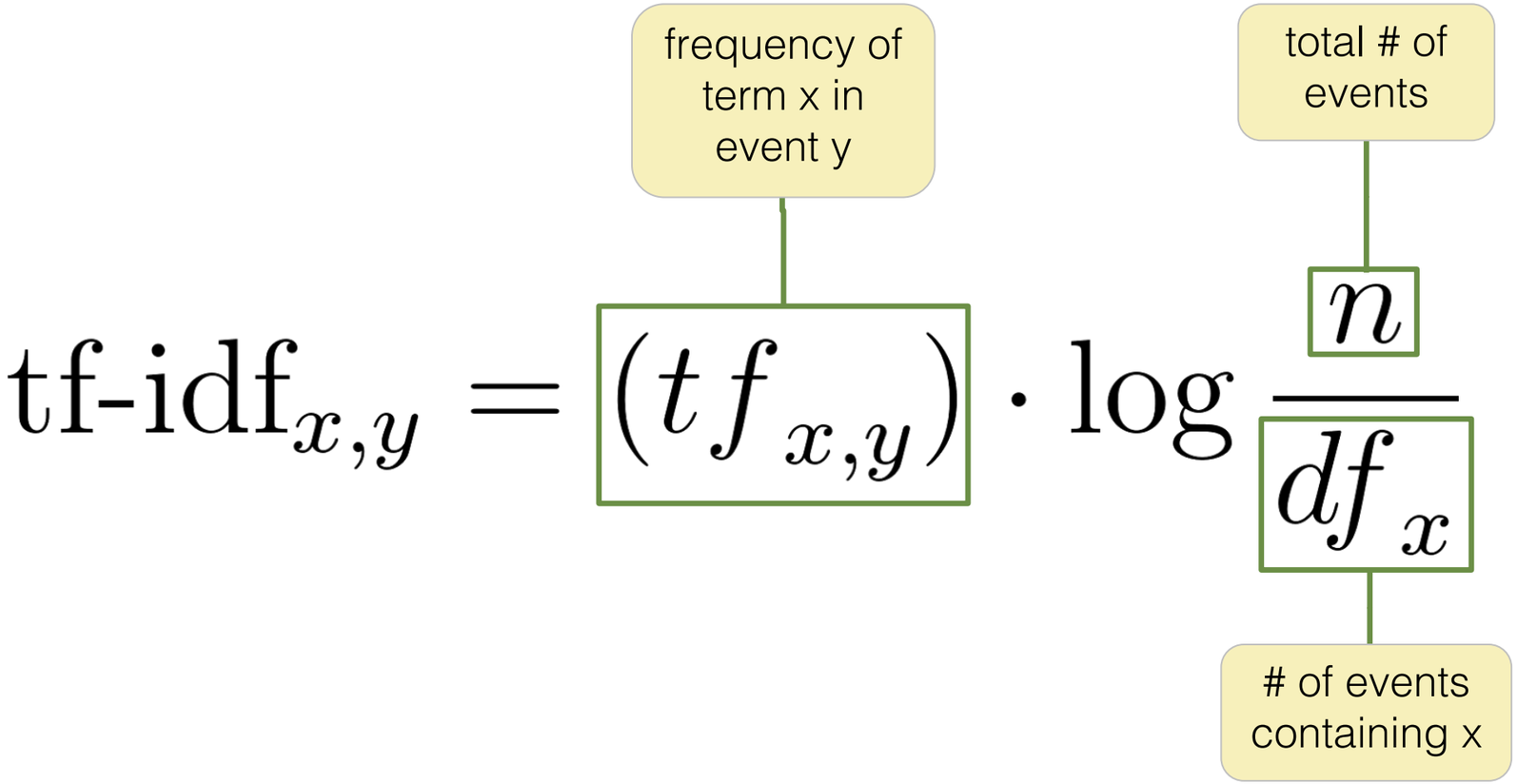
- Click **Preview Results** to see the preprocessing results
  - Fields processed using StandardScaler are prefixed with "`SS_`"
  - PCA or KernelPCA fields will be "`PC_<n>`" (`PC_1, PC_2`)
- Edit until satisfied: try a different method, change the fields or settings, add or remove preprocessing steps
  - For multiple preprocessing steps, view the incremental results of each step by clicking  next to the step
  - For results after all preprocessing steps, click the **Preview Results** link below the steps
- Use preprocessing fields to training and fit the model

Feature Extraction with TF-IDF

1. Count word occurrences by document (**T**erm **F**requency)
   - Document-term matrix (dtm), also called a term frequency matrix
2. Apply **T**erm **F**requency - **I**nverse **D**ocument **F**requency weighting:
   - Words that occur frequently within a document but not frequently within the set of documents receive a higher weighting
     - These words are assumed to contain more meaning in that document

	Term(s) 1	Term(s) 2	Term(s) 3	Term(s) 4	Term(s) 5	Term(s) 6	Term(s) 7
Doc 1	9	0	1	0	0	0	2
Doc 2	0	0	0	0	0	1	0
Doc 3	0	0	0	0	0	0	0
Doc 4	7	0	0	5	0	0	0
Doc 5	0	0	9	0	0	0	1

TF-IDF Details



TF-IDF Parameters

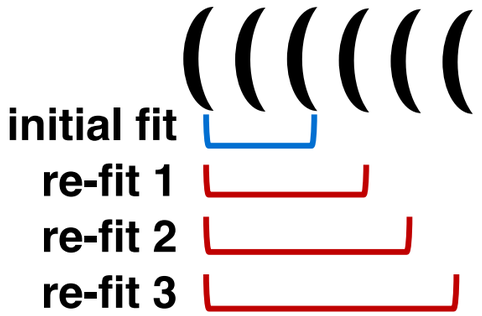
- **`max_df`** max frequency for a feature to be used in the tfi-idf matrix
  - If the term is in most events, it probably carries little meaning
- **`min_idf`** minimum number of events to be considered
  - A lower **`min_df`** may accidentally cluster on names; for example "Michael" or "Tom" are in several movies, but carry no real meaning
- **`analyzer`** select word or character n-grams **`(word or char)`**
- **`stop_words=english`** remove common words like “if” **`analyzer=word`**
- **`ngram_range`** (i.e. **`ngram_range=1-3`**)
  - min_n and max_n range of n-values for different n-grams .
  - (All values of n so that min_n <= n <= max_n is used)

Note

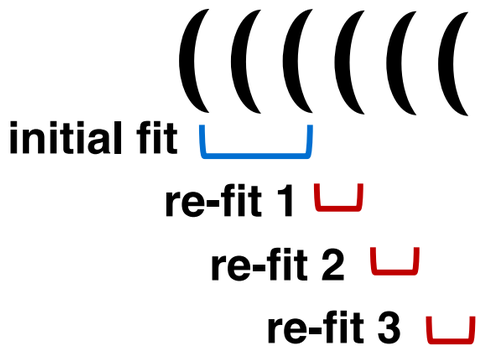
`stop_words` can be customized

Partial Fit

- Regular re-fit
  - Requires all previously-fit data
  - Plus new data



- Partial fit
  - Updates the fit instead of refitting all the data from the beginning
  - (the previous data is not needed)



Partial Fit Parameters

`partial_fit=<true|false>` is a parameter

- Default is `false` (updates model using all data)
- When set to `true`, updates an existing model using only new data
- Available with:
  - `BernoulliNB`
  - `GaussianNB`
  - `Birch`
  - `StandardScaler`
  - `SGDRegressor`
  - `SGDClassifier`

```
fit GaussianNB
<field_to_predict> from
<explanatory_fields>
[into<model name>]
[partial_fit=<true|false>]
```

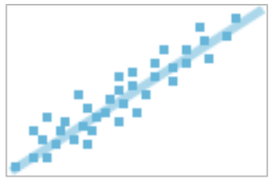
- `n_iter=<int>` sets the number of passes over the training data
  - Is set to 1 when `partial_fit` is set to `true` (default is 5)
  - Relevant only to `SGDRegressor` and `SGDClassifier`

ML Toolkit Custom Visualizations

Available on any Splunk Enterprise instance on which the **Machine Learning Toolkit** is installed

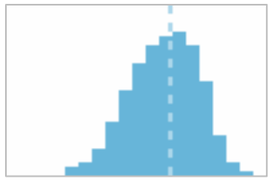
Outliers Chart (OutliersViz)

Forecast Chart (ForecastViz)

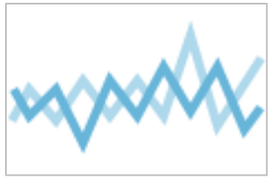


Scatter Line Chart (ScatterLineViz)

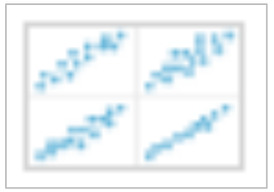
Histogram Chart (HistogramViz)



Downsampled Line Chart (LinesViz)



Scatterplot Matrix



Lab 4 – Algorithms, Preprocessing & Feature Extraction

Time: 35 minutes

Tasks:

- Fit a model on the training data
- Apply the model on the test set
- Compare the effectiveness of different models
- Summarize, list and delete models you saved

Module 5: Market Segmentation & Transactional Analysis

Module Objectives

- Define two types of market segmentation
- Describe the process of segmenting markets
- Use Splunk to find market segments
- Validate the coherence of market segments
- Define transactional analysis
- Identify transactional analysis use cases
- Use Splunk to build, analyze, and visualize higher-level (meta) transactions

Market Segmentation and Transactions?

- What do they have in common?
  - Both focus on analyzing the behavior of anything:
    - People
    - Servers
  - Market segmentation finds groups with similar behavior to address a potential customer experience
  - Transactional analysis groups instances of related behaviors to determine a story or longer term experience that can then be optimized

Types of Market Segmentation

- a priori
  - Easy to define using existing discrete values in fields
    - Male vs. female

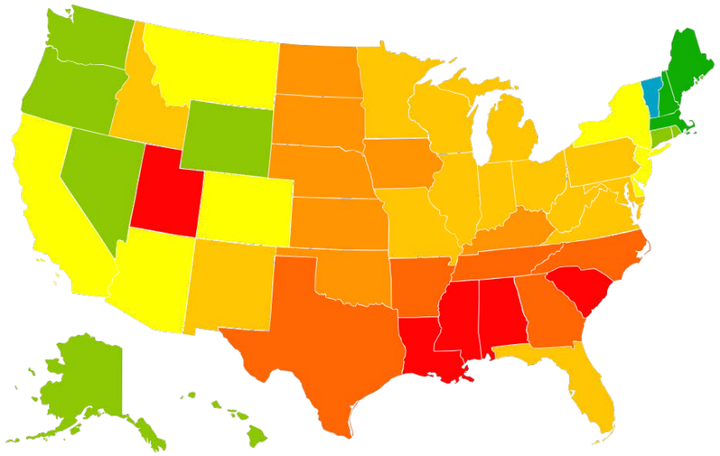
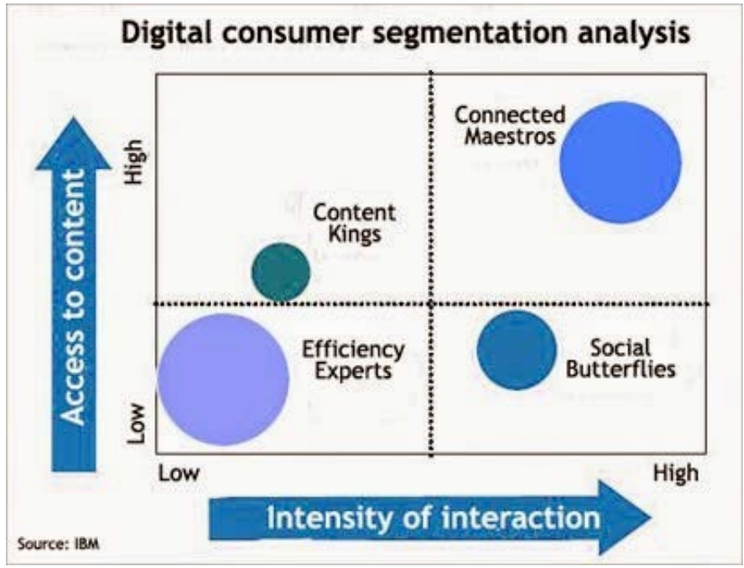


Image: U.S. segmentation. Data: church or synagogue attendance
Source:
http://en.wikipedia.org/wiki/Christianity_in_the_United_States#mediaviewer/File:Church_or_synagogue_attendance_by_state_GFDL.svg
Attribution: Creative Commons, Falcorian

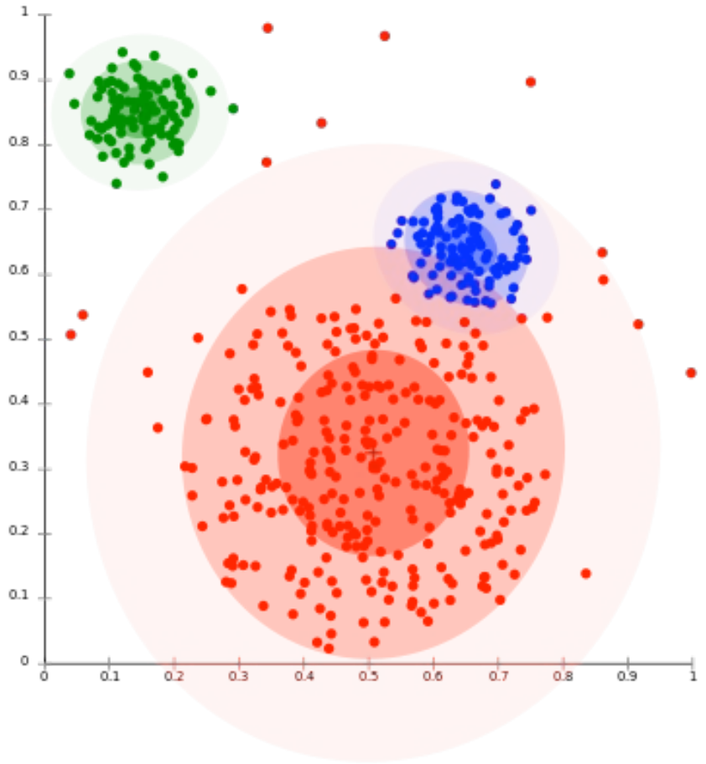
- post hoc
  - Discover new groupings or combinations in the data
    - Cluster analysis



Market Segmentation Process

- Leverage existing labels in the data before using ML
- Human input is key to the success of this process

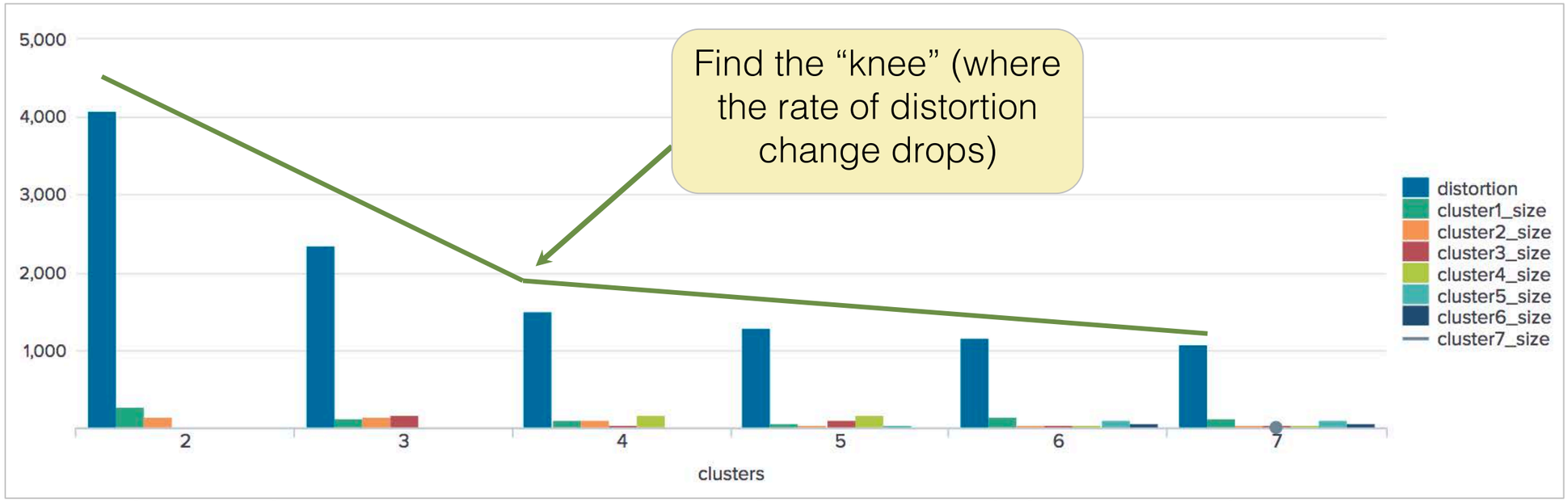
1. Find clusters (market segments)
   - Use cluster commands, ML models
2. Validate (is there coherence?)
   - Assign labels & make meaningful names
3. Target strategies to customer preferences
   - Geographic
   - Demographic
   - Behavioral



`kmeans` Command for Market Segmentation

Instead of a number, use a range for `kmeans` to examine options and optimize the number of market segments

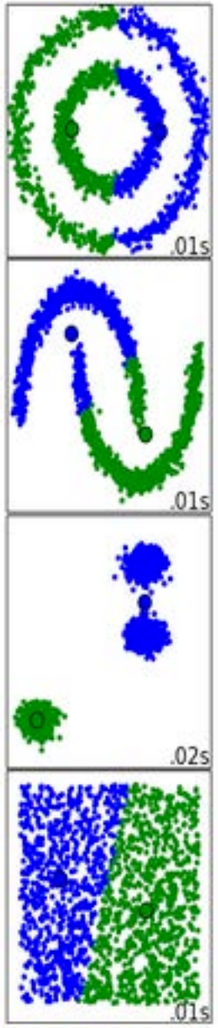
```
sourcetype=access_combined action=purchase
| stats sum(price) as order_total by JSESSIONID
| kmeans k=2-7 order_total
```



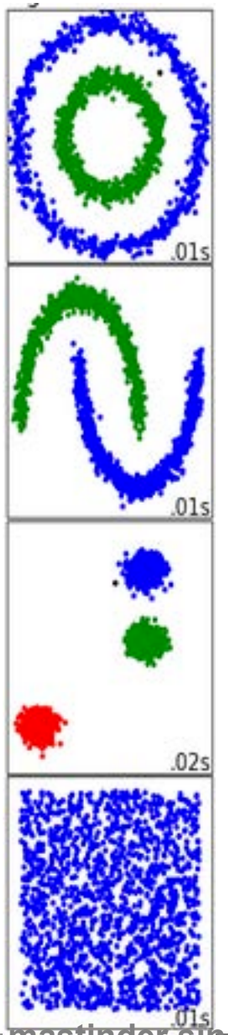
Clustering Plots

(XMeans is similar)

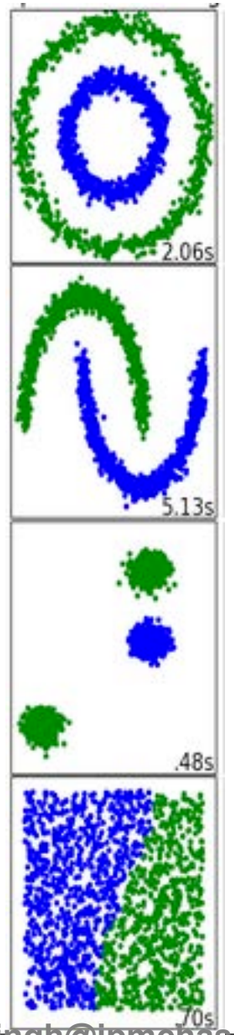
Kmeans



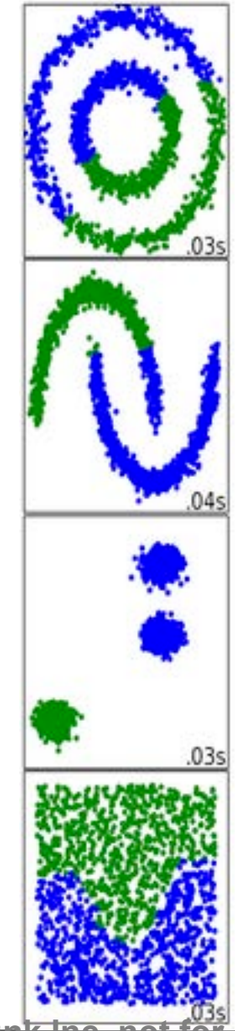
DBSCAN



SpectralClustering

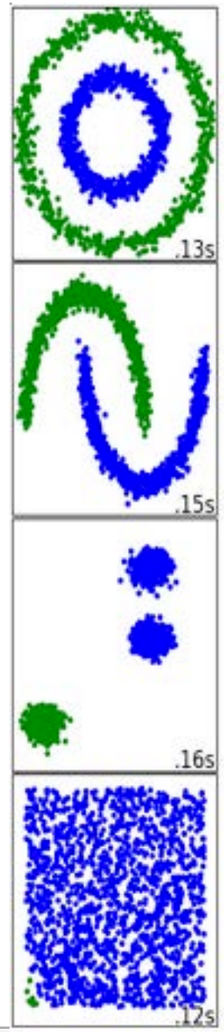


Birch



(Splunk command)

cluster



Validating Coherence

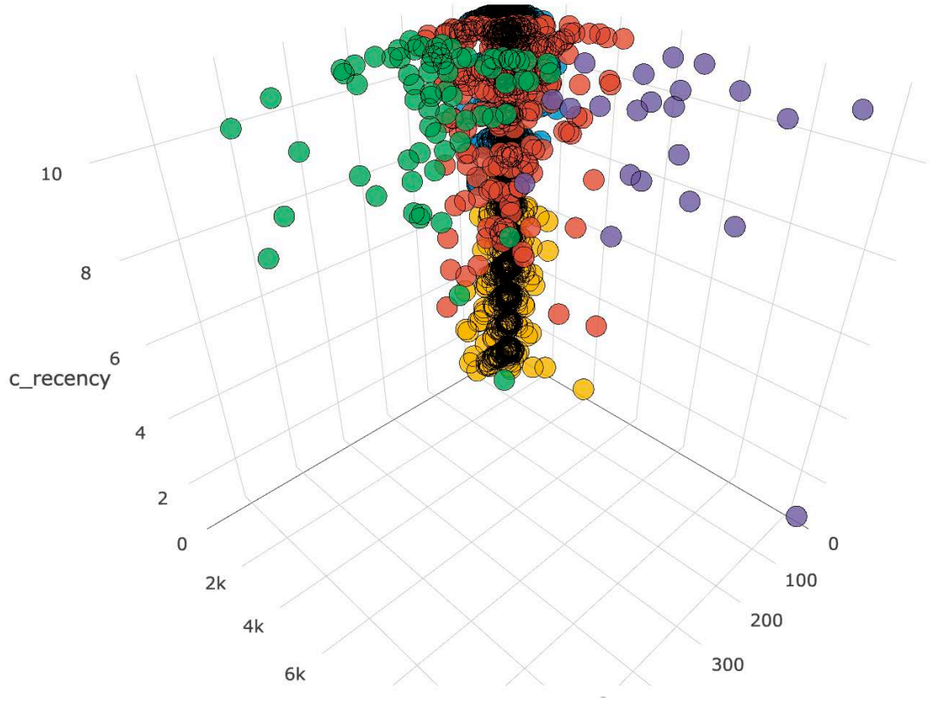
Do the clusters actually relate to anything?

```
sourcetype=access_combined action=purchase AND JSESSIONID=* AND product_name=* earliest=-7d@d
| fields price JSESSIONID product_name
| stats list(product_name) as product_name sum(price) as spending by JSESSIONID
| kmeans k=4 spending
| chart count by product_name CLUSTERNUM
```

product_name	1	2	3	4
Benign Space Debris	116	32	43	0
Curling 2014	136	39	40	0
Dream Crusher	0	102	231	0
Final Sequel	208	54	66	0
Fire Resistance Suit of Provolone	76	17	46	233
Holy Blade of Gouda	59	15	39	186
Manganiello Bros.	0	96	242	0
Manganiello Bros. Tee	36	12	57	197
Mediocre Kingdoms	235	62	92	0
Orvil the Wolverine	0	83	166	0
Puppies vs. Zombies	35	9	33	176
SIM Cubicle	241	58	55	0
World of Cheese	252	70	86	0
World of Cheese Tee	35	12	49	167

