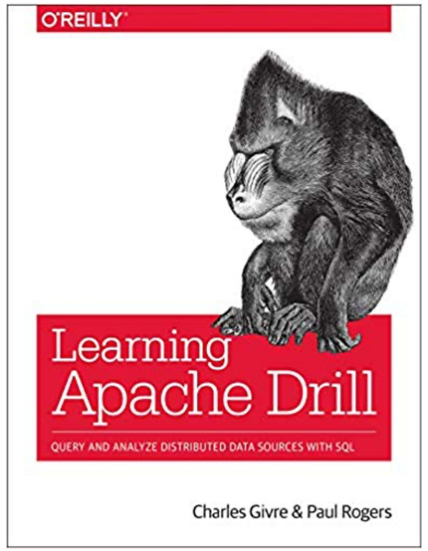
Data Workshop: Applied Supervised Learning for Cyber Security

Presented by Charles S. Givre CISSP
charles.givre@gtkcyber.com

Charles Givre, CISSP

- Lead Data Scientist at JP Morgan Chase
- PMC Chair for Apache Drill
- Senior Lead Data Scientist @ Booz Allen
- 5 Years @ CIA
- Undergraduate in Comp.Sci & Music



ARIZONA

TRUTH · EVEN UNTO ITS INNERMOST PARTS · Brandeis אמת

JPMorgan Chase & Co.

Deutsche Bank

Booz | Allen | Hamilton

100 YEARS

Expectations

- Please participate and **ask questions.**
- Please follow along and **TRY OUT** the examples yourself during the class
- All the answers are in the slide decks or GitHub repository, but please try to complete the exercises **without looking at the answers**.
- Join the conversation in slack!
- Have fun!

Shameless Plug...

http://bit.ly/cyberds-md

What is Machine Learning (ML) Artificial Intelligence (AI)

“Machine Learning is the science of getting computers to act without being explicitly programmed.”

– https://www.coursera.org/course/ml

“A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.”

–Tom Mitchell, Carnegie Mellon University

“Machine learning explores the construction and study of algorithms that can learn from and **make predictions on data.** Such algorithms operate by building a model from example inputs in order to make data-driven predictions or decisions, **rather than following strictly static program instructions.**”

–https://en.wikipedia.org/wiki/Machine_learning


SPAM®
CHOPPED PORK AND HAM
LICENCED BY:
Hormel
Foods
Serving Suggestion


SPAM®
CHOPPED PORK AND HAM
LICENCED BY:
Hormel Foods
Serving Suggestion


SPAM
TERIYAKI
Glorious SPAM!


SPAM
LITE
Glorious SPAM!


SPAM
Oven Roasted TURKEY
Hormel Foods

- Blacklists
- Simple keyword matching
- Naive Bayesian Classifiers
- Deep Learning

Artificial Intelligence

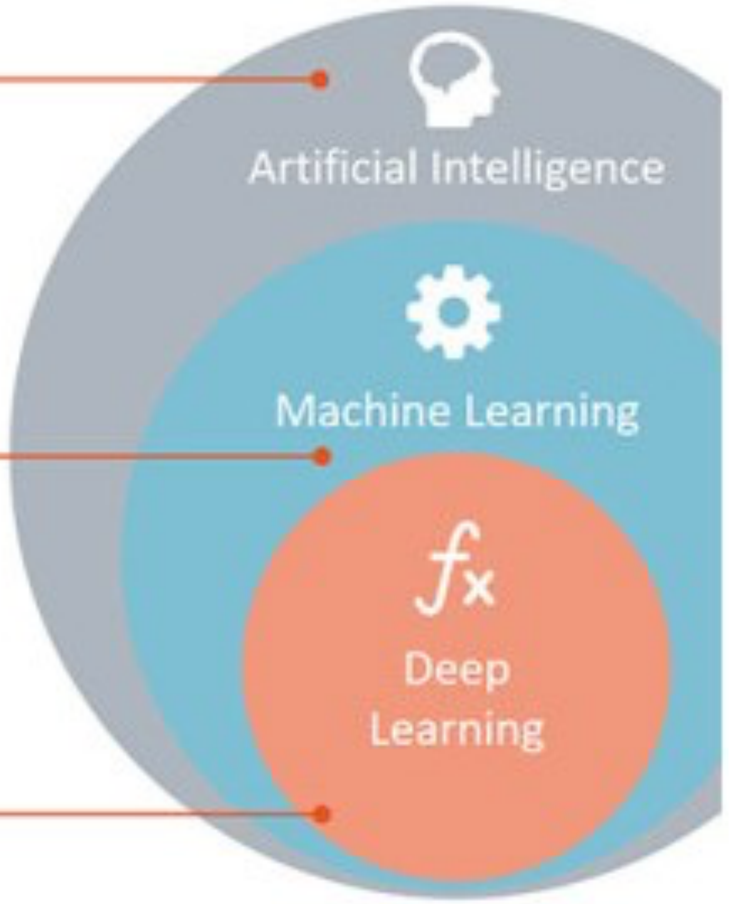
Any technique which enables computers to mimic human behavior.