Retail Market Segmentation Example

- Improve customer-centric marketing
  - Segmented customers into meaningful groups based on time, frequency, and a monetary model using kmeans
  - Main characteristics of the consumers in each segment are clearly identified
- 12-page detailed analysis can be approximated in 10 lines of SPL
  - `Eventstats -StandardScaler`
  - `KMeans -stats`



Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining"
http://www.palgrave-journals.com/dbm/journal/v19/n3/pdf/dbm201217a.pdf

Transactional Analysis Use Cases

- Transactional Analysis: the study of any group of conceptually-related events which spans time and describes behavior
- Transactions are “high-level” events:
  - DDoS attack from unknown IP ranges
  - John Smith purchased a product on the website
- Transactions span IT & business data sources:
  - “Purchase transaction” involves web-server logs, e-commerce data, product and customer lookups
- Business users care about transactions
  - Basic unit of economic activity

Concept: Meta Transactions

- Transactions-of-transactions that represent the entirety of each customer's experiences
  - Do transaction on transaction events by customer id
  - Build a meta transaction of all of a customer's experiences
  - `ServiceTicketHistory -` full customer experience
- With a *collection* of these, you can do statistics
  - Find all customers who had a positive experience with your product

Event	Score
Search for site	9,0
Search products	4.6
Compare products	3.2
Add to wish list	7.4
Place in cart	9.1
Purchase	8.8
Receive shipment	9.3
Review product	7.9
Recommend product	9.6
File a warranty claim	2.1
Receive replacement	4.8

How to Construct Meta Transactions

- Goal: attach KPI of interest to high-level customer experiences
  - Construct transactions
  - Assemble transactions into higher-level transactions
  - They may be ongoing / may not have finished yet
- Avoid running `transaction` on top of `transaction`
  - Can be prohibitively slow
- Use alternative functions, like `stats list()` or `stats values()`

Higher-Level Transactions Example

You can now do statistics on meta transactions

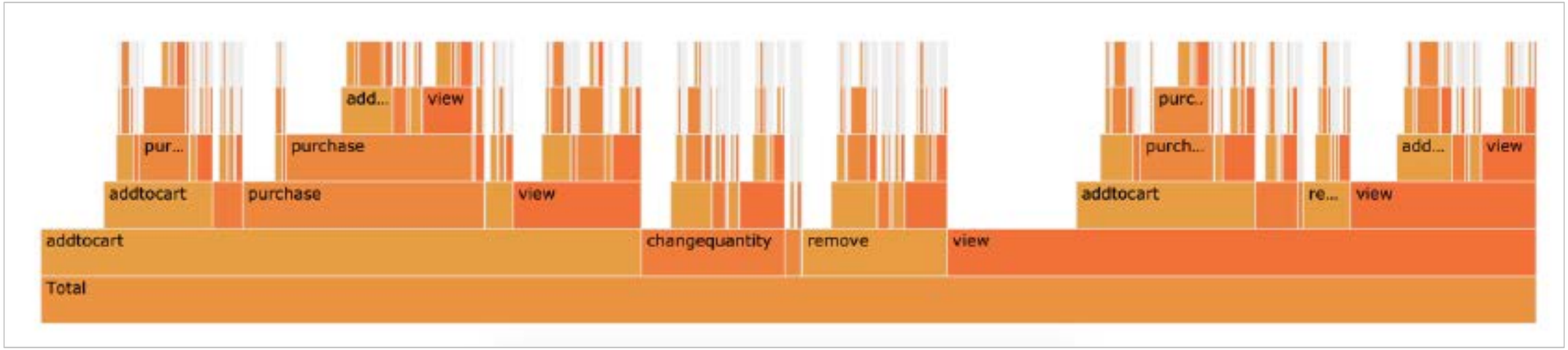
```
sourcetype=access_combined action=*
| reverse
| streamstats count as stage by JSESSIONID
| xyseries JSESSIONID stage action
| fillnull value=.
| stats count by 1 2 3 4 5
```


Aggregate & Visualize

- Higher levels built from lower levels

  https://splunkbase.splunk.com/app/3468/

  - ***Economic transactions:*** understood from low-level machine events
  - ***Customer experiences:*** from mid-level economic transactions
- High levels affect the lower levels, too
  - If customers are happy, they make more raw events (web clicks, etc.)



Module 5 Lab Exercise – Clusters & Transactions

Time: 30 minutes

Tasks:

- Normalize customer data with `StandardScaler` & determine a value for k
- Fit a `KMeans` model & reduce the dimensionality with `PCA`
- Examine the average values fields within each customer cluster
- Use `kmeans` to estimate distinct market groups
- Fit a model on your chosen value of k
- Build transactions and meta-transactions to examine web purchase failure ratios

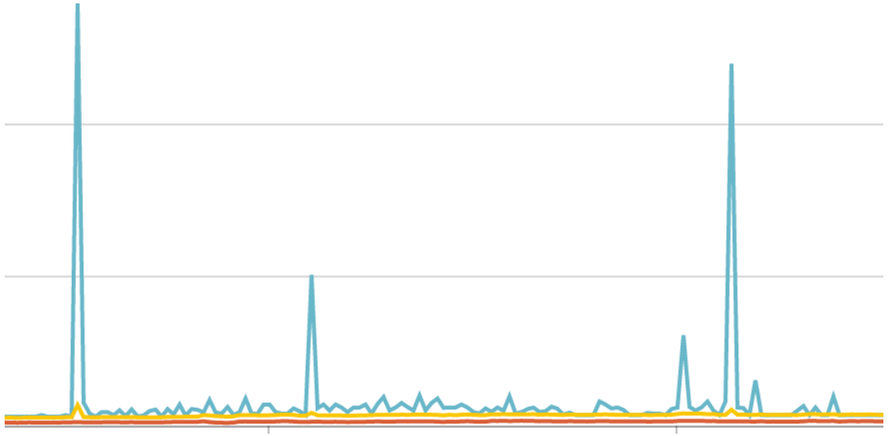
Module 6: Anomaly Detection

Module Objectives

- Define anomaly detection
- Identify anomaly detection use cases
- Use Splunk to detect and analyze outliers in numerical data
  - standard deviation
  - IQR
  - median absolute deviation
  - `anomalydetection command`
  - `cluster command`

What is Anomaly Detection?

- An anomaly is a deviation in the expected behavior of the data
  - Can be a single event
  - Sequences of events (transactions)
  - More complex patterns
- Keep the outliers when:
  - Reporting
  - Alerting

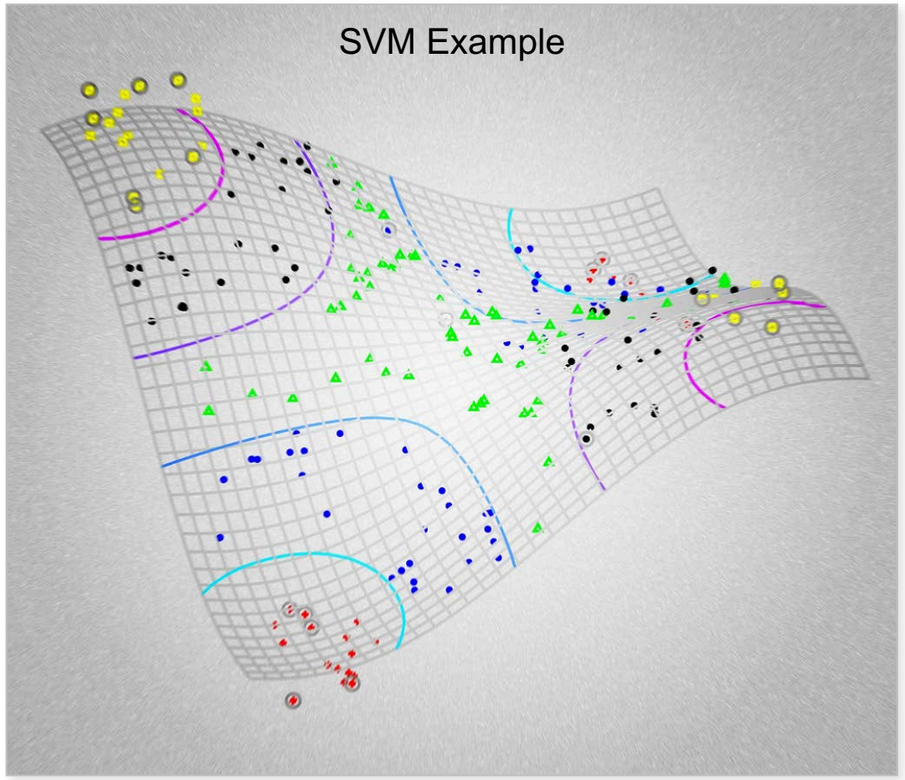


- Remove the outliers in cases like the market segmentation analysis shown earlier and in other cases where the outliers skew the analysis if they aren't removed

* Distributed Denial of Service – uses many devices and connections, often global

Anomaly Detection with ML

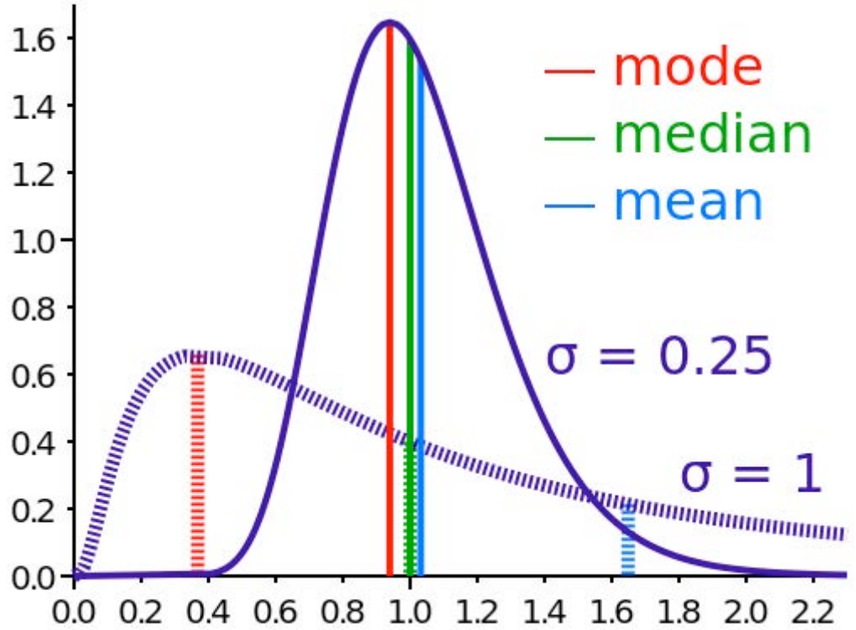
- Machine Learning (ML) helps you model expected behaviors
  - First learn the normal patterns, trends, and behaviors of the system
- ML helps identify sufficiently large deviations
  - Assign anomaly scores to events and patterns
  - Estimate probabilities of rare events
- Create a baseline data image (current state of the system)



https://www.flickr.com/photos/javism/8737879875/

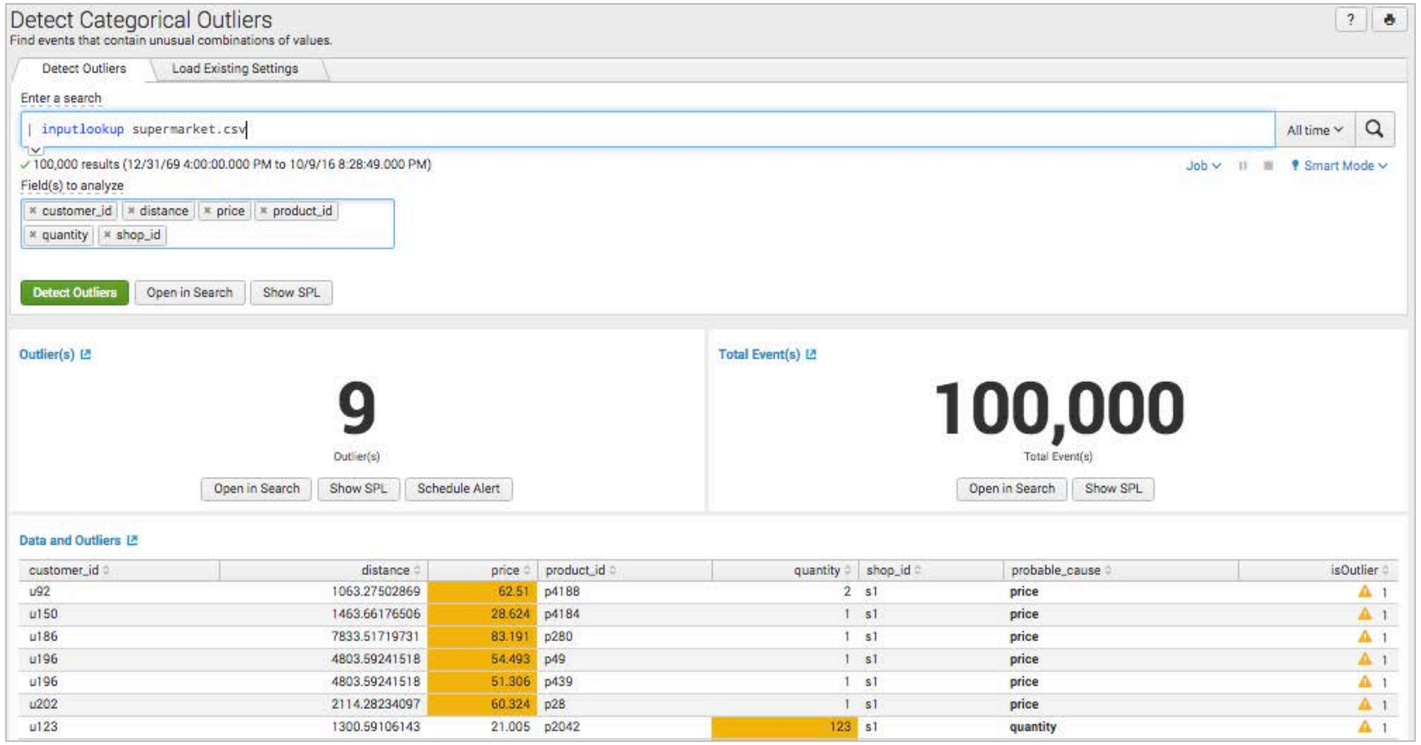
Statistical Outliers for Anomaly Detection

- Finding statistical outliers is a first step to effective anomaly detection
- A statistical outlier is any event which is far from some measure of centrality
  - Some measures of centrality
    - Mean: the average value
    - Median: the typical value
    - Mode: the most frequent value
  - Statistical outliers come in many different flavors
    - Non-average: far from the mean
    - Non-typical: far from the median
    - Non-popular: small count relative to the mode



Categorical Outliers

- Categorical outliers: events that contain unusual combinations of
  - Non-numeric values
  - Multi-dimensional data
    - String identifiers
    - IP addresses



customer_id	distance	price	product_id	quantity	shop_id	probable_cause	isOutlier
u92	1063.27502869	62.51	p4188	2	s1	price	1
u150	1463.66176506	28.624	p4184	1	s1	price	1
u186	7833.51719731	83.191	p280	1	s1	price	1
u196	4803.59241518	54.493	p49	1	s1	price	1
u196	4803.59241518	51.306	p439	1	s1	price	1
u202	2114.28234097	60.324	p28	1	s1	price	1
u123	1300.59106143	21.005	p2042	123	s1	quantity	1

Numeric Outliers – Standard Deviation

```
sourcetype = access_combined status>399 earliest=-1w
| timechart span=1h count
| streamstats window=24 avg(count) as avg, stdev(count) as stdev
| eval multiplier = 2
| eval lower_bound = avg - (stdev * multiplier)
| eval upper_bound = avg + (stdev * multiplier)
| eval outlier = if(count < lower_bound OR count > upper_bound, 1, 0)
| table _time upper_bound count outlier
```


Numeric Outliers – Std. Deviation (cont.)

```
sourcetype = access_combined status>399 earliest=-1w
| timechart span=1h count
| streamstats window=24 avg(count) as avg, stdev(count) as stdev
| eval multiplier = 2
| eval lower_bound = avg - (stdev * multiplier)
| eval upper_bound = avg + (stdev * multiplier)
| eval outlier = if(count < lower_bound OR count > upper_bound, 1, 0)
| table _time upper_bound count outlier
```

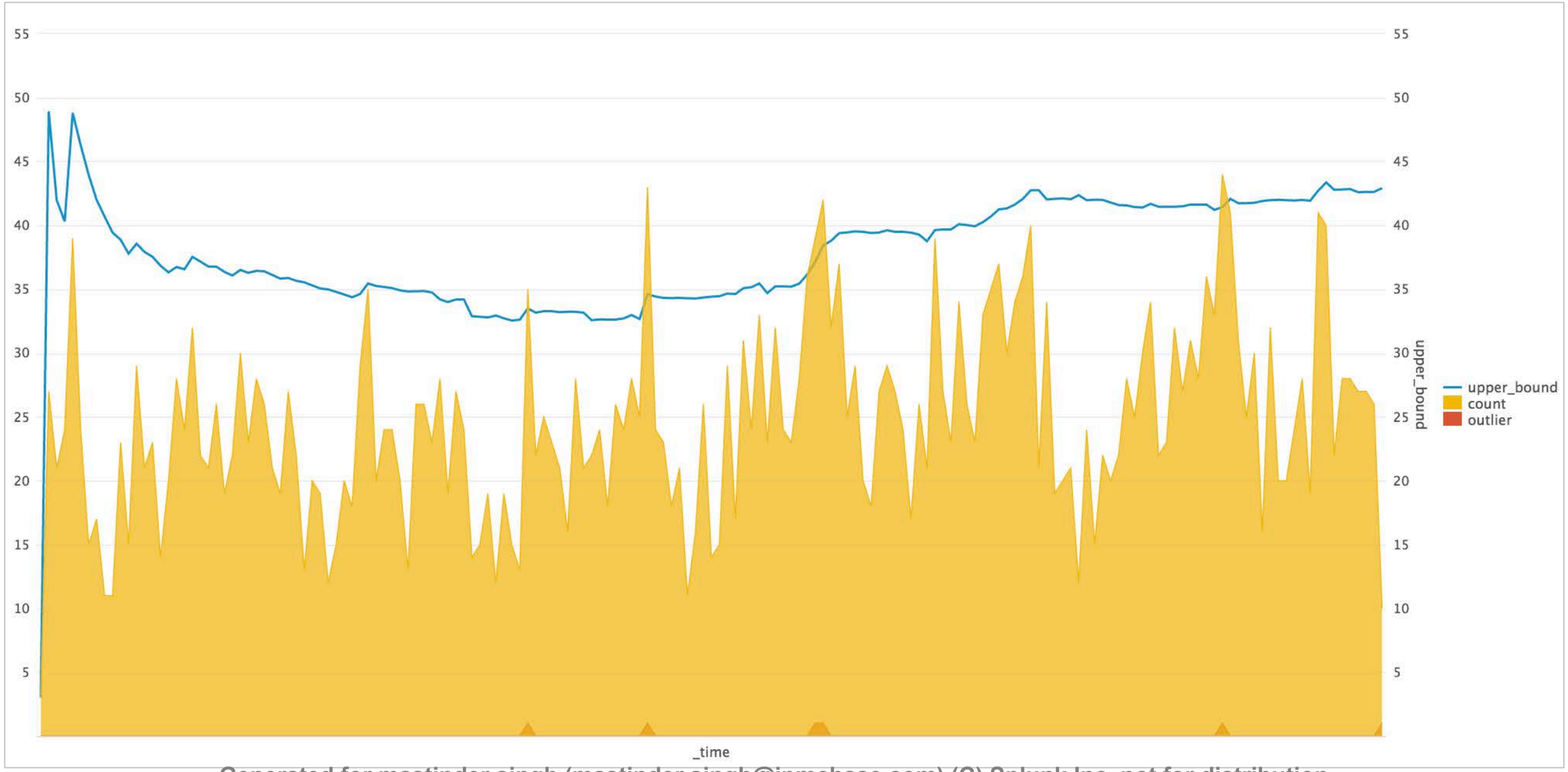

Numeric Outliers – Std. Deviation (cont.)

```
sourcetype = access_combined status>399 earliest=-1w
| timechart span=1h count
| streamstats window=24 avg(count) as avg, stdev(count) as stdev
| eval multiplier = 2
| eval lower_bound = avg - (stdev * multiplier)
| eval upper_bound = avg + (stdev * multiplier)
| eval outlier = if(count < lower_bound OR count > upper_bound, 1, 0)
| table _time upper_bound count outlier
```


Numeric Outliers – Std. Deviation (cont.)

```
sourcetype = access_combined status>399 earliest=-1w
| timechart span=1h count
| streamstats window=24 avg(count) as avg, stdev(count) as stdev
| eval multiplier = 2
| eval lower_bound = avg - (stdev * multiplier)
| eval upper_bound = avg + (stdev * multiplier)
| eval outlier = if(count < lower_bound OR count > upper_bound, 1, 0)
| table _time upper_bound count outlier
```


Numeric Outliers – Std. Dev. Visualization



Numeric Outliers - IQR

```
sourcetype=access_combined status>399 earliest=-1w
| timechart span=1h count
| eventstats median(count) as median, p25(count) as p25, p75(count) as p75
| eval IQR = p75 - p25
| eval multiplier = 2
| eval lower_bound = median - (IQR * multiplier)
| eval upper_bound = median + (IQR * multiplier)
| eval outlier = if(count < lower_bound OR count > upper_bound, 1, 0)
| table _time count lower_bound upper_bound outlier
```


Numeric Outliers – IQR (cont.)

```
sourcetype=access_combined status>399 earliest=-1w
| timechart span=1h count
| eventstats median(count) as median, p25(count) as p25, p75(count) as p75
| eval IQR = p75 - p25
| eval multiplier = 2
| eval lower_bound = median - (IQR * multiplier)
| eval upper_bound = median + (IQR * multiplier)
| eval outlier = if(count < lower_bound OR count > upper_bound, 1, 0)
| table _time count lower_bound upper_bound outlier
```


Numeric Outliers – IQR (cont.)