Machine Learning

Subset of AI techniques which use statistical methods to enable machines to improve with experiences.

Deep Learning

Subset of ML which make the computation of multi-layer neural networks feasible.



https://www.kdnuggets.com/2017/07/rapidminer-ai-machine-learning-deep-learning.html

@katherinebailey Because marketing? Every time someone calls simple linear regression “AI” Gauss turns over in his grave.

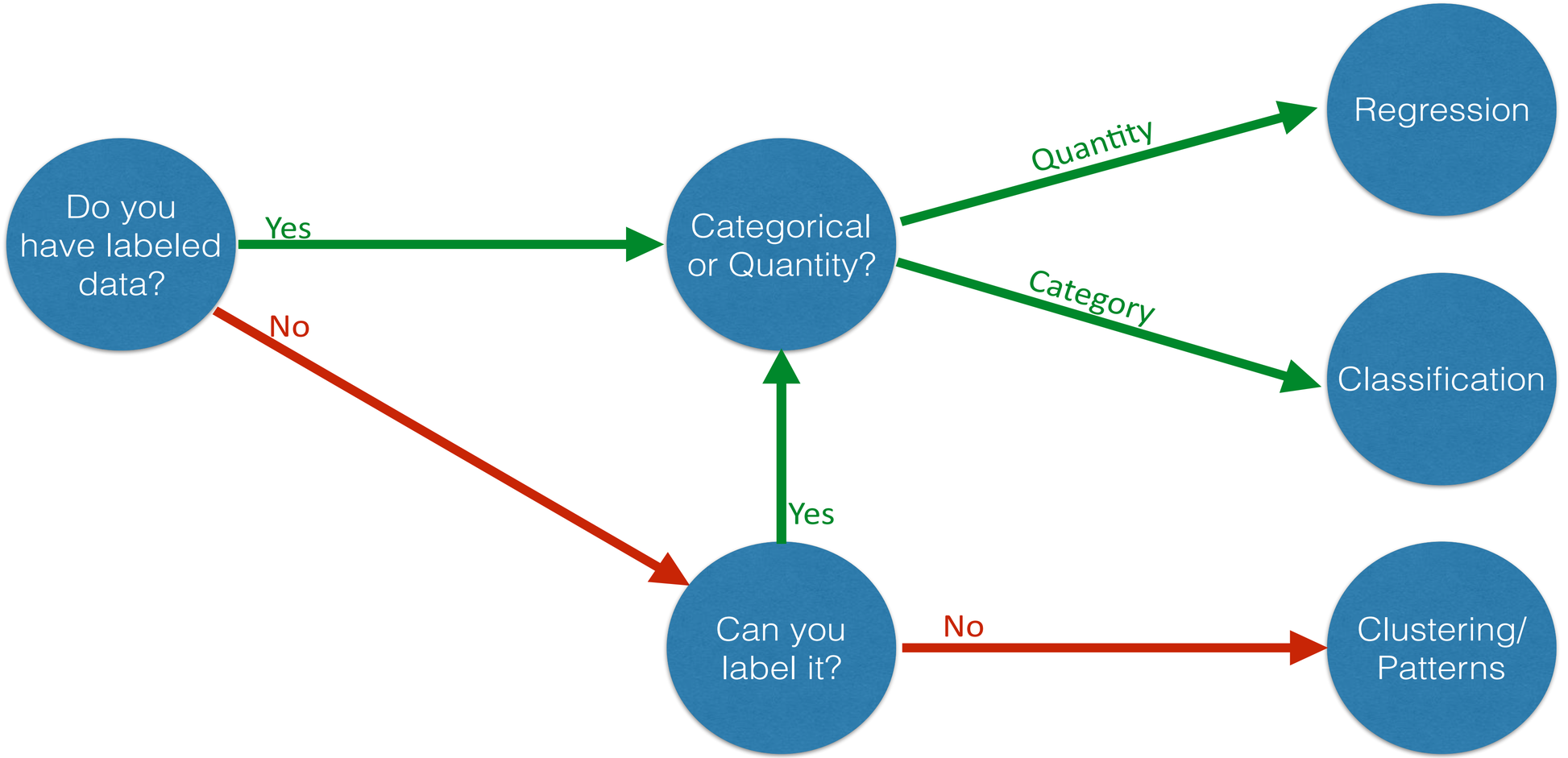
Machine Learning Problems

- **Supervised Learning:** Supervised Learning is a class of Machine Learning in which a model is "trained" using a set of pre-existing labeled data.
- **Unsupervised Learning:** A class of Machine Learning algorithms in which a model is built without the use of labeled data.

Machine Learning Problem Types

- **Classification**: Assigning or predicting a observation's membership in discrete class
- **Regression:** Predicting a continuous value based on the observations' features
- **Clustering:** Identifying groupings within a dataset
- **Dimensionality Reduction:** Reducing the number of variables in a feature set

What Problem am I solving?