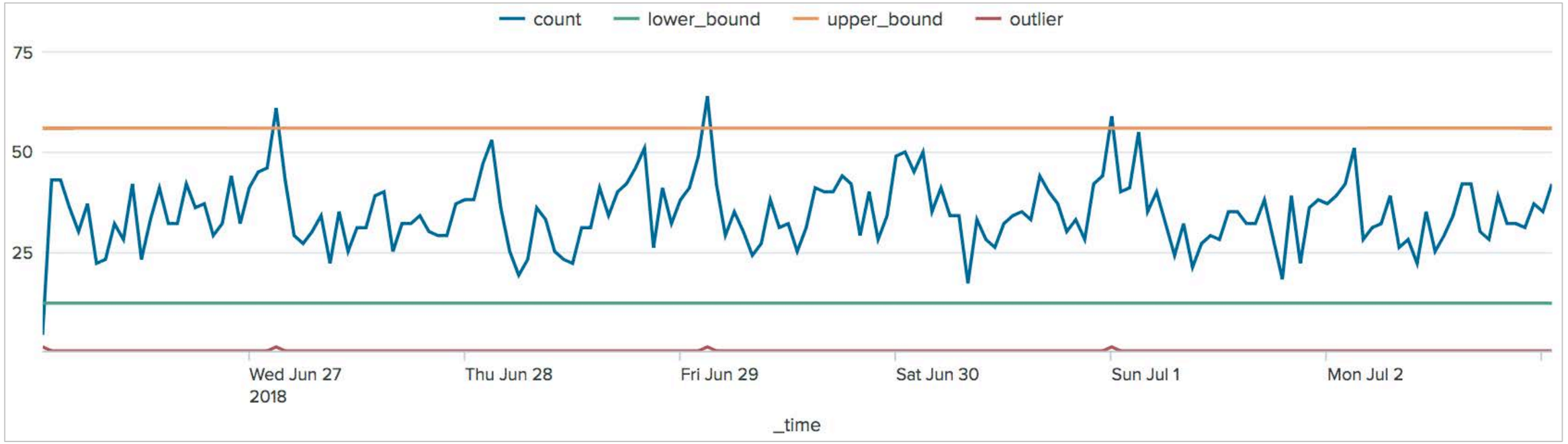
```
sourcetype=access_combined status>399 earliest=-1w
| timechart span=1h count
| eventstats median(count) as median, p25(count) as p25, p75(count) as p75
| eval IQR = p75 - p25
| eval multiplier = 2
| eval lower_bound = median - (IQR * multiplier)
| eval upper_bound = median + (IQR * multiplier)
| eval outlier = if(count < lower_bound OR count > upper_bound, 1, 0)
| table _time count lower_bound upper_bound outlier
```


Numeric Outliers – IQR (cont.)

```
sourcetype=access_combined status>399 earliest=-1w
| timechart span=1h count
| eventstats median(count) as median, p25(count) as p25, p75(count) as p75
| eval IQR = p75 - p25
| eval multiplier = 2
| eval lower_bound = median - (IQR * multiplier)
| eval upper_bound = median + (IQR * multiplier)
| eval outlier = if(count < lower_bound OR count > upper_bound, 1, 0)
| table _time count lower_bound upper_bound outlier
```


Numeric Outliers – IQR Visualization

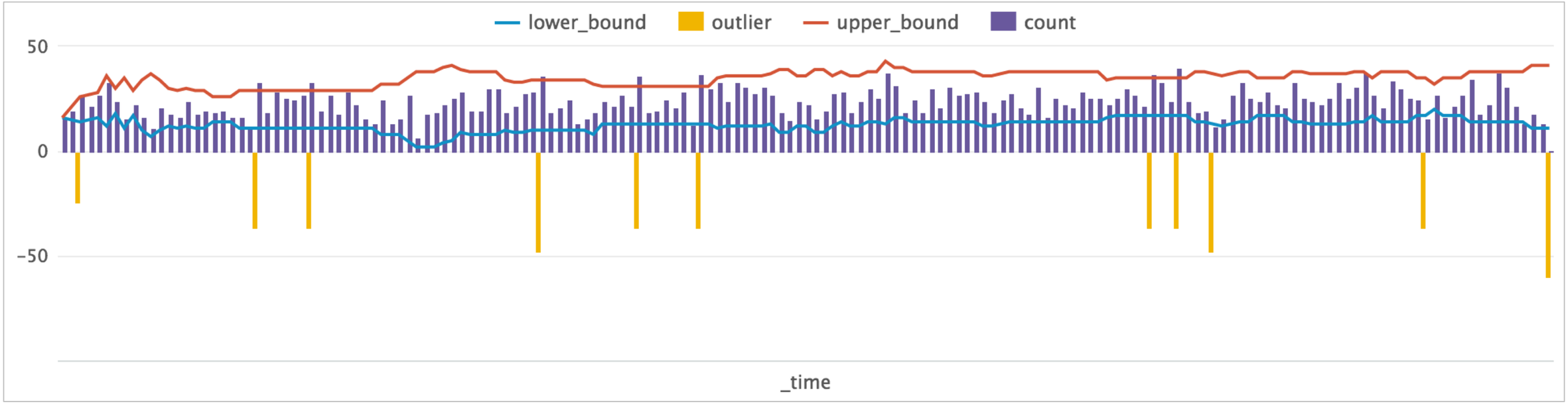
Column chart with all tabled fields as overlays except **`outlier`**



Numeric Outliers: Median Abs. Deviation

```
sourcetype=access_combined status>399 earliest=-1w
| timechart span=1h count
| streamstats window=24 median(count) as median
| eval abs_dev = abs(count - median)
| streamstats window=24 median(abs_dev) as median_abs_dev
| eval lower_bound = median - (median_abs_dev * 3)
| eval upper_bound = median + (median_abs_dev * 3)
| eval outlier = if(count < lower_bound OR count > upper_bound, 1, 0)
| table _time lower_bound outlier upper_bound count
```


Median Absolute Deviation Visualization




```
. . . | eval outlier = if(count < lower_bound OR
count > upper_bound, -median_abs_dev * 5 , 0)
| table _time lower_bound count upper_bound outlier
```

anomalydetection Command

```
anomalydetection [<method-option>] [<action-option>]
[<pthresh-option>] [<cutoff-option>] [<field-list>]
```

- If `method = histogram` or `method = zscore` available actions are:
  - `action = filter`
  - `action = annotate`
  - `action = summary`
- If `method = IQR` available actions are:
  - `action = remove`
  - `action = transform`
- `cutoff <bool>`

anomalydetection method = zscore

- Finds, filters, or summarizes irregular or uncommon search results
  - Assigns an “anomaly score” for individual fields
  - Looks at the entire event set and considers the distribution of values
  - For numeric fields, identifies values that are anomalous either
    - By frequency of occurrence or
    - By number of standard deviations from the mean
  - For anomalous fields, a new field is added with the following scheme
    - If the field is numeric, e.g. size, the new field will be `Anomaly_Score_Num(size)`
    - If the field is non-numeric, e.g. name, the new field will be `Anomaly_Score_Cat(name)`

anomalydetection zscore Details

(action=annotate | action=filter)

If numeric	If categorical or < 100 distinct values
$\texttt{Anomaly\_Score\_Num(x)} = p(x) \sim \mathcal{N}(\mu, \sigma)$ p(x) is the two-tailed probability of seeing such a value in a normal distribution with mean mu and standard deviation sigma	1.If frequency(x) < pThresh, it counts as anomalous $frequency(x) = \frac{count(x)}{dc(x) \times count}$ 2.Find the average frequency for non-anomalous values $avg_{freq} = \frac{count(x \neq anomalous)}{dc(x \neq anomalous) \times count}$ 3.Calculate the anomaly score: $\texttt{Anomaly\_Score\_Cat(x)} = \frac{frequency(x)}{avg_{freq}(x)}$

anomalydetection method = IQR

- Specifies what to do with outliers
  - **action=remove** removes event containing outlying numerical value
    - abbreviation **rm**
  - **action=transform**: truncates outlying value to the outlier threshold
    - default
    - abbreviation **tf**
    - If **mark=true**, the transform action prefixes the value with "000"

```
. . .
| stats sum(filesize) as volume, count as count by _time, weekday
| anomalydetection volume count method=iqr action=transform
```


anomalydetection method = histogram

1. Computes a probability for each event, default
   - Product of the frequencies of each individual field value in the event
2. Detects unusually small probabilities
   - Categorical fields
     - Frequency of X is the number of times X occurs divided by the total number of events
   - Numerical fields
     - Builds a histogram for all the values
     - Computes the frequency of a value X as the size of the bin that contains X divided by the number of events

anomalydetection Command Example

```
| inputlookup supermarket.csv | anomalydetection action=filter
```

Last 24 hours

✓ 9 results (6/8/18 11:00:00.000 PM to 6/9/18 11:14:24.000 PM) No Event Sampling Job Smart Mode

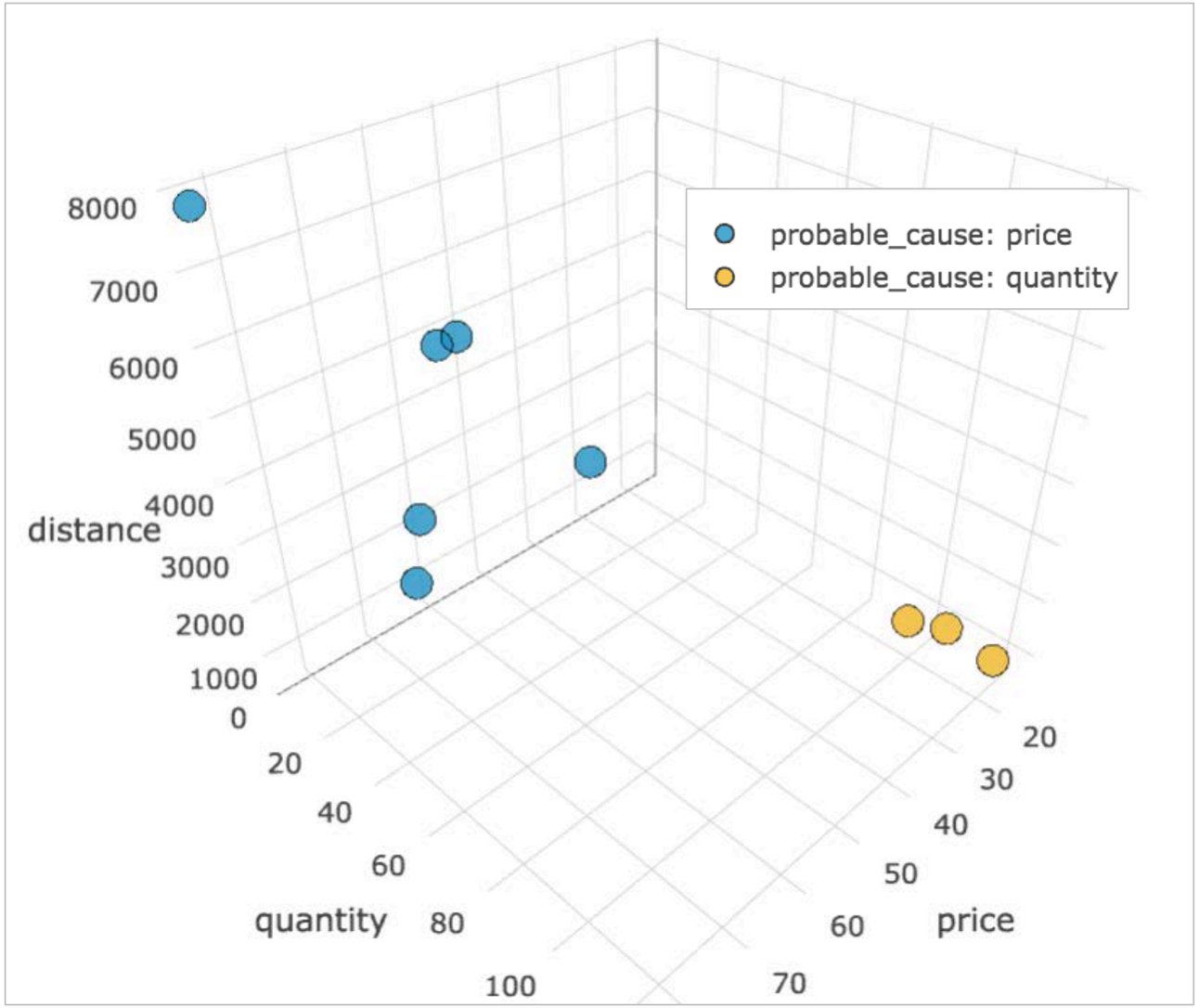
Events Patterns **Statistics (9)** Visualization

20 Per Page Format Preview

customer_id	distance	log_event_prob	max_freq	price	probable_cause	probable_cause_freq	product_id	quantity	shop_id
u62	2681.15450705	-36.1423	0.35050	0.172	price	0.00000	p325	9	s2
u72	248.699619449	-36.6300	0.35050	0.172	price	0.00000	p325	9	s4
u137	961.408935339	-35.6178	0.35050	0.172	price	0.00000	p325	5	s3
u137	961.408935339	-35.4043	0.61832	16.92	quantity	0.00004	p4029	106	s3
u166	2695.99047282	-35.1999	0.35050	0.172	price	0.00000	p325	1	s4
u176	1708.6583926	-35.5145	0.35050	0.172	price	0.00000	p325	7	s1
u196	4803.59241518	-36.0327	0.35050	51.306	price	0.00001	p439	1	s1
u214	1529.69862056	-35.9738	0.35050	0.172	price	0.00000	p325	6	s3
u231	583.590151221	-36.2727	0.35050	0.172	price	0.00000	p325	18	s2

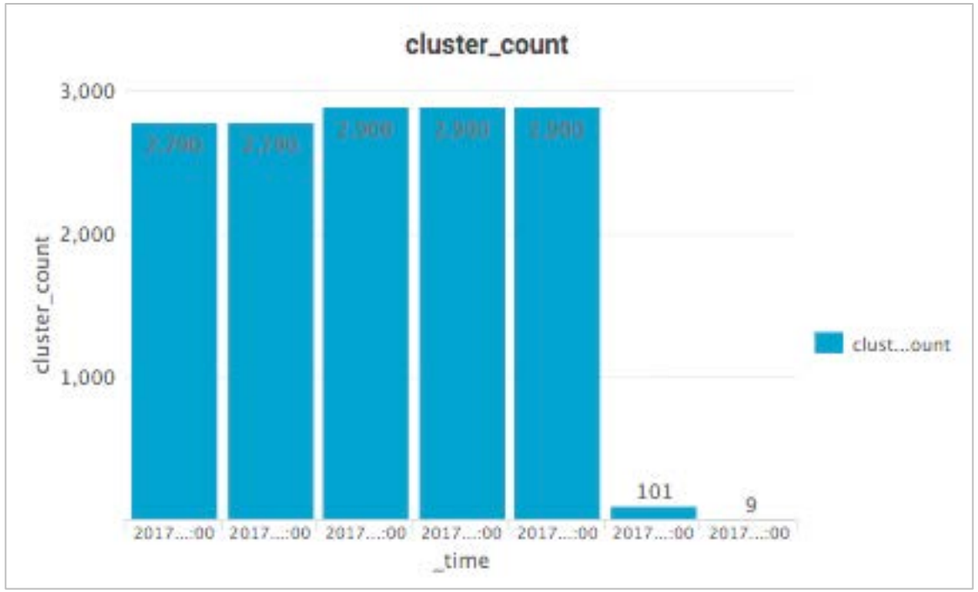
anomalydetection Command Example (cont.)

```
| inputlookup supermarket.csv
| anomalydetection action=filter
| table probable_cause price quantity distance
```



`cluster` Command for Anomaly Detection

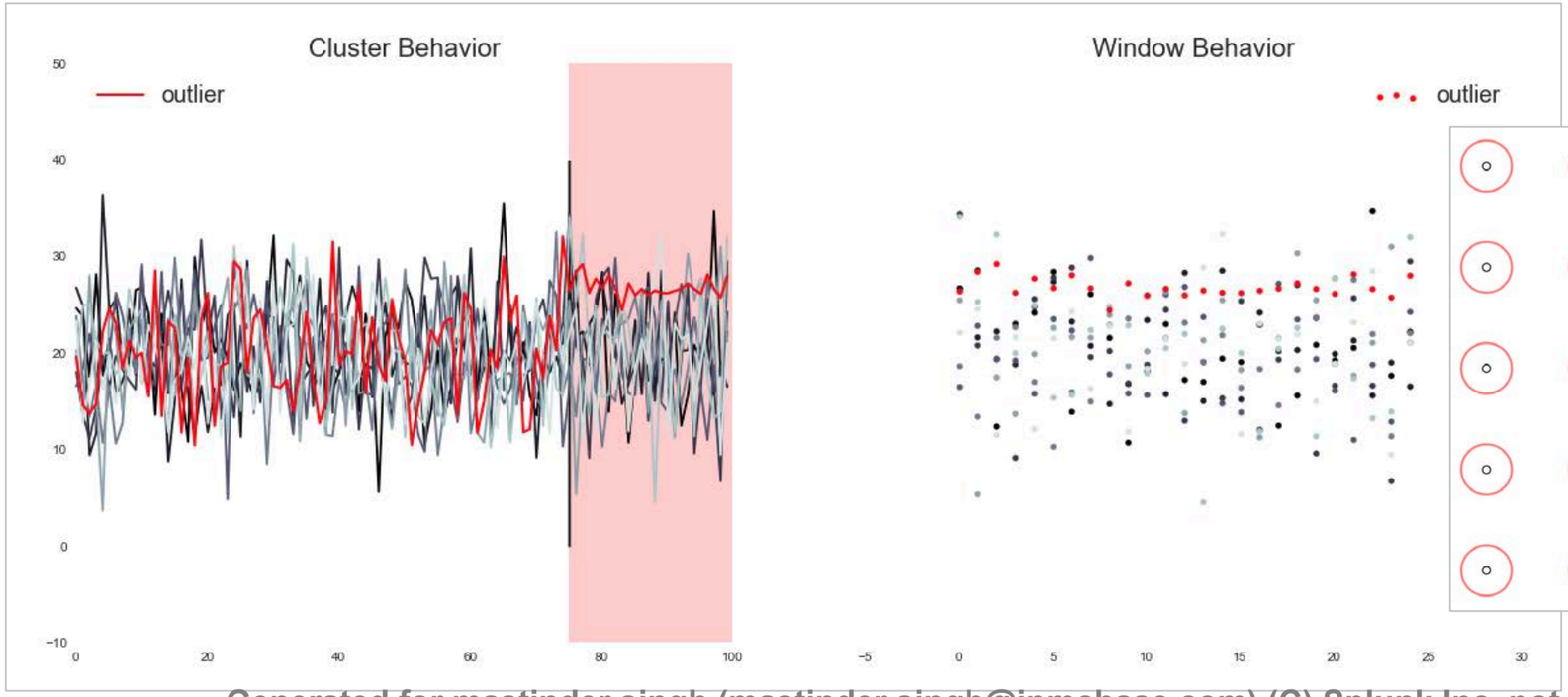
- Small group anomaly: 2 successful logins from a terminated user
  - For small groups, sort *ascending* by **`cluster_count`**
- Large group anomaly: a DDoS attack of 1000s of similar events
  - For large groups, sort *descending* by **`cluster_count`**

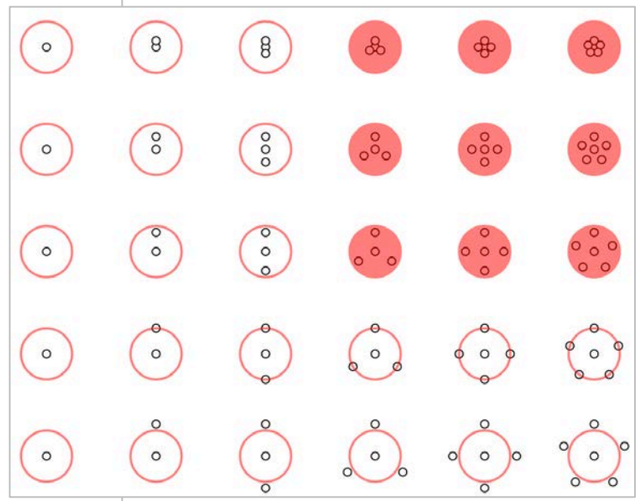



```
source="/opt/log/ecommsv1/sales_entries.log"
| cluster showcount=t t=0.7 labelonly=t
| table _time, cluster_count, cluster_label, _raw
| dedup 1 cluster_label
| table cluster_count, _time by cluster_label
| sort -_time, cluster_count
```

Netflix Example

- Fewer than 1% of servers are slow (often within set tolerances)
  - Collect only outliers and do cluster analysis with DBSCAN
  - Remove or terminate server

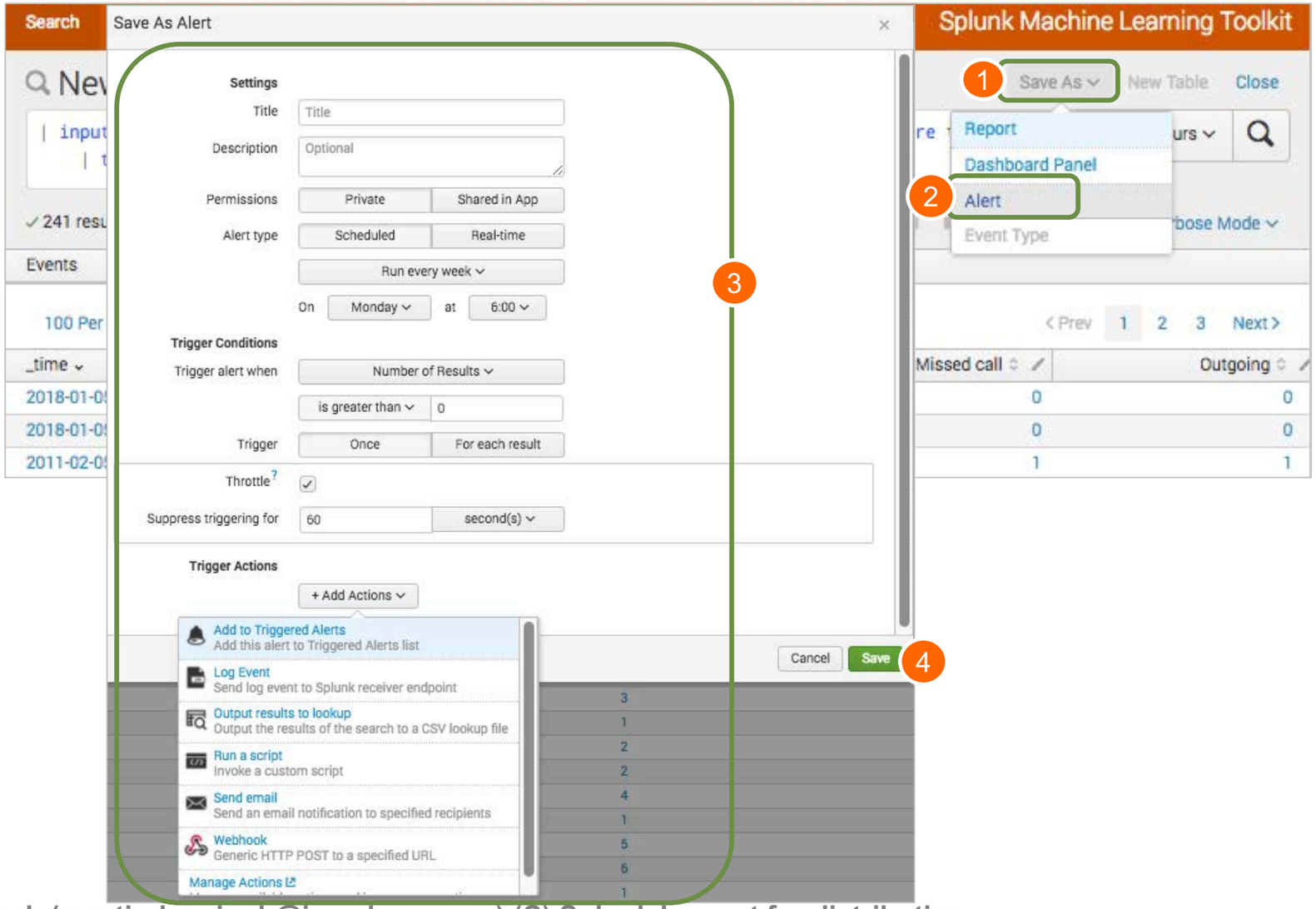




http://techblog.netflix.com/2015/07/tracking-down-villains-outlier.html

Alert for Outliers with a Throttle

- Throttling suppresses actions for results that have the same field value in a time range
- Too many outliers (5% in normal data) to alert on each
  - Throttle and provide count & percentage
  - Alert only when that count or percentage changes drastically



Module 6 Lab Exercise – Analyze Potential Anomalies

Time: 35 – 45 minutes

Tasks:

- Compare recent clusters to older clusters to determine if there are any anomalies
  - Use `streamstats` and `eval` to calculate standard deviation
  - Use `streamstats` and `eval` to calculate median absolute deviation
  - Use `eventstats` and `eval` to calculate IQR
- Compare standard deviation, median absolute deviation, and IQR against a second dataset
- Use the `cluster` command to find anomalies in other types of data
- Use the IQR method of removing outliers