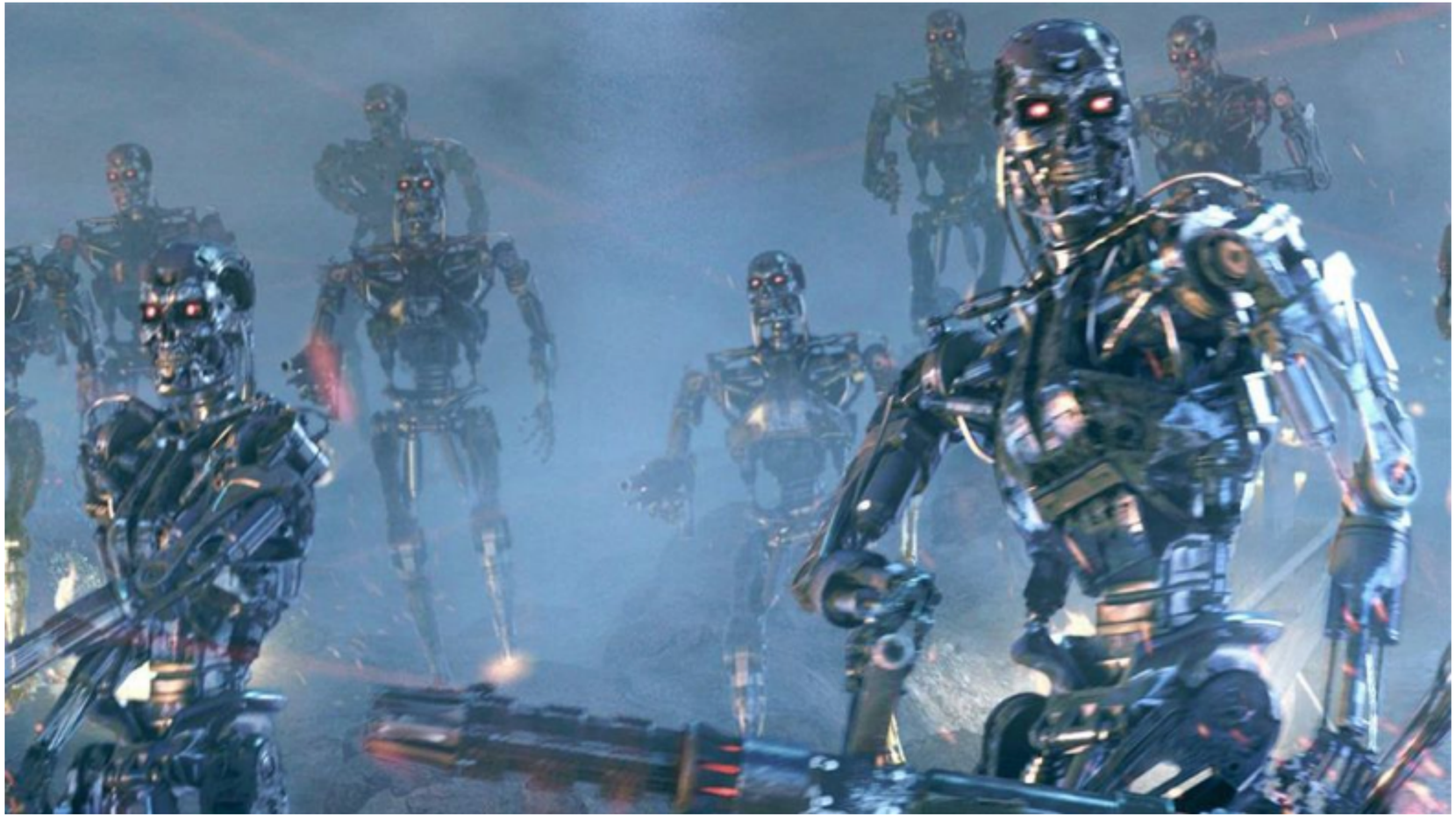


Prediction, Inference

Prediction

Nothing proven

Recommenders, Similar, anomalous



What it is not

Applications to Security

Regression Example