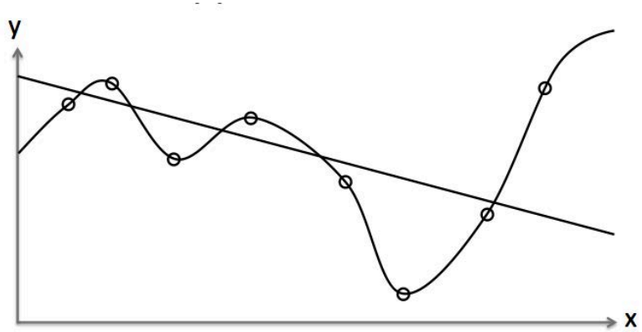
Module 7: Estimation & Prediction

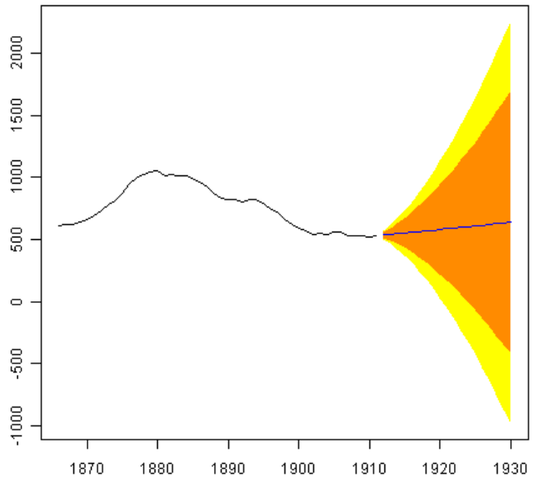
Module Objectives

- Define estimation and prediction
- Identify estimation and prediction use cases
- Define linear regression
- Describe common assumptions in linear regression
- Describe Splunk **`predict`** command

Estimating and Predicting

Estimation	Prediction
Calculated approximation of how x and y are related	Estimation of y based on known values of y
Could involve time or not	Could estimate future values (forecast) or not
Effectiveness is measured by p-values	Effectiveness is measured by cross-validation (accuracy)
Fixed effects	Effects may be random





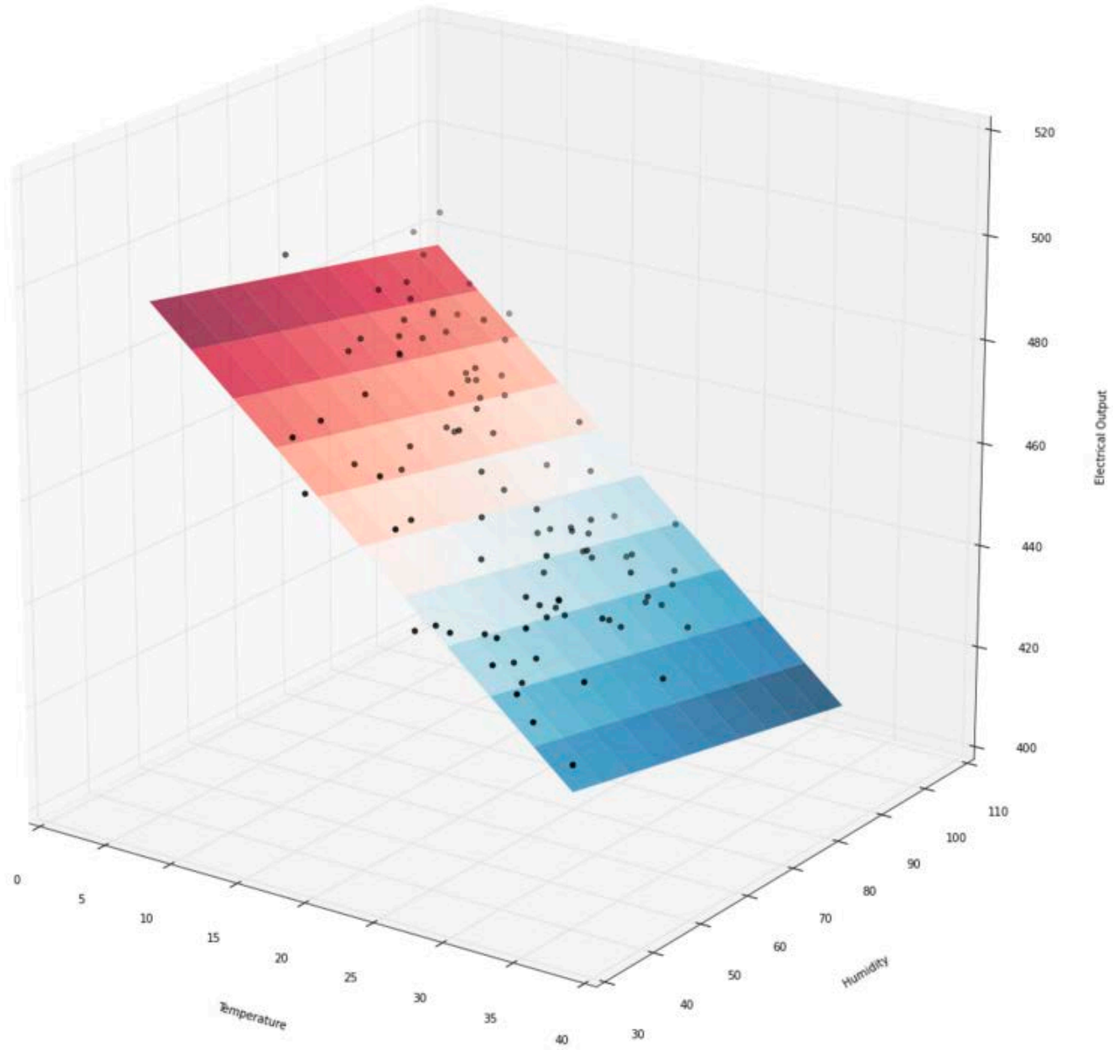
https://dartthrowingchimp.wordpress.com/2013/10/ by DARTTHROWINGCHIMP on OCTOBER 24, 2013

Estimation and Prediction Use Cases

1. Business problem: capacity planning (IT)
   - Forecast web traffic and more
     - To estimate server usage in the future to improve resource allocation
2. Security threats (Security)
   - Identify patterns of anomalous behavior in network, firewall and ecommerce data
     - To target threats and block bad actors from disrupting the system
3. Customer conversion rates (Marketing)
   - Identify patterns in customer behavior
     - To target strategies to those market segments to maximize conversion rates

What is Linear Regression?

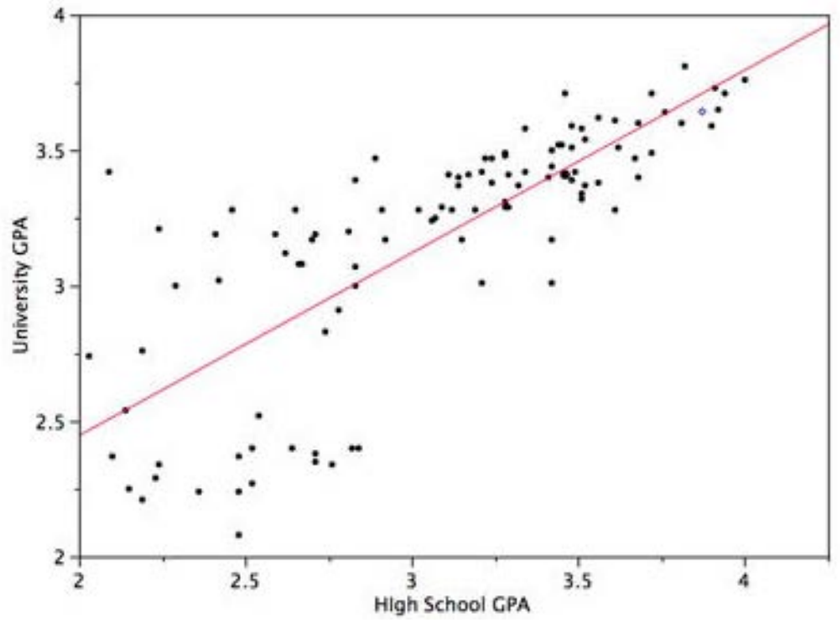
- Predict values on one variable from the values of a second variable
  - Variable you are predicting is the criterion variable, Y
  - Variable that is the basis from which you predict is the predictor, X
- If there is only one predictor, the prediction method is called simple regression



Assumptions in Linear Regression

To infer, or draw conclusions about data from a sample, assumptions include:

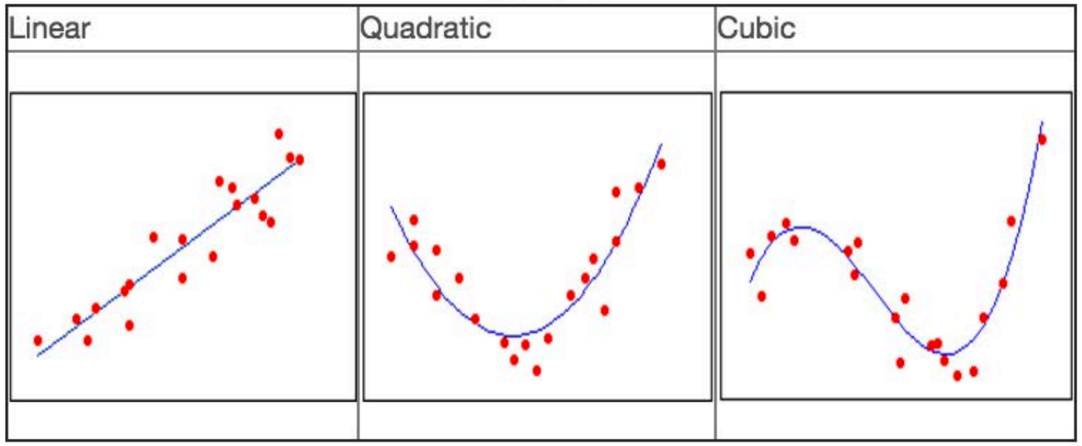
1. Linearity: The relationship between the two variables is linear
2. Homoscedasticity: The variance around the regression line is the same for all values of X
3. Errors of prediction are distributed normally (deviations from regression line are normal, not that X or Y is distributed normally)



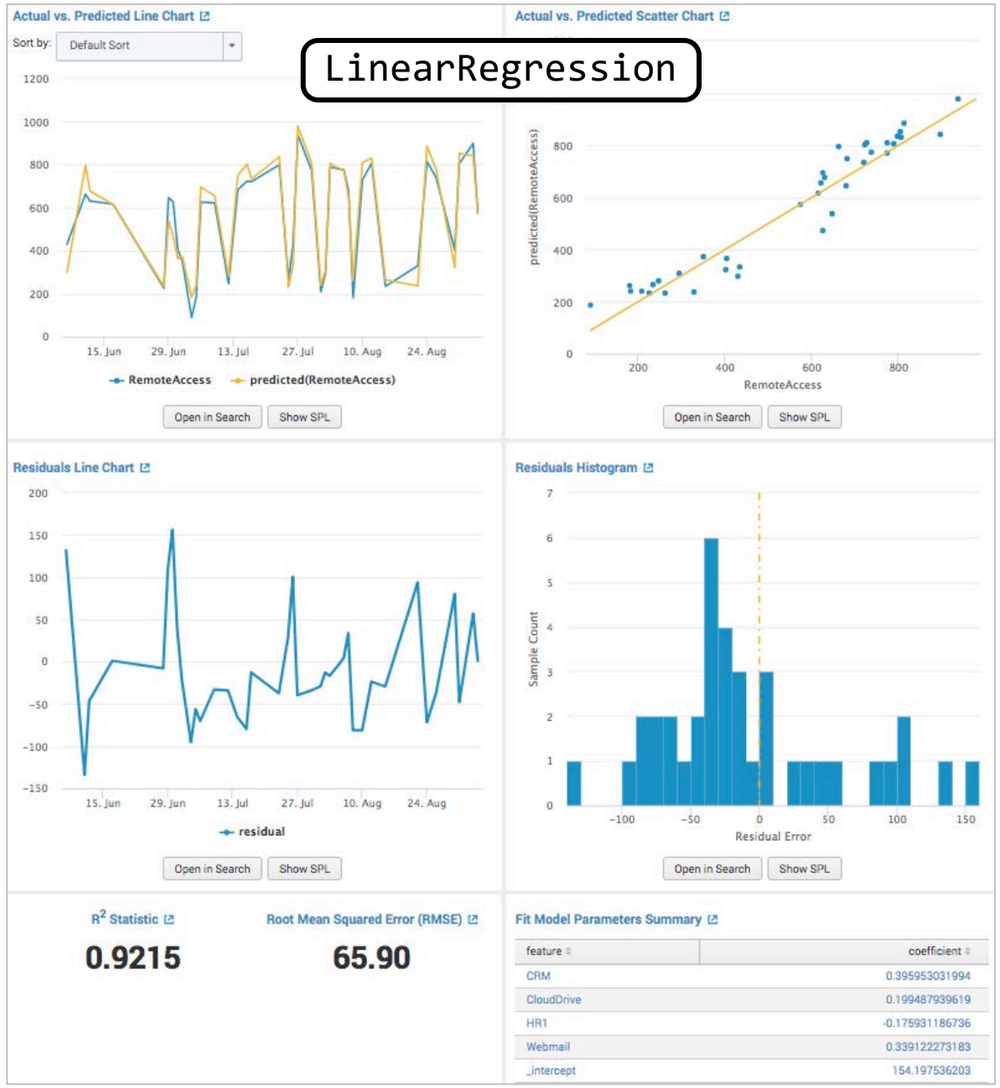
http://onlinestatbook.com/2/regression/inferential.html

Linear Regression with Curves

- If the relationship between X and Y is not linear, multiply fields or transform the values to allow curve fitting that would otherwise require nonlinear regression
- Use the ML Toolkit and Showcase to fit linear regression models with _time as an independent variable



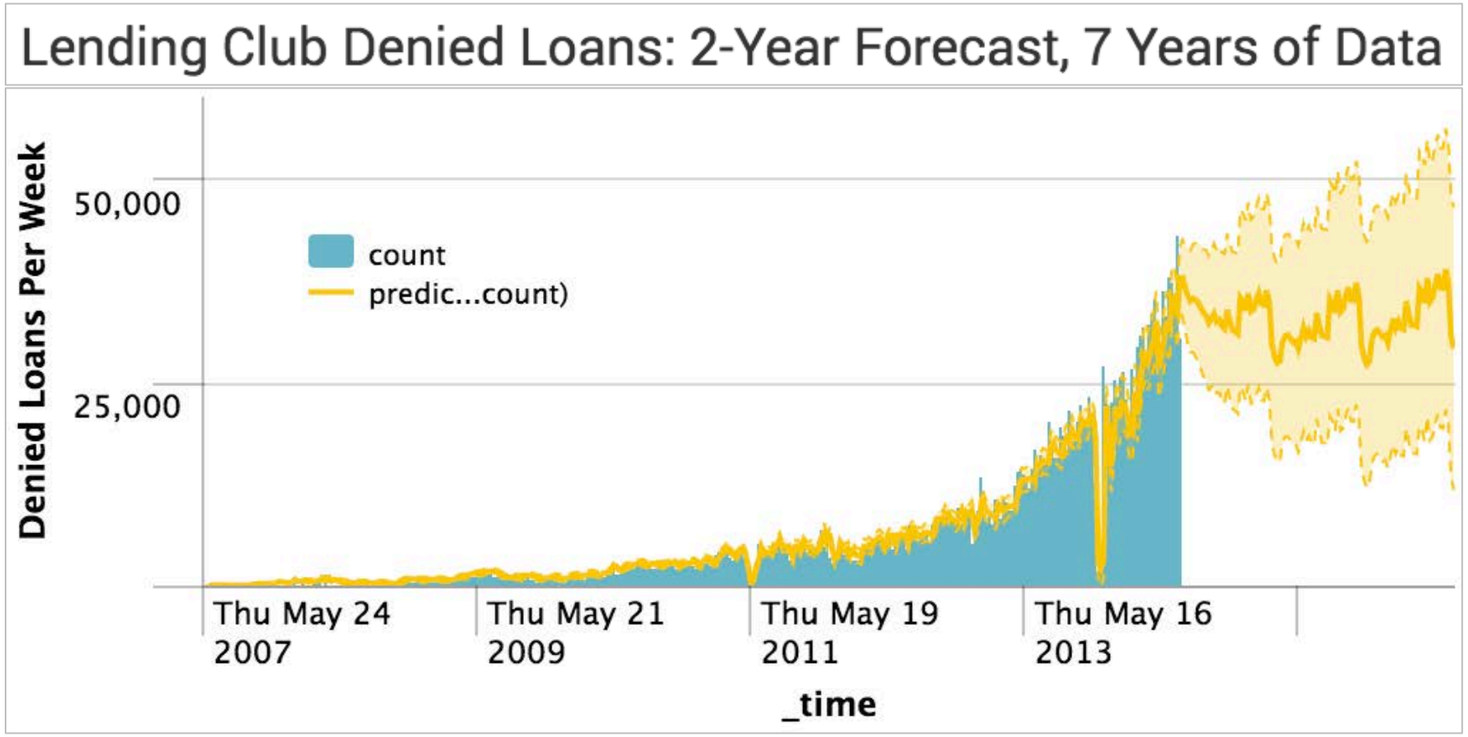
Comparing Algorithms



`predict` Command

- Forecasts trajectories of time series
  - Uses Kalman filter to identify seasonal trends
  - Gives a confidence interval as a buffer around the trend
- Uses lots of past data
- Includes low & high-frequency trends

```
sourcetype=lending_club_denied_loan
| timechart span=7d count
| predict count future_timespan=104
```



`predict` Command Algorithm Options

Model (Variations on the Kalman filter)	**Code**	**Univariate**	**Bivariate**	**Trends**	**Seasonality**
Local Level	LL	Yes	No	No	No
Seasonal Local Level	LLP	Yes	No	No	Yes
Local Level Trend	LLT	Yes	No	Yes	No
Bivariate Local Level	BiLL	No	Yes	No	No
Combo of LLT LLP	LLP5	Prediction and confidence interval based on weighted averages of LLT and LLP			

- Some predictions fall outside the confidence interval because:
  - The confidence interval does not cover 100% of the predictions
  - The confidence interval describes probabilities

`predict` Optional Arguments

predict <variable_to_predict> [AS <newfield_name>] [<predict_option>]

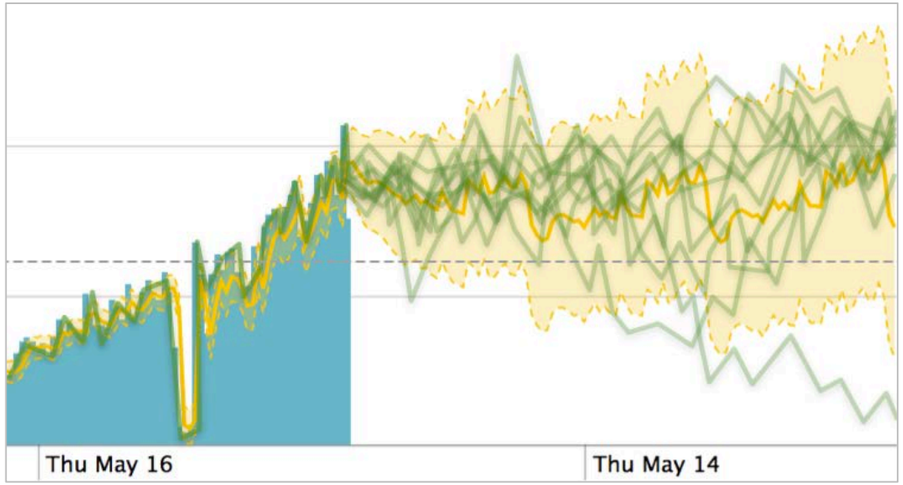
`<newfield>` (string) renames `<variable_to_predict>`

<predict_option> options can be specified anywhere, in any order

- `correlate=<field>` for bivariate, field to correlate against
- `future_timespan=<#>` length of prediction, not used for LLB
- `holdback=<#>` data points from the end NOT used in the model
- `upper and lower lower<int>=<field>` field name for upper and lower **<int>** percentage confidence interval **<int>** is 0-100; Defaults to lower95 and upper95
- `period=<num>` length of recurring cycle, an event=unit, default=none

Interpreting the `predict` Command

- Interpreting **`predict`** is subtle
  - Probabilities (of most likely paths) concentrate around best-fit curve
  - Avoid choosing a single “best prediction”
- Uncertainty envelope (95% confidence)
  - With high probability, “true future path” stays *mostly* within
  - <=5% of the time it doesn’t

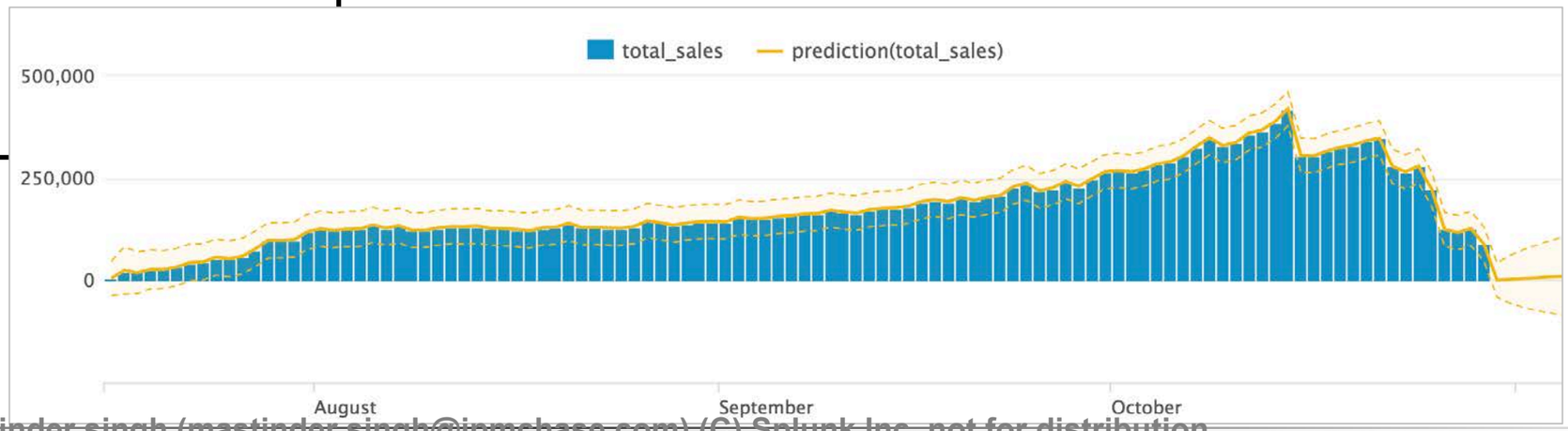


`predict` Command Example

Minimum of 2 - 4 cycles/seasons/periods of data

- To forecast quarterly, include at least 2 - 4 previous quarters
- Predict counts every time segment in your span as 1 data point
- More data? Go farther back in time and/or choose a finer span

```
sourcetype=access_combined
| timechart span=1d sum(price)
as total_sales
| predict total_sales
```



Module 7 Lab Exercise –Estimate and Predict

Time: 35 – 45 minutes

Tasks:

- Fit a linear regression to predict one field value from another
- Evaluate the “goodness of fit” for the models you just saved
- Add more features to the linear regression models

Module 8: Classification

Module Objectives

- Define key classification terms
  - True positives and true negatives
  - False positives and false negatives
  - Precision, recall, and F1 score
- Evaluate classifier results and tradeoffs
- Make classifications using The ML Toolkit and Showcase for Splunk. Some examples include:
  - Logistic regression
  - Support Vector Machines (SVM)

True Positives & Negatives

- 100% accuracy is not achievable. Results are a mix of:

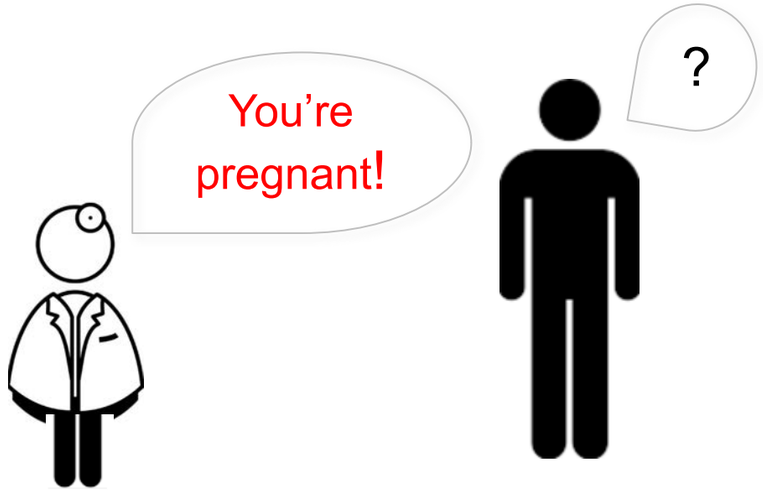
False Positives		**False Negatives**	
Model **inaccurately** predicts	an **existence or occurrence** in a group	Model **inaccurately** predicts	a non-existence or non-occurrence in a group

True Positives		**True Negatives**	
Model **accurately** predicts	an **existence or occurrence** in a group	Model **accurately** predicts	a non-existence or non-occurrence in a group

- Aim for a mix that allows for more of the error that causes less damage
- Ex: When detecting city bus passengers who didn't pay, more false negatives are likely to be tolerable than when detecting security breaches

Classifier Error Types

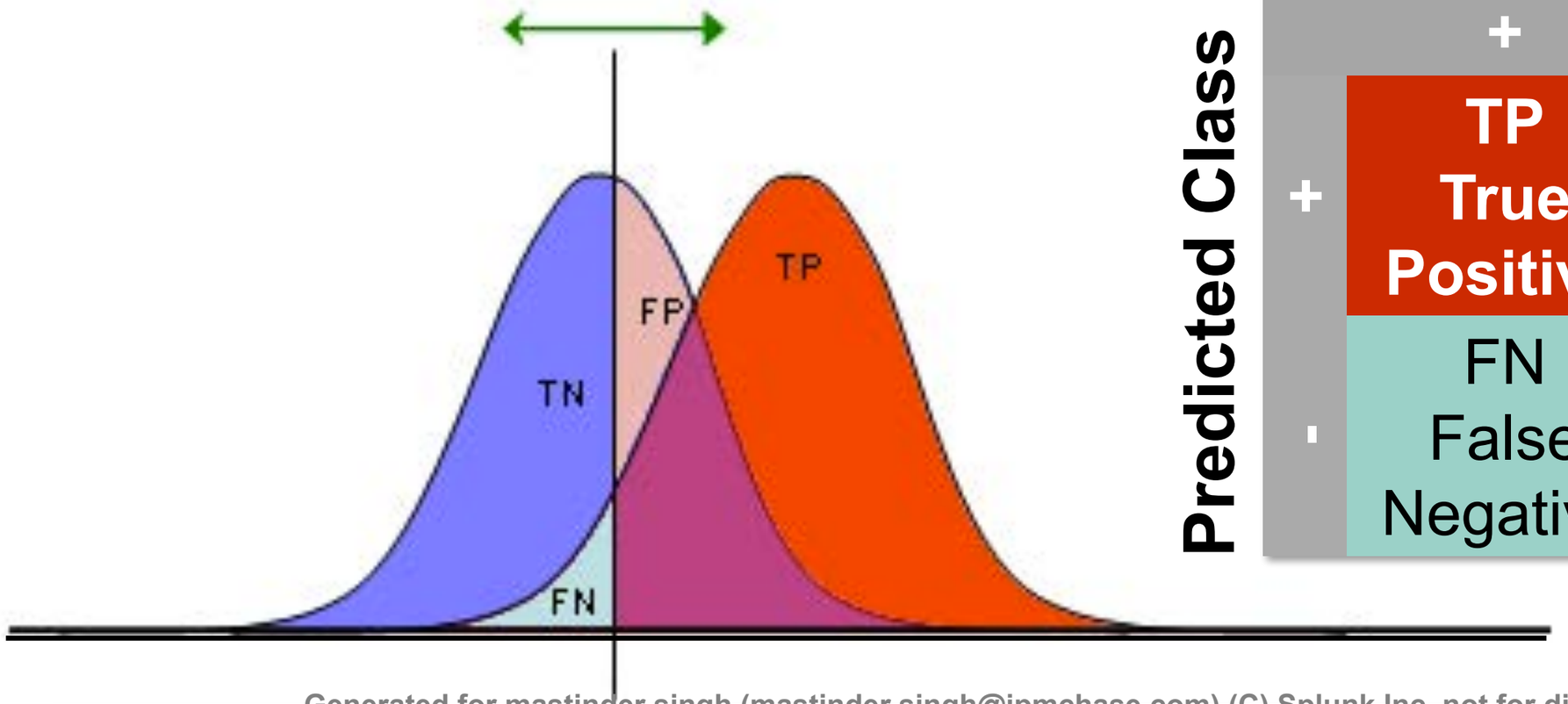
- Type I Error
  - Also called a false positive
    - Classifier falsely predicts a positive result
    - (data actually returns a negative result)



- Type II Error
  - Also called a false negative
    - Classifier falsely predicts a negative result
    - (data actually returns a positive result)

Classifier Results

- It can be helpful to depict the results of a classifier visually
  - The table at the right is called a confusion matrix
  - The graph plots a classifier's effectiveness



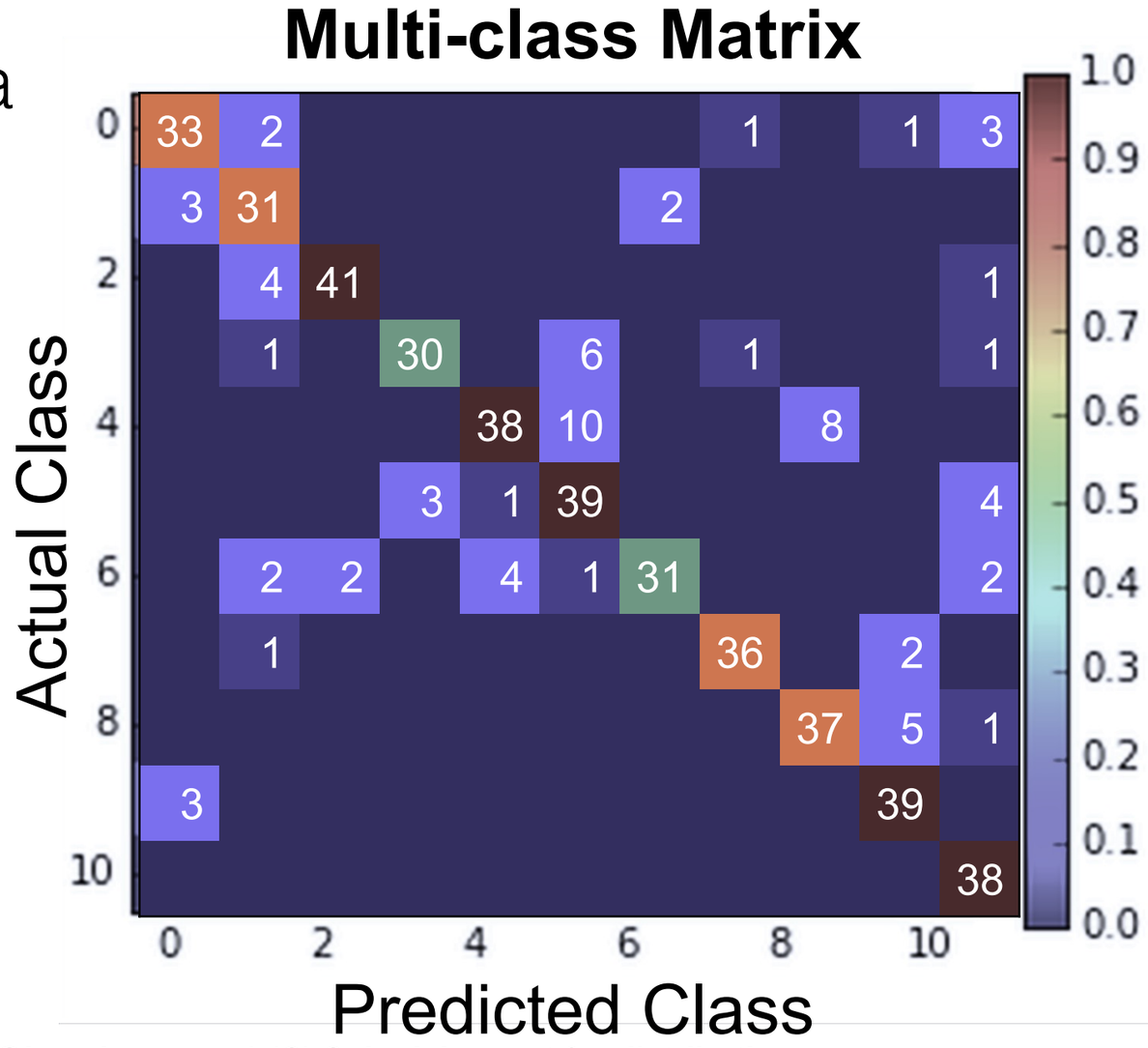
Predicted Class \ Actual Class	+	-
+	TP True Positive	FP False Positive
-	FN False Negative	TN True Negative

Confusion Matrices

- Useful with high dimensionality data
- Actual and predicted classes will match on one of the diagonals
  - In these examples, actual and predicted match on the upper left to lower right diagonal

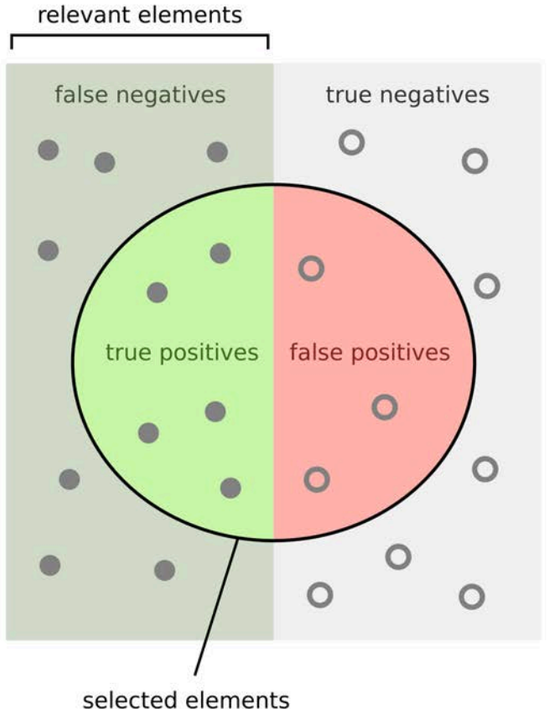
Binary Matrix

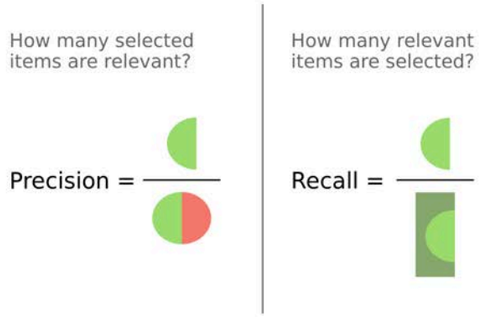
Actual \ Predicted	0	1
0	99	12
1	42	15



Evaluating Classifiers

- Testing a classification = evaluation
- Common classification evaluation metrics:
  - Precision (or specificity)
    - Given a positive example
    - How likely is it to be detected as in the cluster
  - Recall (or sensitivity)
    - Given a positive *prediction*
    - How likely is the prediction to be correct
  - F1 Score: harmonic mean of precision & recall
    - For large datasets with a few extreme outliers
    - Does *not* take true negatives into account





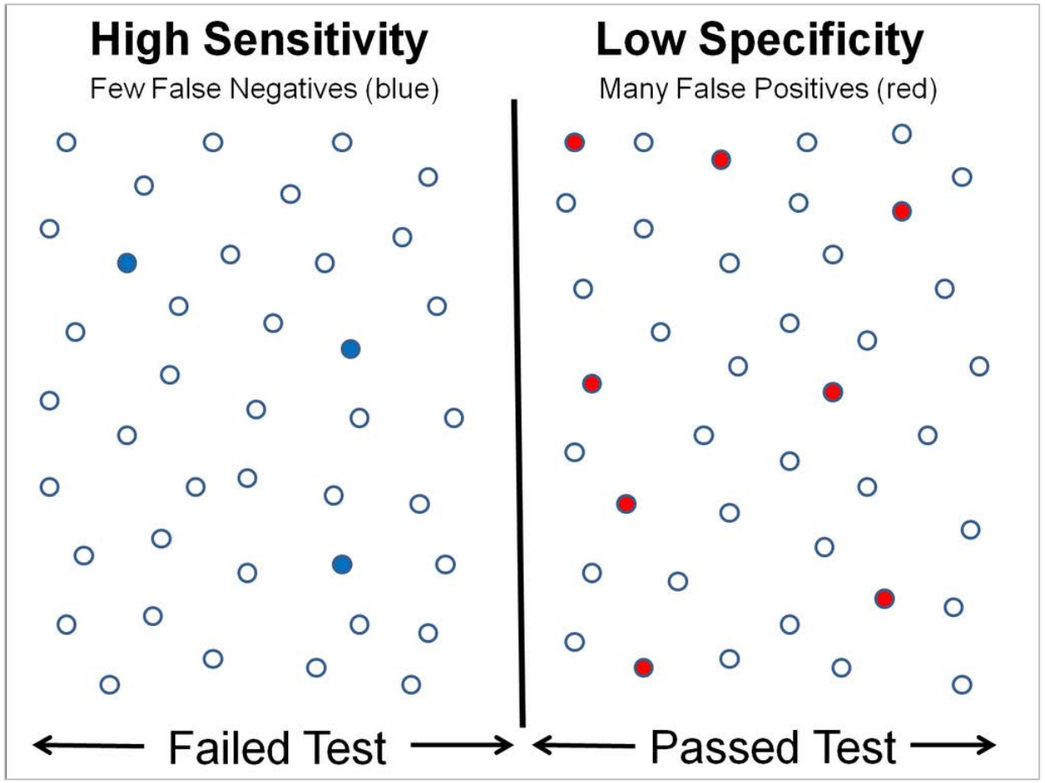
Calculating Metrics

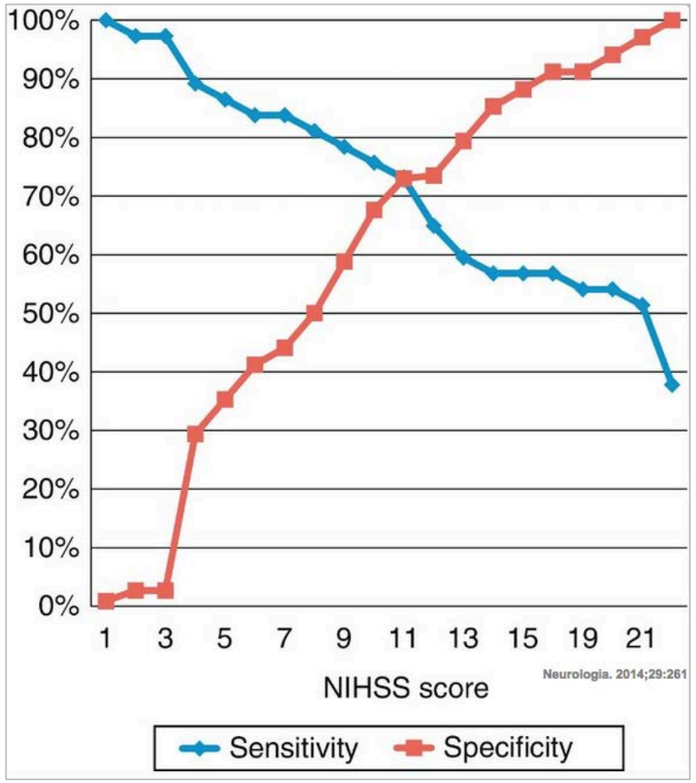
$$precision = \frac{tp}{tp + fp} \qquad recall = \frac{tp}{tp + fn}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

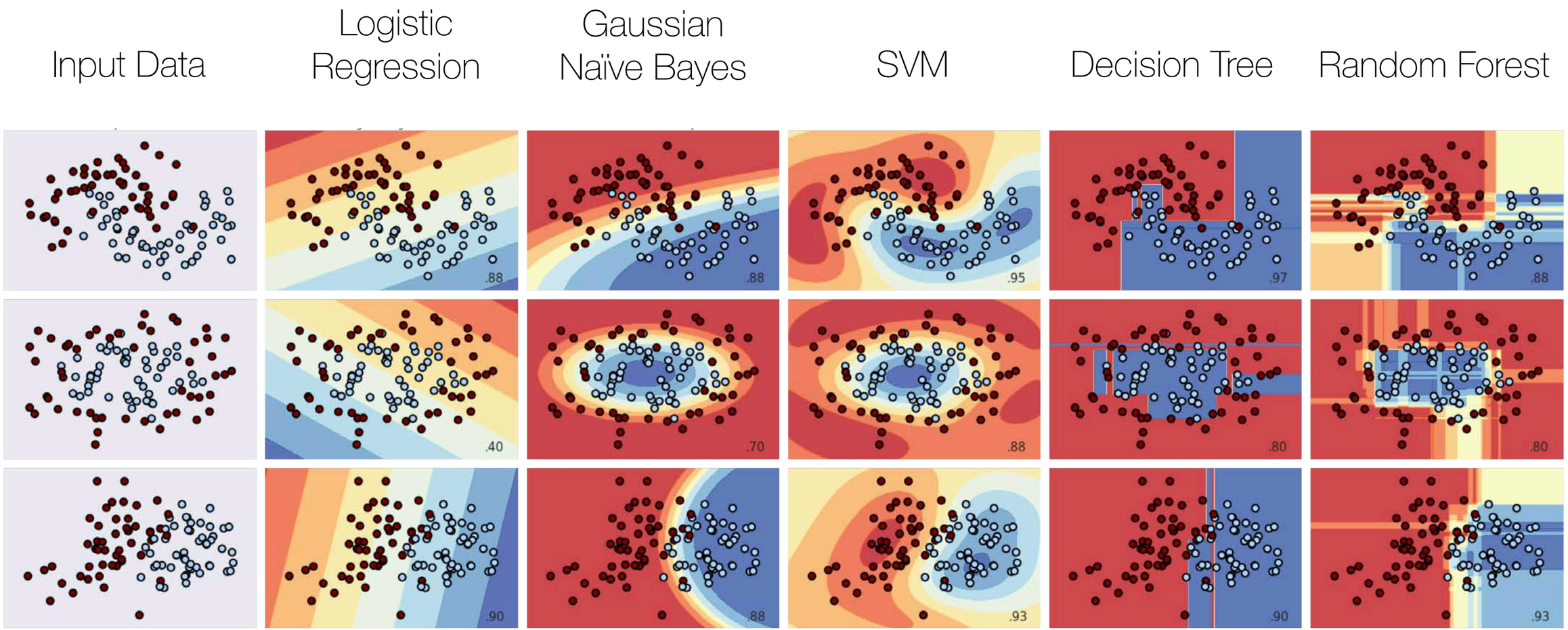
Prediction / Classifier Tradeoffs

- The ideal evaluation profile of a classifier depends on its context (cost, risk)
- Example: acceptable rate of unpaid bus tickets vs. undetected gas leaks



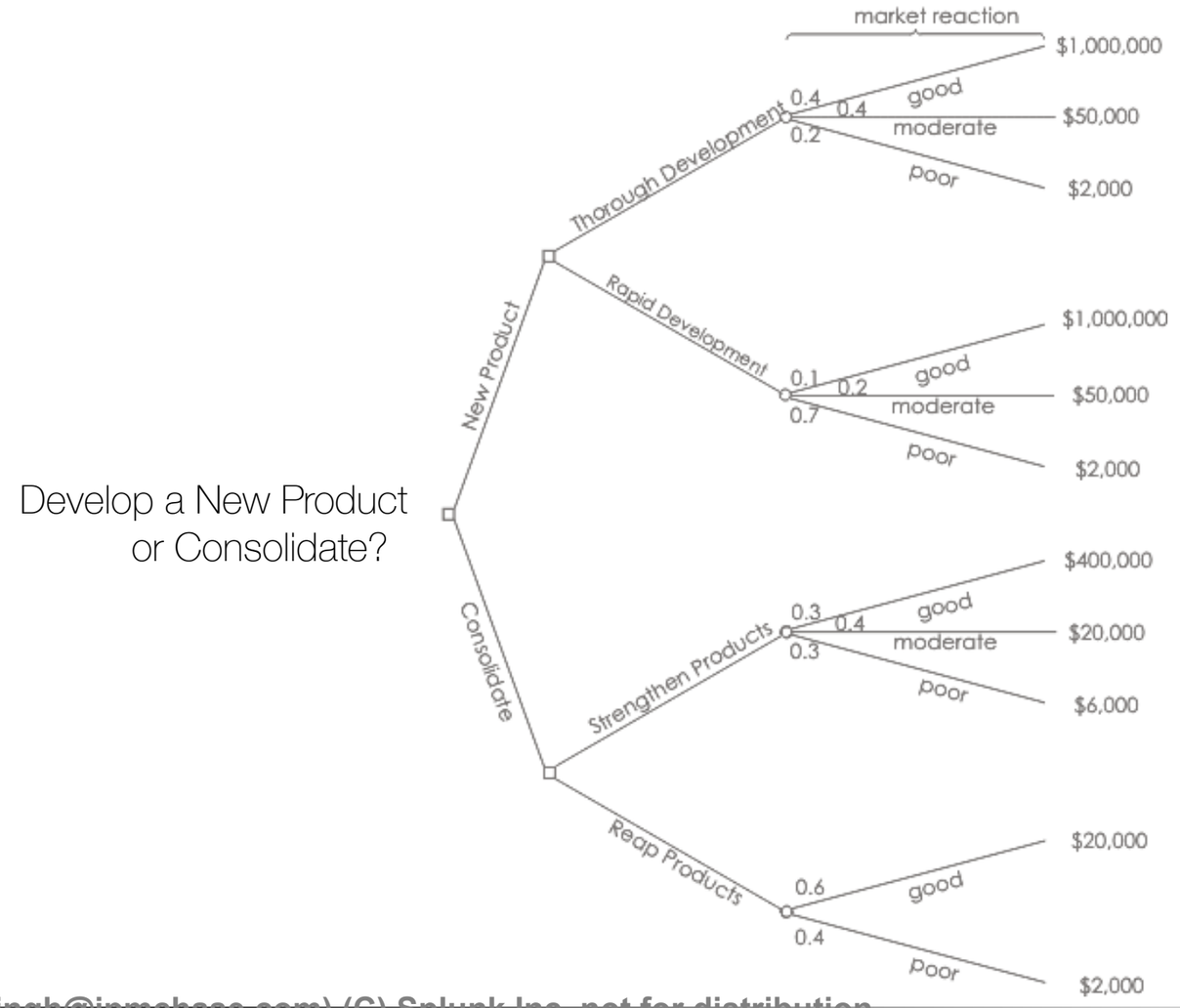


Common Algorithm Plots



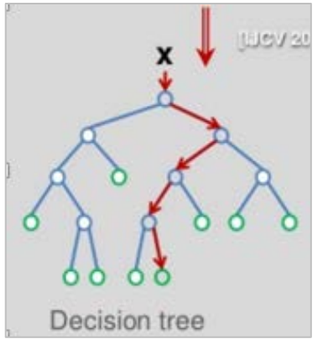
Decision Trees

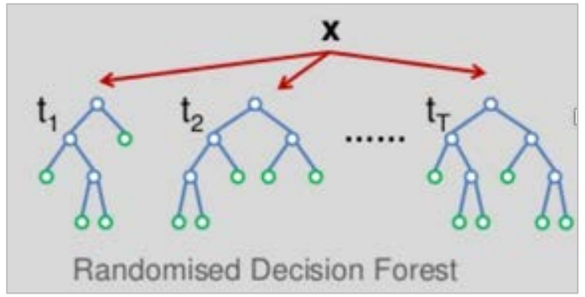
- Consists of nodes (splits), edges (branches), and (terminal) leaves
- Finds the attribute that returns the highest information gain (the most homogeneous branches)
  - Calculates the drop in entropy after a dataset is split based on attributes