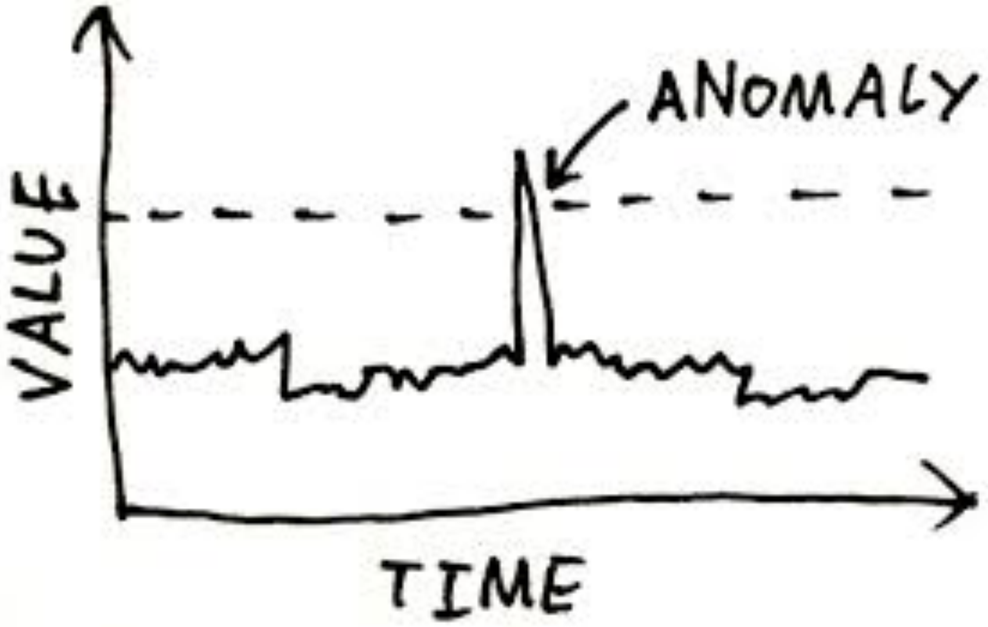
Server Capacity Prediction: Regression analysis can be used to predict a server's capacity (or CPU usage) based on the server's historical performance.