Random Forest

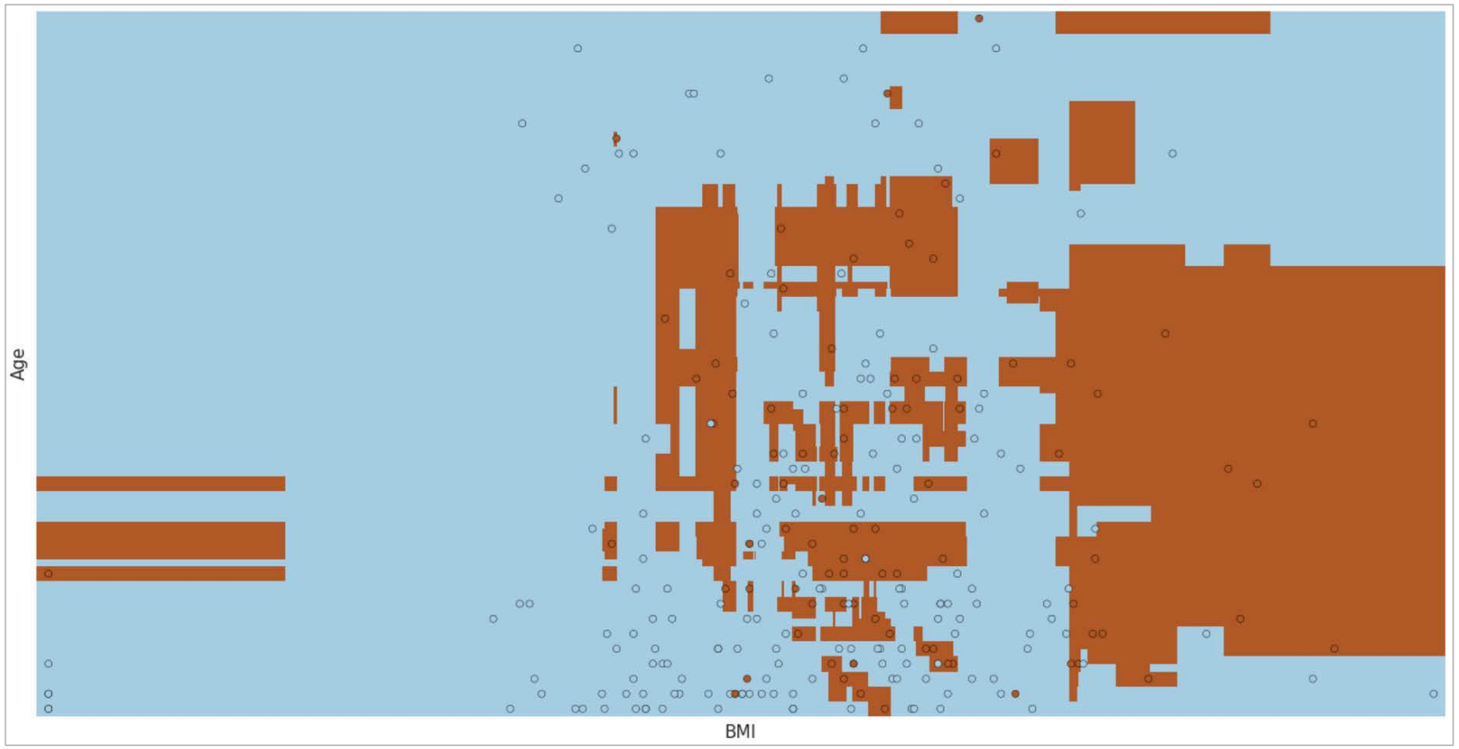
- Ensembles of decision trees: with big data, often used instead of decision trees due to its efficiency
- CART Classification and Regression Trees
  - Classification - split top-down by similarity
  - Regression – target variable is a real number
    - Fit a regression model to target variable
    - Use each independent variable
      - Find SSE (Sum of Squared Error) - predicted vs. actual, at each split point
      - Selects the best variable (minimum SSE) at each node to "decide"
  - Nonlinear multiple regression
  - Each leaf contains a distribution for the continuous output variable(s)





Random Forest Example

```
| inputlookup diabetes.csv
| sample partitions=3 seed=42
| search partition_number < 2
| fit RandomForestClassifier into RFC response from BMI age
```



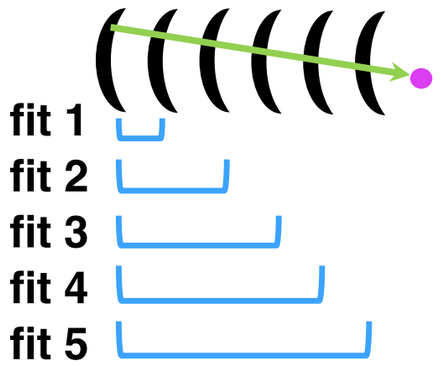
Random Forest Results

```
| inputlookup diabetes.csv
| sample partitions=3 seed=42
| search partition_number = 2
| apply RFC as prediction
| `confusionmatrix(response, prediction)`
```

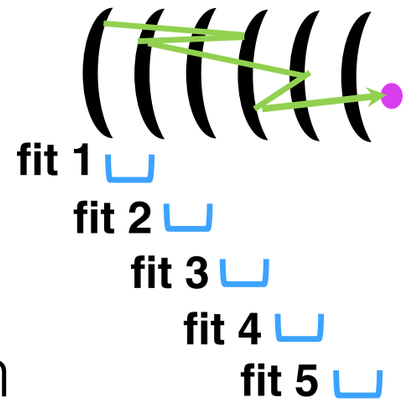
Predicted actual	Predicted 0	Predicted 1
0	110	48
1	58	42

SGD (Stochastic Gradient Descent)

- Gradient Descent
  - Each fit requires all the previous data
  - Not feasible for large datasets



- Stochastic Gradient Descent
  - The fit updates for each iteration
    - Only the new data is needed for each fit
    - Makes it possible to train algorithms on larger data sets
  - Tests the hypothesis on a single random example for each
    - Sometimes needs many iterations to complete
    - Often zig zags toward the global minimum



Naïve Bayes: `BernoulliNB` & `GaussianNB`

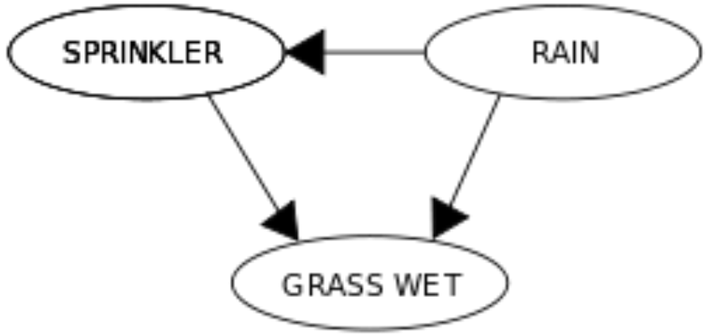
- Assumes that all features are conditionally independent
- Uses Bayesian rule for probability (evidence updates predictions)
  - To represent frequency, use **`GaussianNB`**
  - If the concern is presence or absence, use **`BernoulliNB`**
- Formerly called inverse probability because it infers backward from observations to parameters (from effects to causes)

Simple Naïve Bayes

For each known class value:

1. Calculate probabilities for each attribute based on the class value
2. Make joint conditional probability for attributes with the product rule
3. Use Bayes rule to derive conditional probabilities for the class
4. Find the class with the highest probability

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99



	RAIN	
T	F	
0.2	0.8	

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

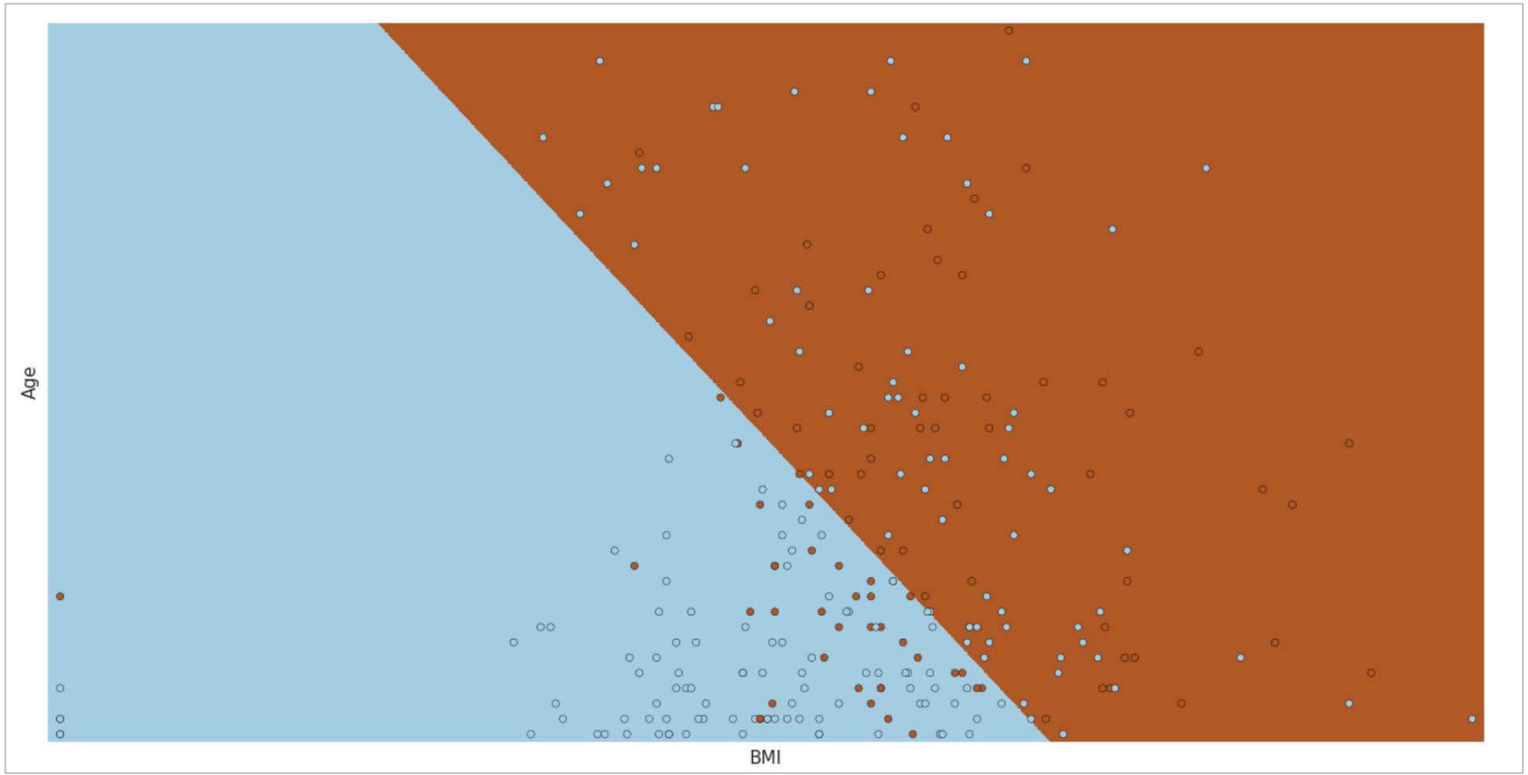
Naïve Bayes vs. Other Algorithms

- Effective in most cases, even when features are not independent
- Not as useful for very large datasets

	Mixed Data	Missing Values	Outliers	Monotone Transformations	Scalability	Irrelevant Input	Linear Combinations	Interpretable	Accurate
Logistic Regression	n	n	Y	n	Y	n	Y	Y	Y
Decision Trees	Y	Y	Y	Y	Y	some	n	Y	n
SVM	n	n	Y	n	n	some	Y	some	Y
Naïve Bayes	Y	Y	disc	disc	Y	some	Y	Y	Y

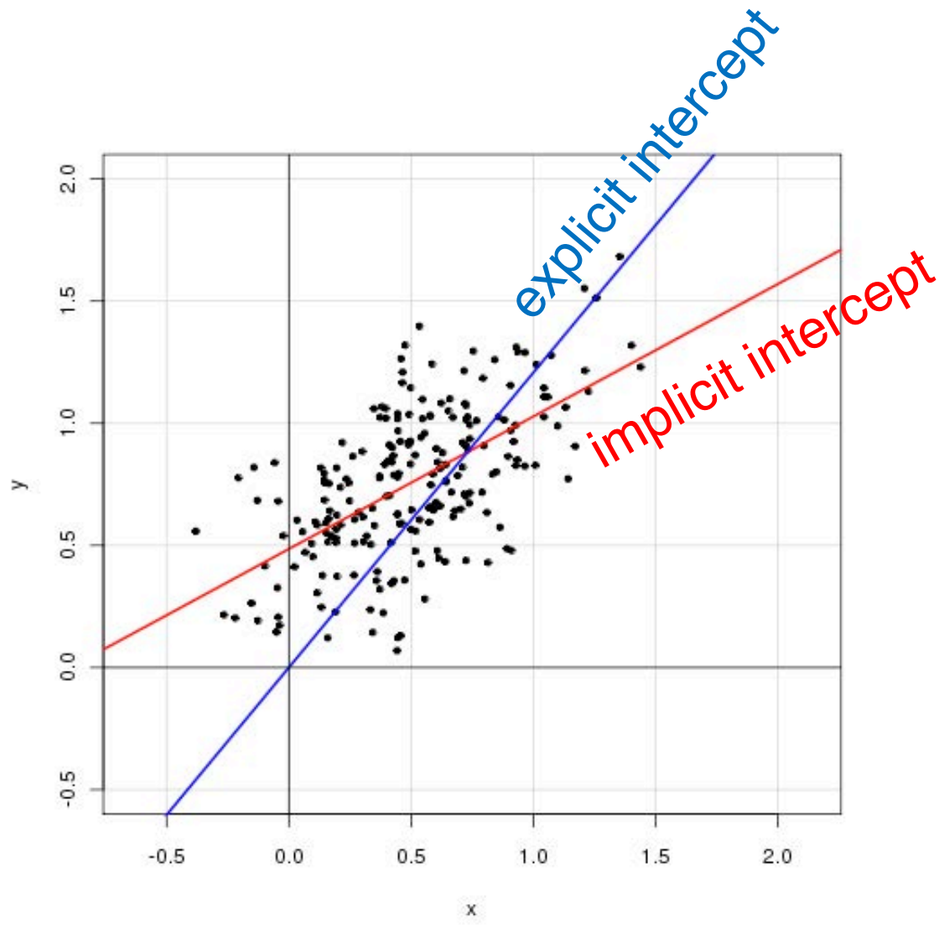
Logistic Regression Example

```
| inputlookup diabetes.csv
| sample partitions=3 seed=42
| search partition_number < 2
| fit LogisticRegression response from BMI age into LogisticRegressionClassifier
```



Logistic Regression Parameters

- `fit_intercept`
  - Specifies whether the model should include an implicit intercept term (default: "true," as models without an implied intercept are rare)
- `probabilities`
  - Specifies whether probabilities for each possible field value should be returned alongside the predicted value (the default value is "false")



Logistic Regression Probabilities

```
| inputlookup diabetes.csv
| sample partitions=3 seed=42
| search partition_number < 2
| fit LogisticRegression response from BMI age into LogisticRegressionClassifier probabilities=t
```

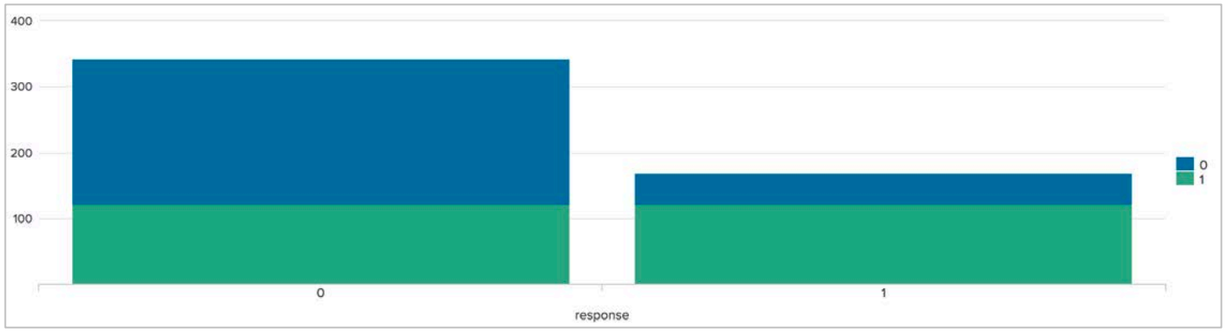
BMI	age	blood_pressure	diabetes_pedigree	glucose_concentration	insulin	partition_number	predicted(response)	probability(response=0)	probability(response=1)	response	skin_thickness
33.6	50	72	0.627	148	0	1	1	0.334389520514	0.665610479486	1	35
26.6	31	66	0.351	85	0	0	0	0.646027193849	0.353972806151	0	29
23.3	32	64	0.672	183	0	0	0	0.700323646094	0.299676353906	1	0
28.1	21	66	0.167	89	94	0	0	0.69746610428	0.30253389572	0	23
35.3	29	0	0.134	115	0	0	1	0.481707559449	0.518292440551	0	0
30.5	53	70	0.158	197	543	1	1	0.370316931261	0.629683068739	1	45

Confusion Matrix Macro

```
| inputlookup diabetes.csv
| sample partitions=3 seed=42
| search partition_number = 2
| apply LogisticRegressionClassifier as prediction
| chart count by response, prediction
```

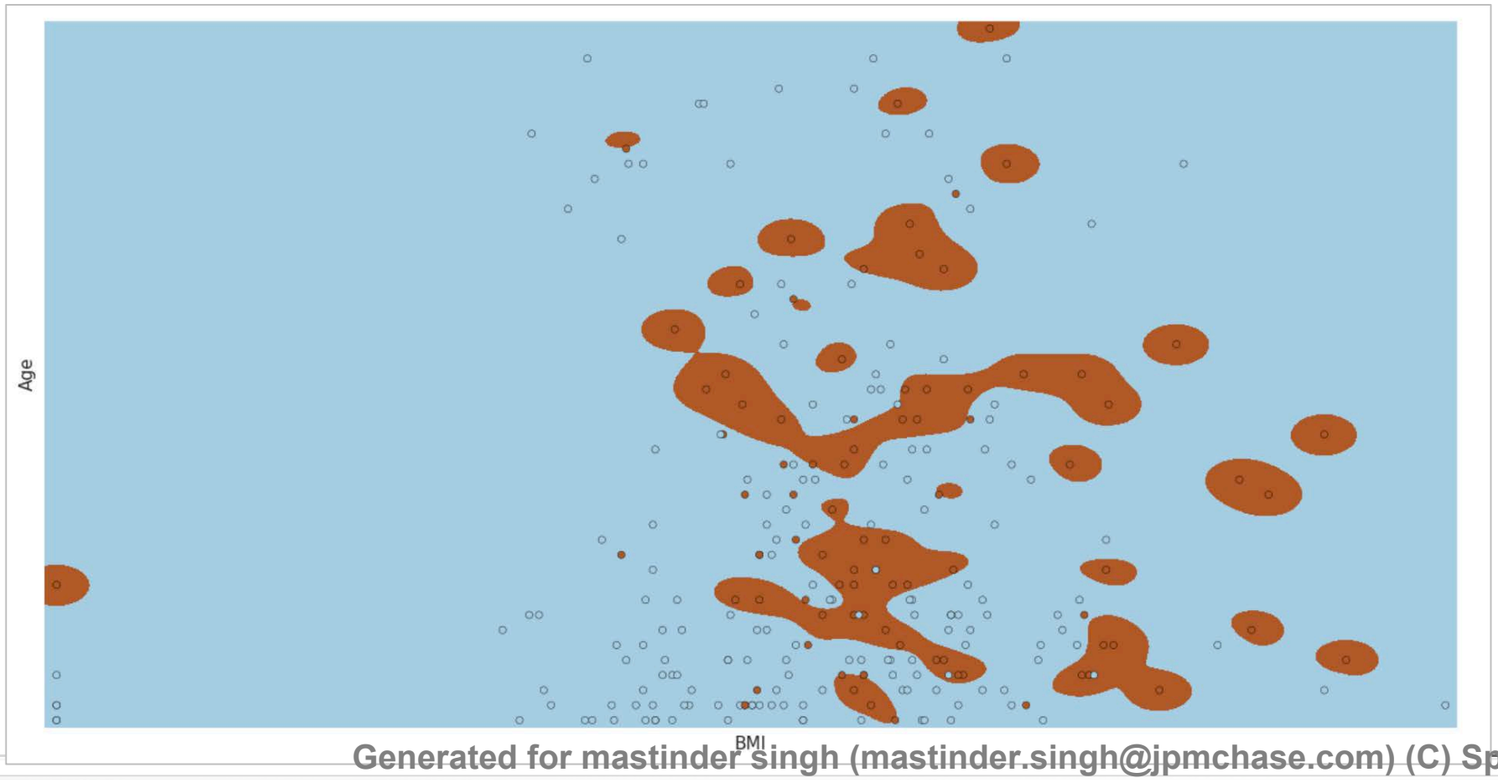
response	0	1
0	220	122
1	46	122

```
| inputlookup diabetes.csv
| sample partitions=3 seed=42
| search partition_number = 2
| apply classifier as prediction
| `confusionmatrix(response, prediction)`
```



Support Vector Machine Example

```
| inputlookup diabetes.csv
| sample partitions=3 seed=42
| search partition_number < 2
| fit SVM into svm_model response from BMI age
```



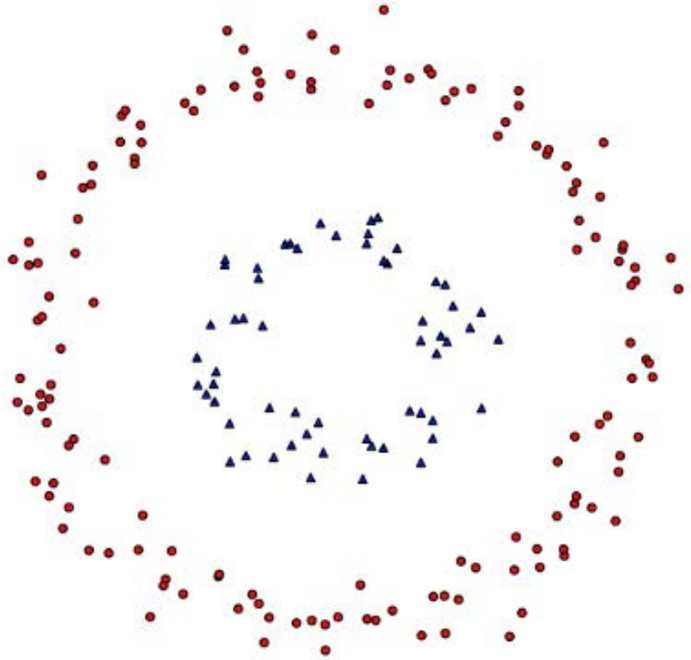
Support Vector Machine Results

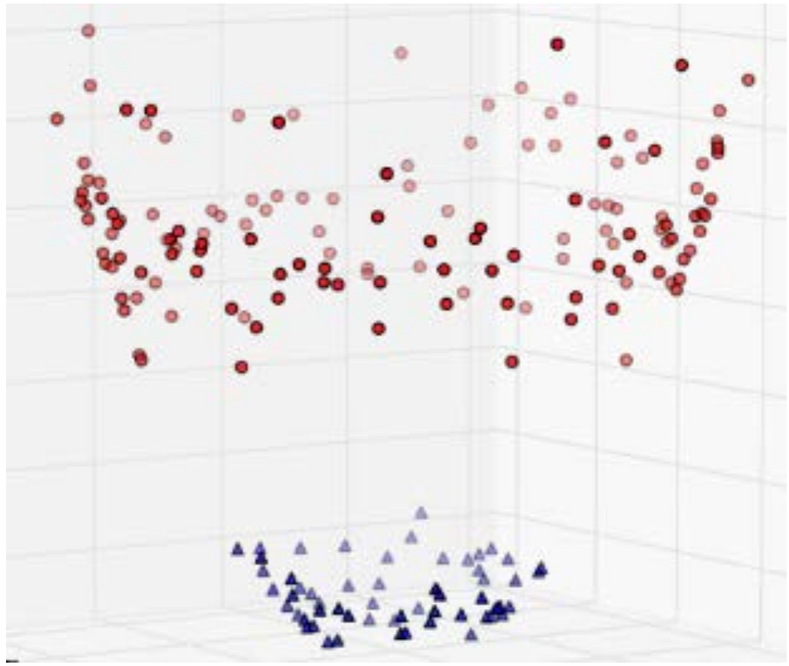
```
| inputlookup diabetes.csv
| sample partitions=3 seed=42
| search partition_number = 2
| apply svm_model as prediction
| `confusionmatrix(response, prediction)`
```

Predicted actual	Predicted 0	Predicted 1
0	92	66
1	34	66

SVM Kernel Example

- Data that is not linearly separable (left) could be made separable in a high-dimensional space (right), but would be prohibitively intensive to represent
- Replace dot products with a kernel function to implicitly work in higher dimensional space without building the representation
- The SVM can learn a nonlinear decision boundary in the original, which corresponds to a linear decision boundary in higher dimensional “space”





Module 8 Lab Exercise – Classify Events & Evaluate

Time: 20 – 30 minutes

Tasks:

- Cluster (classify) based on field values
- Modify classification criteria
- Examine the results
- Visualize classifications
- Create a classifier for another type of data

Support Programs

- **Splunkbase Answers:** answers.splunk.com
  Post specific questions and get them answered by Splunk community experts.
- **Splunk Docs:** docs.splunk.com
  These are constantly updated. Be sure to select the version of Splunk you are using.
- **Wiki:** wiki.splunk.com
  A community space where you can share what you know with other Splunk users.
- **IRC Channel:** #splunk on the EFNet IRC server
  Many well-informed Splunk users “hang out” here.

.conf18™

splunk>

.conf18:
Monday, October 1 – Thursday, October 4

Splunk University:
Saturday, September 29 – Monday, October 1

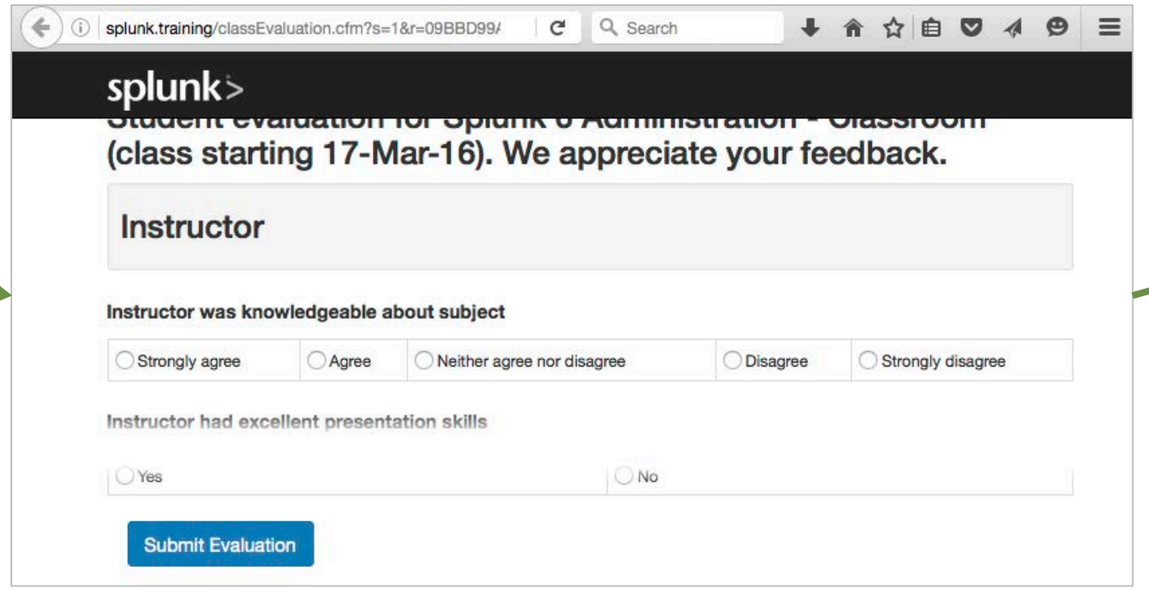
ORLANDO FLORIDA

Walt Disney World Swan and Dolphin Hotels

Thank You

Complete the Class Evaluation to be in this month's drawing for a $100 Splunk Store voucher

1. Look for the invitation email, *What did you think of your Splunk Education class*, in your inbox
2. Click the link or go to the specified URL in the email



Thank You

Appendix

Further Reference

- Python for Scientific Computing app (required for ML Toolkit and Showcase app)
  - Mac: https://splunkbase.splunk.com/app/2881/
  - Linux 64-bit: https://splunkbase.splunk.com/app/2882/
  - Linux 32-bit: https://splunkbase.splunk.com/app/2884/
  - Windows 64-bit: https://splunkbase.splunk.com/app/2883/
- Machine Learning Toolkit and Showcase app

  https://splunkbase.splunk.com/app/2890/
- Machine Learning Toolkit documentation & Cheat Sheet

  http://docs.splunk.com/Documentation/MLApp/3.2.0/User/About

  https://www.splunk.com/pdfs/solution-guides/machine-learning-quick-ref-guide.pdf

Macros

Name	Definition	Arguments
classificationreport(2)	rename "a" as actual, "p" as predicted | stats count by actual predicted | xyseries actual predicted count | fillnull | untable actual predicted count | stats sum(eval(if(actual == predicted, count, 0))) as t sum(eval(if(actual != predicted, count, 0))) as f by actual predicted | eventstats sum(eval(if(actual==predicted, t, 0))) as tp sum(eval(if(actual!=predicted, f, 0))) as fn by actual | eventstats sum(eval(if(actual!=predicted, f, 0))) as fp by predicted | eval count=if(actual==predicted,t,f), fp=if(actual==predicted, fp, 0), fn=if(actual==predicted, fn,0), tp=if(actual==predicted, tp, 0) | stats sum(count) as count sum(t) as t sum(f) as f sum(tp) as tp sum(fn) as fn sum(fp) as fp by actual | eventstats sum(count) as total | eval precision=tp/(tp+fp) | eval recall=tp/(tp+fn) | eval f1=2*precision*recall/(precision+recall) | eval accuracy=t/count | fillnull | appendpipe [stats sum(count) as count sum(eval(accuracy*count/total)) as accuracy sum(eval(precision*count/total)) as precision sum(eval(recall*count/total)) as recall sum(eval(f1*count/total)) as f1 | eval actual="Weighted Average"] | rename actual as class | table class accuracy precision recall f1 count	a, p
classificationstatistics(2)	`classificationreport("$a$", "$p$")` | tail 1 | eval precision=round(precision, 2), recall=round(recall, 2), accuracy=round(accuracy, 2), f1=round(2*precision*recall/(precision+recall), 2)	a, p
confusionmatrix(2)	rename "a" as actual, "p" as predicted | stats count by actual predicted | appendpipe [eval predicted=actual, count=0] | stats sum(count) as count by actual predicted | xyseries actual predicted count | rename * as "Predicted *" | rename "Predicted a" as "Actual a" | fillnull value=0	a, p
forecastviz(4)	eval _ft=ft, _hb=hb, _var="v", _ci=ci	ft, hb, v, ci
histogram(2)	bin "var" bins=$bins$ | stats count by "var" | makecontinuous "var" | fillnull	var, bins
modvizpredict(6)	predict "v" as prediction algorithm=a future_timespan=f holdback=h p lowerci=lowerci upperci=upperci | eval _ft=f, _hb=h, _var="v", _ci=ci	v, a, f, h, p, ci

Macros (cont.)

Name	Definition	Arguments
regressionstatistics(2)	rename "a" as _actual, "p" as _predicted | eventstats avg(_actual) as _avgActual | eval _actualMinusAvg = _actual - _avgActual, _residual = _actual - _predicted | stats sumsq(_actualMinusAvg) as _sumsqActualMinusAvg, sumsq(_residual) as _sumsqResidual, count(_residual) as _sampleCount | eval rSquared = round(1 - _sumsqResidual / _sumsqActualMinusAvg, 4), RMSE = round(sqrt(_sumsqResidual / _sampleCount), 2) | table rSquared RMSE	a, p
splitby(1)	eval _split_by="s"	s
splitby(2)	eval _split_by=mvappend("$s1$", "$s2$")	s1, s2
splitby(3)	eval _split_by=mvappend("$s1$", "$s2$", "$s3$")	s1, s2, s3
splitby(4)	eval _split_by=mvappend("$s1$", "$s2$", "$s3$", "$s4$")	s1, s2, s3, s4
splitby(5)	eval _split_by=mvappend("$s1$", "$s2$", "$s3$", "$s4$", "$s5$")	s1, s2, s3, s4, s5

Example Projects

- I want to know when I'll need to buy more hardware
- I want to know the number of concurrent customers I can service
- How can I determine which component in a server / system / datacenter has my highest rate of failure, dollar adjusted
- I'm looking at a trend line for foo. I want splunk to alert me when it thinks foo will go beyond x in y minutes
- App crash forecasting
- Network monitoring
- Transactional monitoring
- Anomalous kpi detection

`analyzefields` Command

Finds the fields that best predict the value of a field you specify

- Assumes normal distribution
- **`classfield=`** signifies the field whose value you want to predict
- **`field`** should be a discrete random variable
  - Does not need to be binary, but is often helpful if it is
- Can be abbreviated to **`af`**

```
analyzefields
classfield=<field>
```


`analyzefields` Results

Returns a table with 5 columns

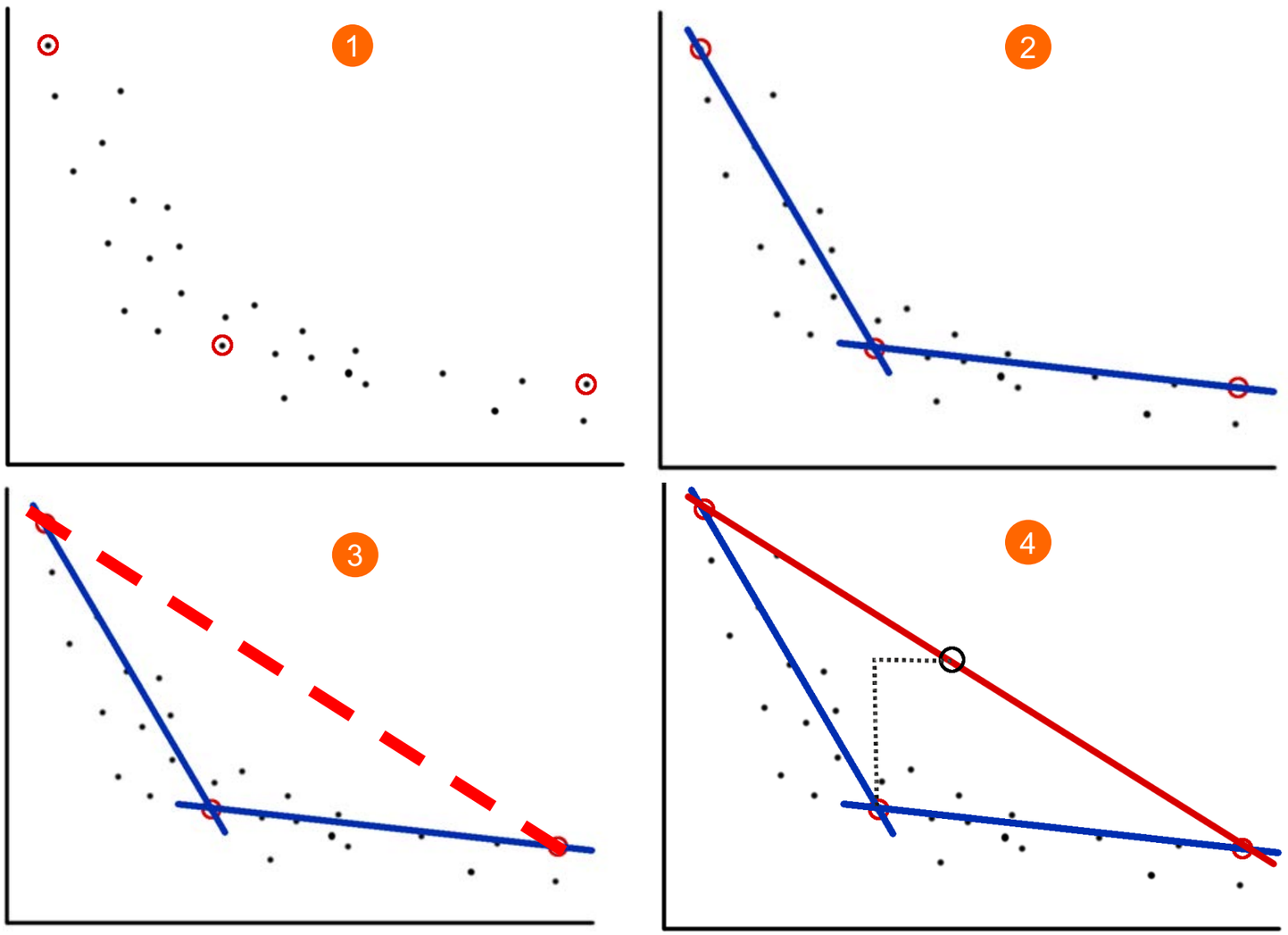
- **`Field`** a field from your search results that may predict your **`classfield`** field
- **`Count`** number of times that **`field`** occurs in your search results
- **`Cocur`** ratio of field and **`classfield`** both occurring (1=all events)
- **`Acc`** how accurately **`field`** value predicts **`classfield`** value (accurate predictions divided by total number of events)
- **`Balacc`** non weighted average of accuracies (mean of true positive and true negative rates)

analyzefields Example

```
sourcetype=cisco_wsa_squid
| eval violation = if(usage="violation", 1, 0)
| analyzefields classfield=violation
```

field	count	cocur	acc	balacc
Department				
Email				
First_Name				
Last_Name				
Username				
bytes_in	2490	1	0.9983935742971888	0.9983935742971888
change_type				
date_hour	2490	1	1	1
date_mday	2490	1	1	1
date_minute	2490	1	1	1
date_second	2490	1	1	1
date_year	2490	1	1	1
date_zone	2490	1	1	1
end_time	2490	1	1	1
linecount	2490	1	1	1
sc_bytes	2490	1	0.9983935742971888	0.9983935742971888

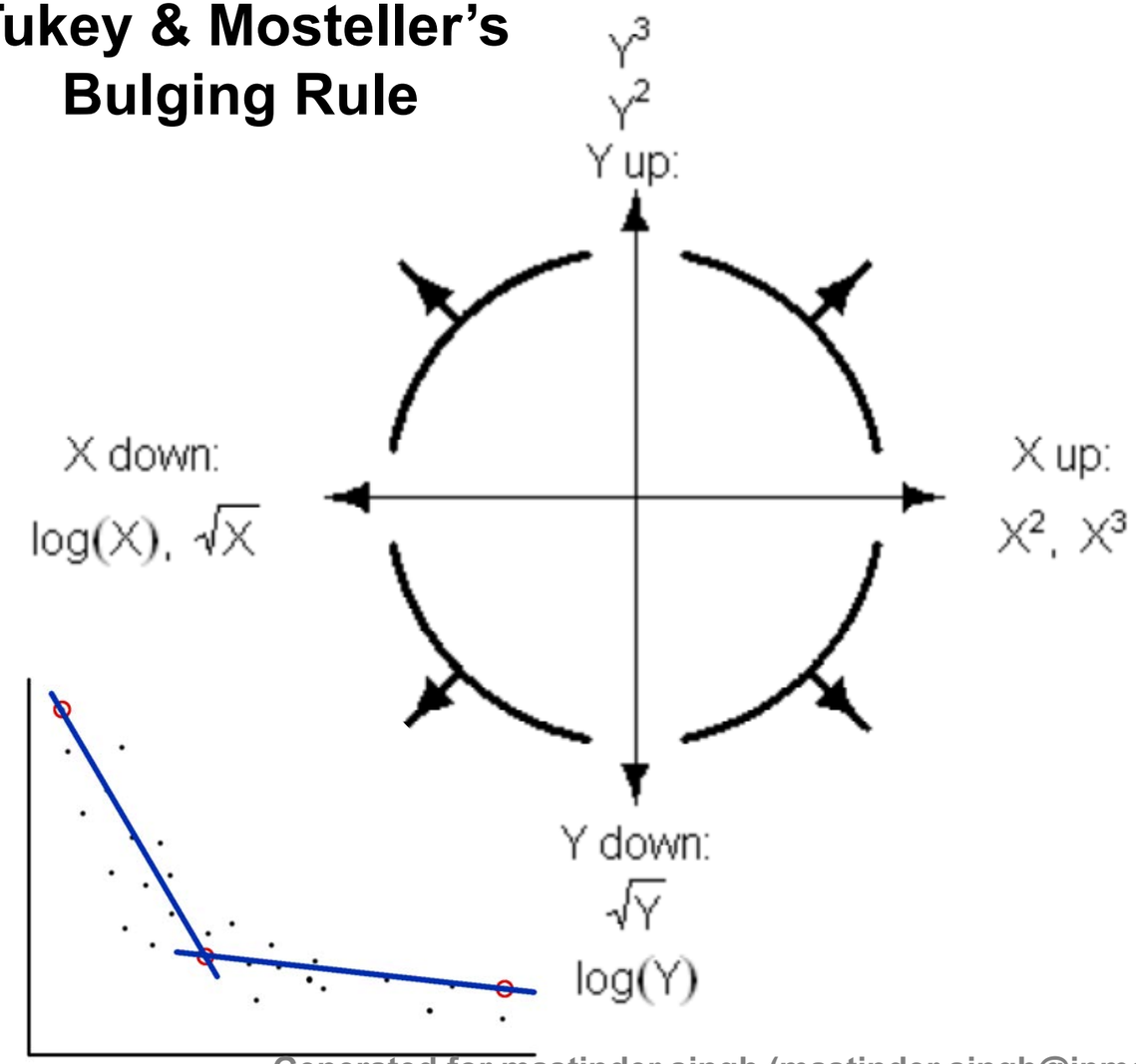
Linearizing: Tukey's Visual Rule of Thumb



1. Split the data into 3 roughly even groups by choosing a few reference points
2. Draw lines through the reference points
3. Imagine the ideal line through the data
4. Describe the middle point's offset from the ideal line in terms of x, y, up, and down.
   - In this case, the dotted line is x down, y down

Find the Direction of the Transformation

Tukey & Mosteller's Bulging Rule



- Match the curve of the diagram to the curve of the data
- The curve of the dataset graphed is most similar to the bottom left of the diagram, so this dataset's offset from the ideal line is x down and y down

Ladder of Powers Calculations

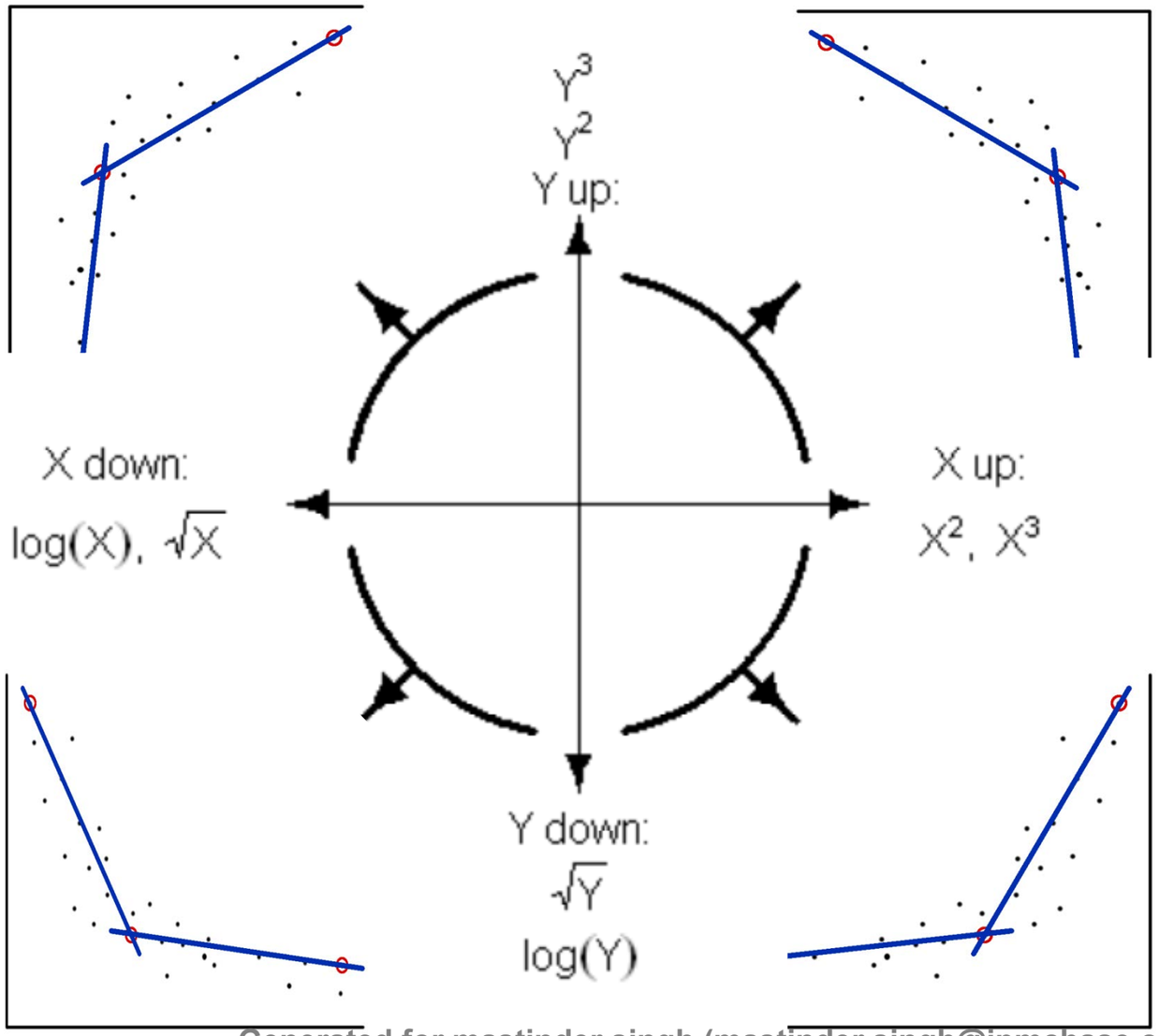
$$y = \begin{cases} x^{\lambda} & \text{if } \lambda > 0 \\ \log x & \text{if } \lambda = 0 \\ -(x^{\lambda}) & \text{if } \lambda < 0 \end{cases}$$

λ	-2	-1	-1/2	0	1/2	1	2
y	$\frac{-1}{x^2}$	$\frac{-1}{x}$	$\frac{-1}{\sqrt{x}}$	$\log x$	$\sqrt{x}$	x	x^2