https://www.researchgate.net/publication/256645877_LiRCUP_Linear_Regression_based_CPU_Usage_Prediction_Algorithm_for_Live_Migration_of_Virtual_Machines_in_Data_Centers

https://jgreenemi.com/predicting-capacity-with-linear-regression-ml/

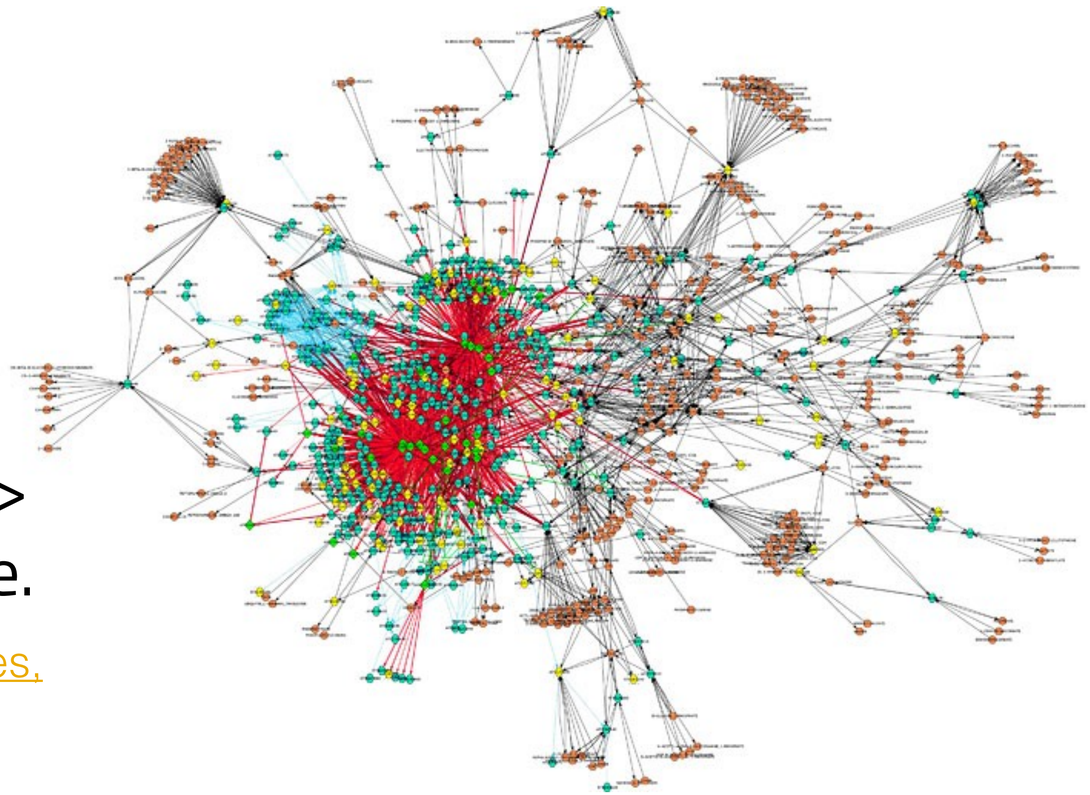
Clustering Example

Anomaly Detection: Clustering techniques can be used to detect anomalous traffic or loads or anything really.



Network-Based Intrusion Detection

- Derive Features from Network Traffic Captures “pcap” at packet level or NetFlow level (tools: tshark, tcpdump, bro…)
- Example Features based on header information: 2s-windowing of connections > duration, protocol, src and dat bytes, service.
- Get data sets: http://www.netresec.com/?page=PcapFiles, https://maccdc.org/, http://www.westpoint.edu/crc/SitePages/DataSets.aspx http://www.unb.ca/cic/research/datasets/,



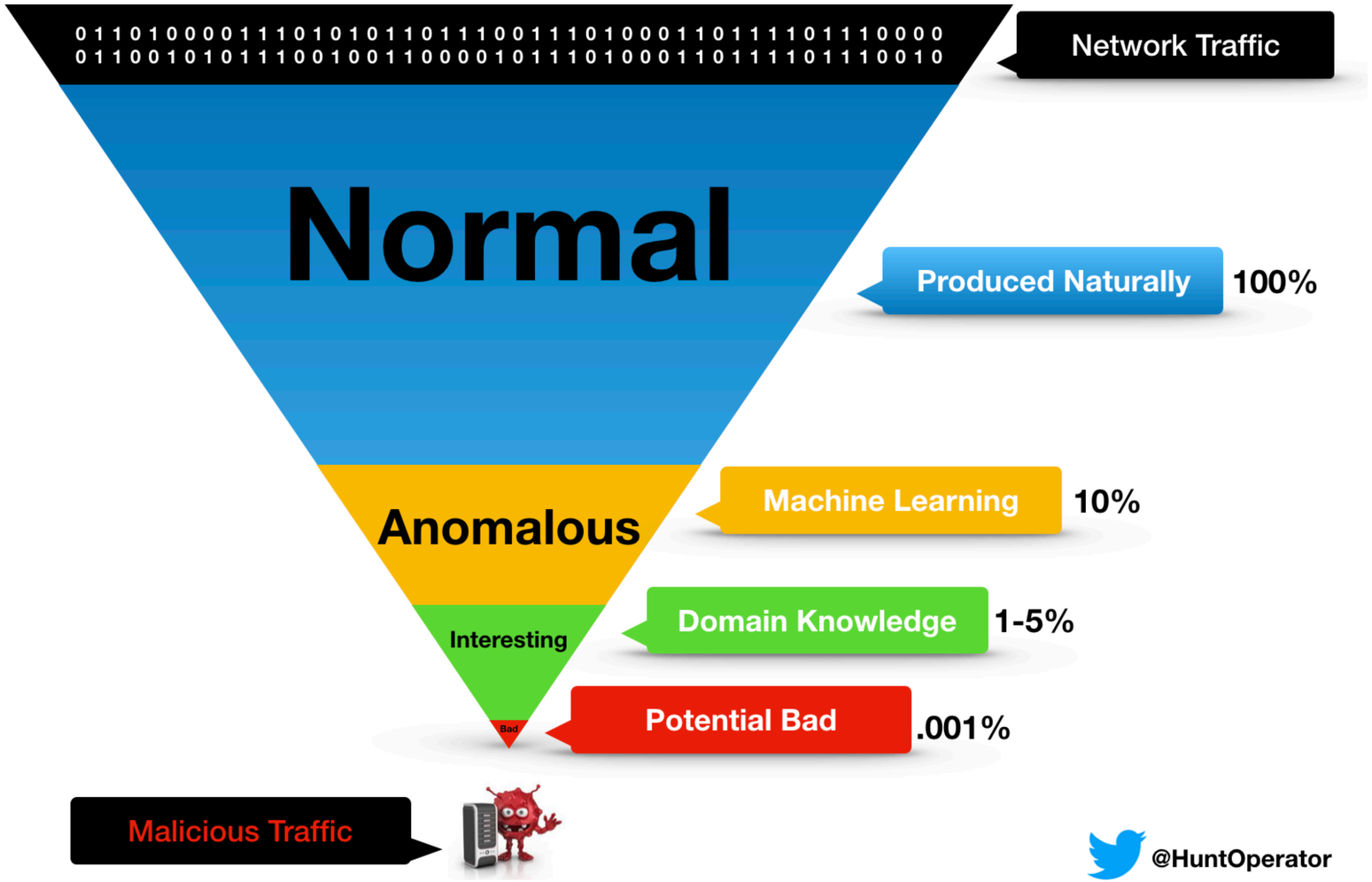
Malware Detection/Classification

- Derive Features from Binary Content and metadata manifest (function calls, string obtained from IDA Disassembler)
- Example Features: opcode count (n-grams), segment count, asm pixel intensity, n-gramming of bytes, function name.
- Featureless Deep Learning with word2vec embedding
- Get open source malware samples: Vx Heaven, Virus Share, Maltrieve, Open Malware

Security Applications of Machine Learning

- Domain Generation Algorithm (DGA) Detection (Classification)
- Malicious URL Detection (Classification)
- Network Traffic: Beaconing Detection (Classification/Clustering)
- Detection of new classes of malware (Classification/Clustering)
- General Network Traffic Anomaly Detection (Classification/Clustering)
- Log Analysis - Anomaly Detection (Classification/Clustering)
- Phishing Detection (Classification)
- Identifying SQL Injection (Classification)
- Identifying XSS cross-site scripting (Classification)
- DOS/DDOS Detection (Classification)
- Authentication (Classification)