Modify λ (lambda) to go up or down the ladder of powers

- Increase λ to go up the ladder, decrease λ to go down
- Generally, λ > 0
- If λ < 0, then flip the sign to preserve the relationship between x and y

Tukey & Mosteller's Bulging Rule



This image depicts all four possibilities

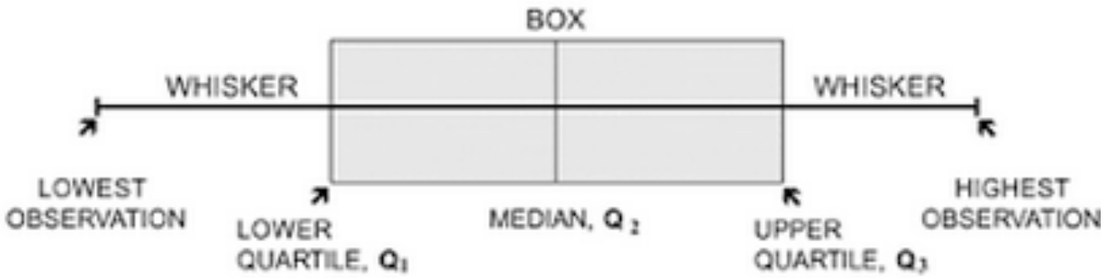
- Note that each partial line (the blue lines) has the same sign on its slope
- If one slope is positive and the other negative, nothing can be done to linearize

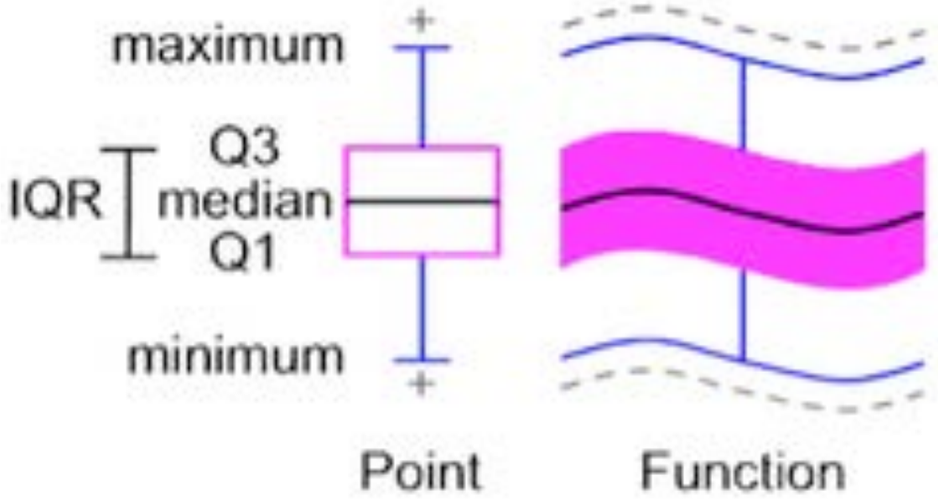
Supervised & Unsupervised ML

- **Supervised** Learning: generalizing from labeled data
  - Classification
  - Prediction
  - Estimation
  - Regression
- **Unsupervised** Learning: generalizing from unlabeled data
  - Clustering
  - Association-rule learning
  - Summarization

Functional Boxplots

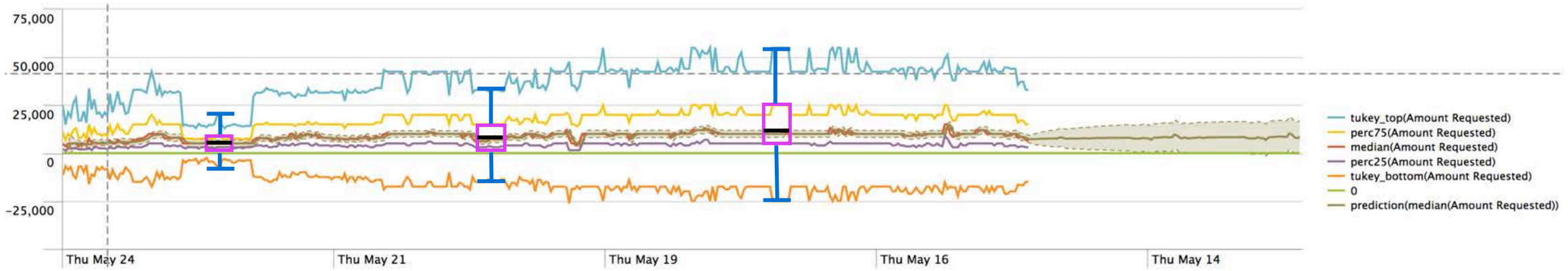
- Functional boxplots show how trends change over time
- Like box and whisker plots
  - x-axis: category of comparison
  - center of boxes = **`median`**
  - bottom = **`perc25`**
  - top = **`perc75`**
  - extreme outliers indicated





Functional Boxplots (cont.)

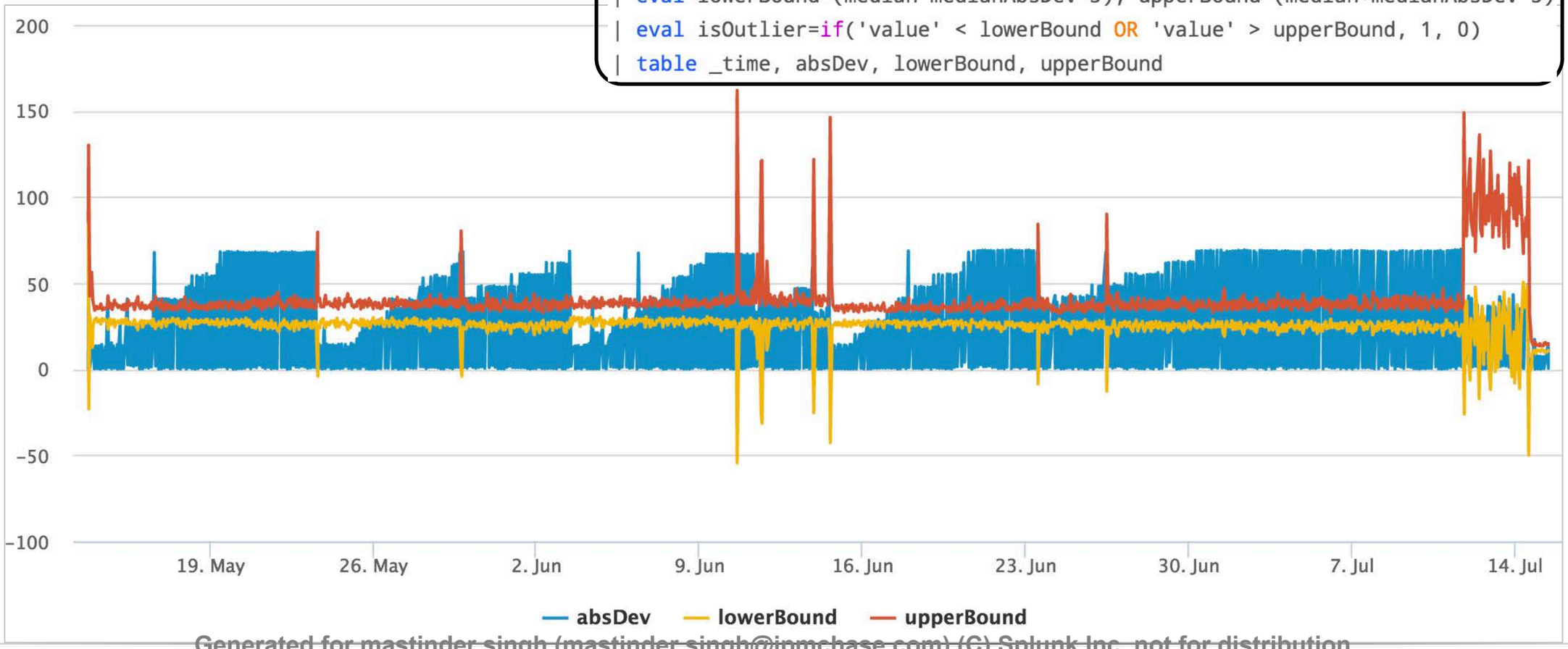
- Box-and-whisker spread along an axis
- Outlier envelopes around the trend
  - Most data is in the box (between **`perc25 & perc75`**)
  - Nearly all the data is between the whiskers



https://www.lendingclub.com/info/download-data.action

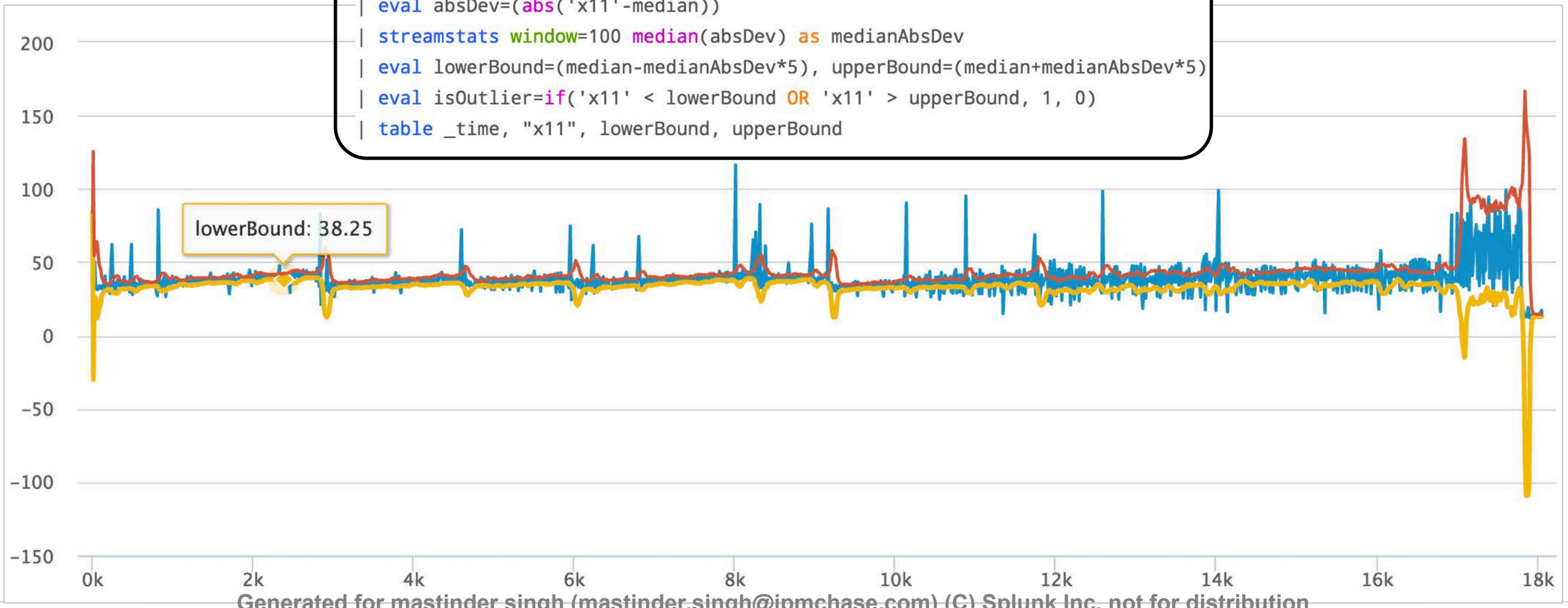
Data with Periodic Elements

```
| inputlookup cpu.csv
| streamstats window=20 median("value") as median
| eval absDev=(abs('value'-median))
| streamstats window=20 median(absDev) as medianAbsDev
| eval lowerBound=(median-medianAbsDev*5), upperBound=(median+medianAbsDev*5)
| eval isOutlier=if('value' < lowerBound OR 'value' > upperBound, 1, 0)
| table _time, absDev, lowerBound, upperBound
```



Same Periodic Data with **x11**

```
| inputlookup cpu.csv
| eval _time = strptime(timestamp, "%F %T")
| x11 value as x11
| streamstats window=100 median("x11") as median
| eval absDev=(abs('x11'-median))
| streamstats window=100 median(absDev) as medianAbsDev
| eval lowerBound=(median-medianAbsDev*5), upperBound=(median+medianAbsDev*5)
| eval isOutlier=if('x11' < lowerBound OR 'x11' > upperBound, 1, 0)
| table _time, "x11", lowerBound, upperBound
```



ARIMA

- **A**uto **R**egressive **I**ntegrated **M**oving **A**verage describe autocorrelations (correlation of a time series with its own past values)
- `ARIMA` requires order (needs three values):
  - Number of autoregressive (AR) parameters: express the dependency of the current value of time series to its previous ones
  - Number of differencing operations (D): model the effect of previous forecast errors (also called random shocks or white noise) on the current value.
  - Number of moving average (MA) parameters: make non stationary data stationary (distribution does not change by time)

ARIMA Recommendations

- You need to know the parameters in advance
- Send the time series through timechart before sending it into ARIMA (unless _time is not to be specified)
- If there are missing samples in the data, expand the span in timechart or use streamstats
- ARIMA supports just one time series at a time
- ARIMA models cannot be saved

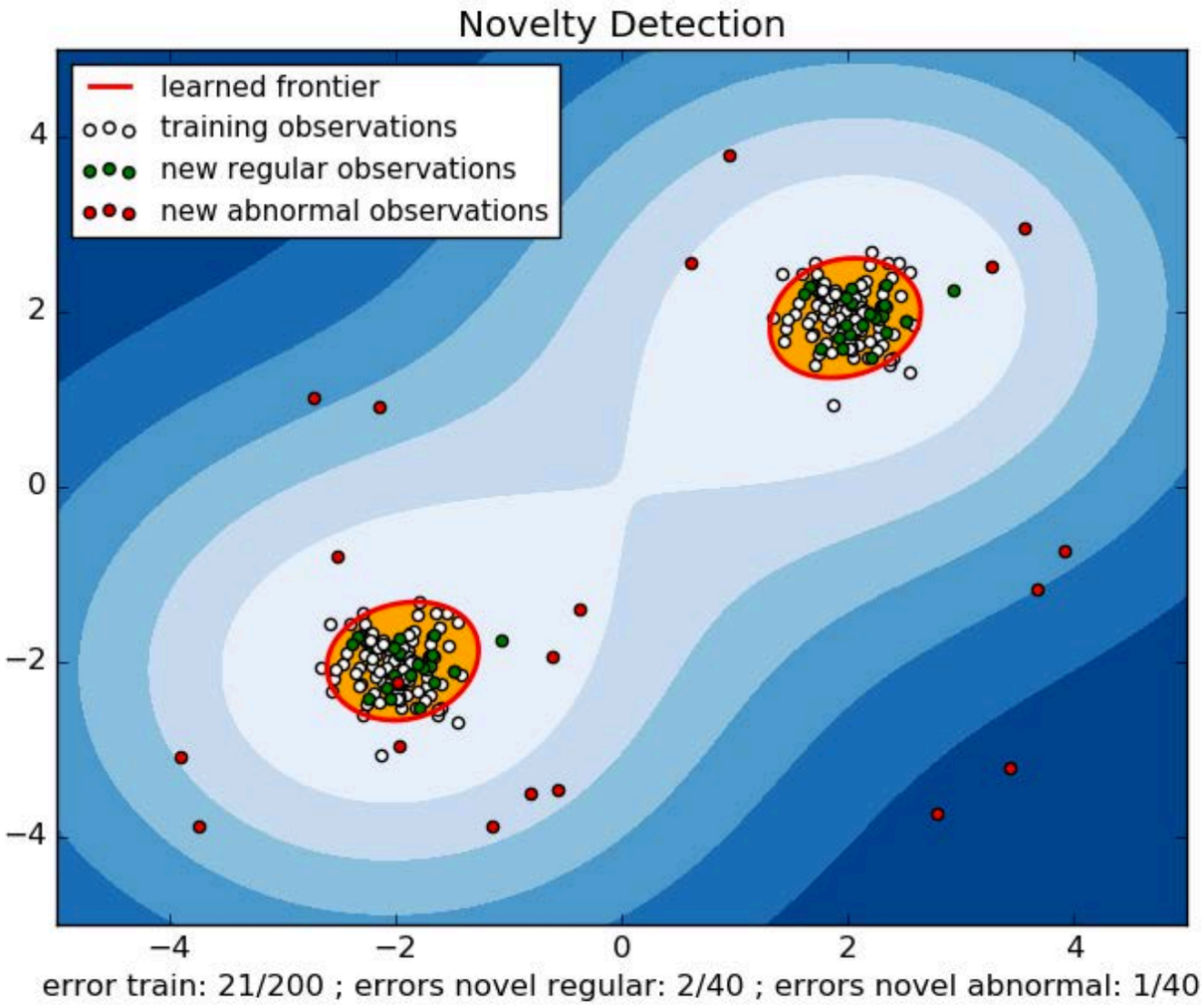
ARIMA Parameters

- forecast_k=<int>: how many points into the future should be forecasted
  - If _time is specified during fitting along with the <field_to_forecast>, ARIMA also generates the timestamps for forecasted values
  - default forecast_k is zero
- conf_interval=<1..99>: confidence interval around forecasted values as a %
  - default is 95%

```
fit ARIMA [_time] <field_to_forecast>
order=<N>-<N>-<N> [forecast_k=<N>] [conf_interval=<N>]
```

`OneClassSVM` Overview

- For detecting anomalies/outliers
- Features (fields) are expected to contain numerical values
- Unsupervised
- Learns a decision function to classify new data as similar or different to the training set



scikit-learn's OneClassSVM

OneClassSVM Syntax

```
fit OneClassSVM <fields> [into <model name>]
[kernel=<str>] [nu=<float>] [coef0=<float>]
[gamma=<float>] [tol=<float>] [degree=<number>]
[shrinking=<t|f>]
```

- Kernel specifies the kernel type `("linear", "rbf", "poly", "sigmoid")` for use in the algorithm, where the default value of `kernel="rbf”`
- `nu` can specify the upper bound on the fraction of training error as well as the lower bound of the fraction of support vectors
  - Default value is 0.5

`OneClassSVM` Options

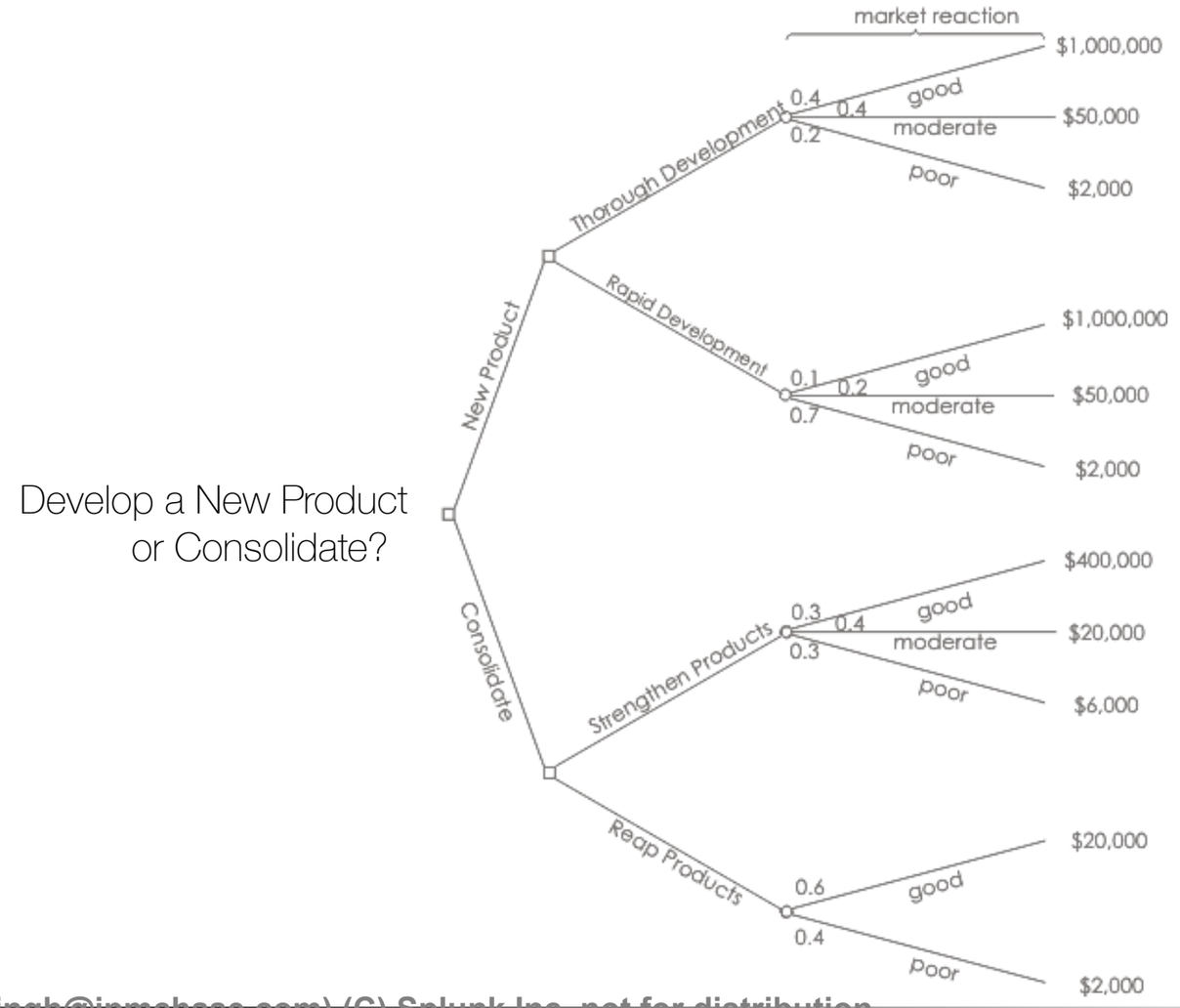
- `degree` is ignored by all kernels except the polynomial kernel
  - Default value is 3
- `gamma` is the kernel co-efficient that specifies how much influence a single data instance has
  - Default value is 1/numberOfFeatures
- `coef0` is the independent term in the kernel function which is only significant if you have polynomial or sigmoid function
- `tol` is the tolerance for stopping criteria
- `shrinkingparameter` sets whether to use the shrinking heuristic

Working with OneClassSVM Models

- You can save OneClassSVM models using the into keyword and apply the saved model later to new data using the apply command
  - After running **`fit`** or **`apply`**, a new field named **`"isNormal"`** is generated that defines whether a particular record (row) is normal **`("isNormal"=1)`** or anomalous **`("isNormal"=-1)`**
- The summary command is not available with **`OneClassSVM`**

Decision Trees

- Consists of nodes (splits), edges (branches), and (terminal) leaves
- Finds the attribute that returns the highest information gain (the most homogeneous branches)
  - By calculating the drop in entropy after a dataset is split based on attributes



SGD Details

- Linear model fitted by minimizing a regularized empirical loss
  - Many algorithms are derived by making a cost function or an optimization objective
  - Then gradient descent is used to minimize that cost function
  - SGD is more efficient / feasible for large datasets than GD
- The gradient of the loss is estimated one sample at a time and the model is updated with a decreasing strength schedule (learning rate)
- The regularizer is a penalty added to the loss function that shrinks model parameters towards the zero vector

SGD Key Parameters (Optional)

- **partial_fit**=< true | **false** >: incrementally updated or not
- **loss**=< **hinge** | log | modified_huber | squared_hinge | perceptron >: loss function
- **n_iter**=<int>: # of passes over the training data (aka epochs)
  - **default 5**
- **penalty**=< **l2** | l1 | elasticnet >: which regularization term to use
- **l1_ratio**=<float>: The Elastic Net mixing parameter,
  - **default 0.15**
- **alpha**=<float>:constant, multiplies regularization term **default 0.000**

Bootstrap Aggregating (Bagging)

- Bagging is a model averaging meta algorithm
- Generates multiple training sets from the same data (resampling) with replacement (default in random forest)
  - Reduces variance
  - Helps prevent overfitting
  - Often used with
    - Decision trees