Data Science Hunting Funnel

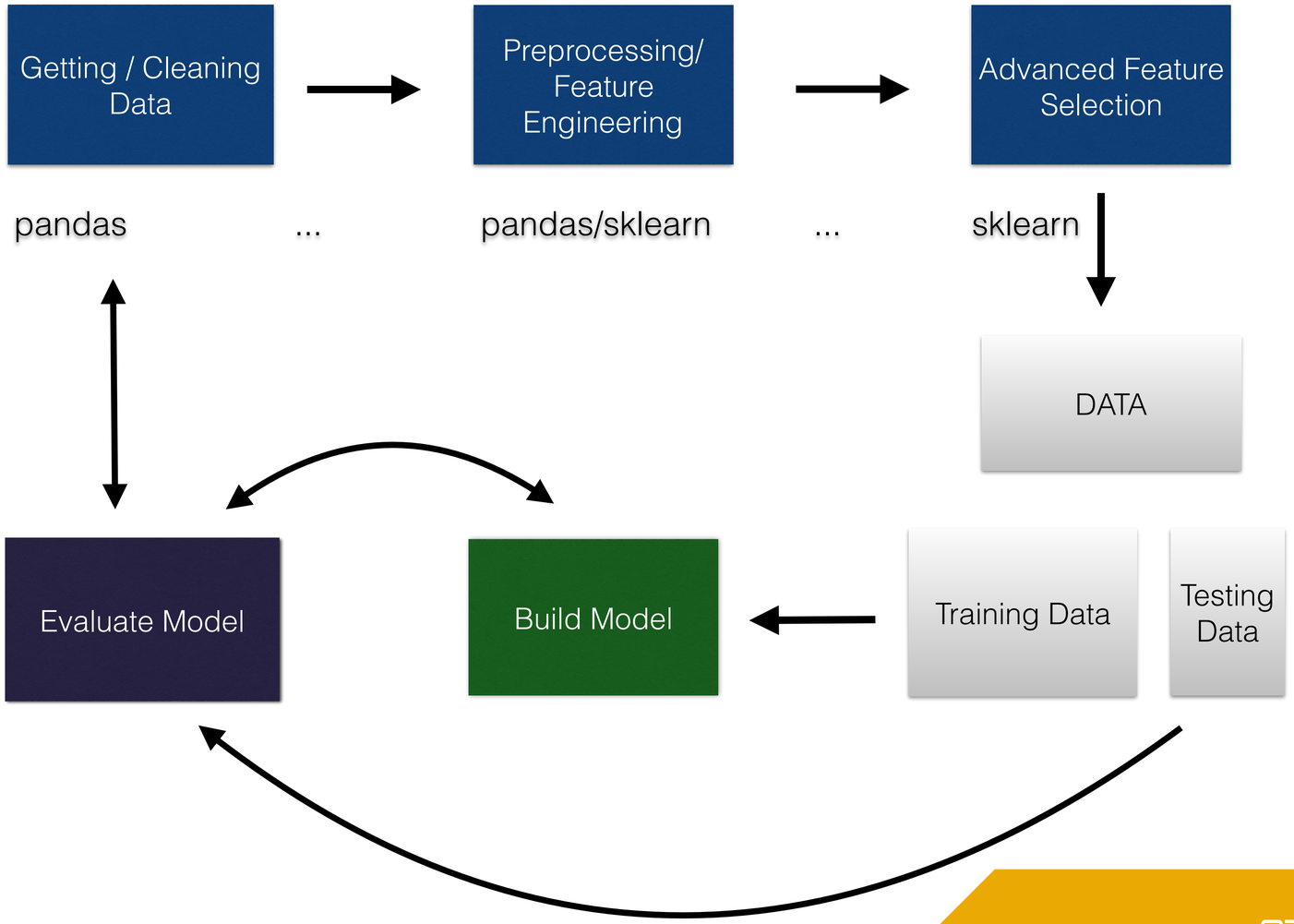


@HuntOperator

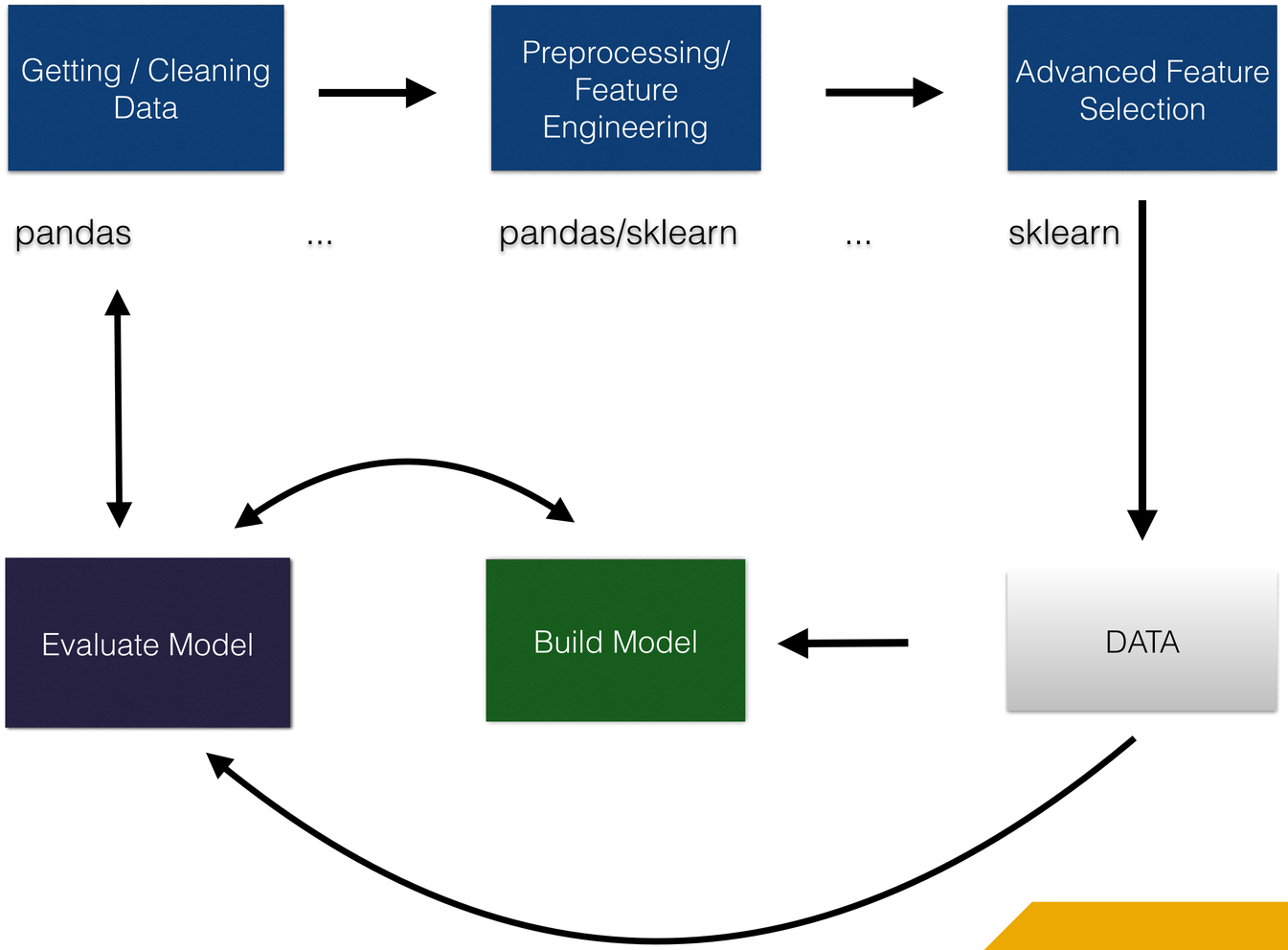
http://www.austintaylor.io/network/traffic/threat/data/science/hunting/funnel/machine/learning/domain/expertise/2017/07/11/data-science-hunting-funnel/

The Machine Learning Process

Supervised Machine Learning Process



Unsupervised Machine Learning Process



First, define your analytic question.

What are you trying to do?

How do you define success?
What are you measuring?

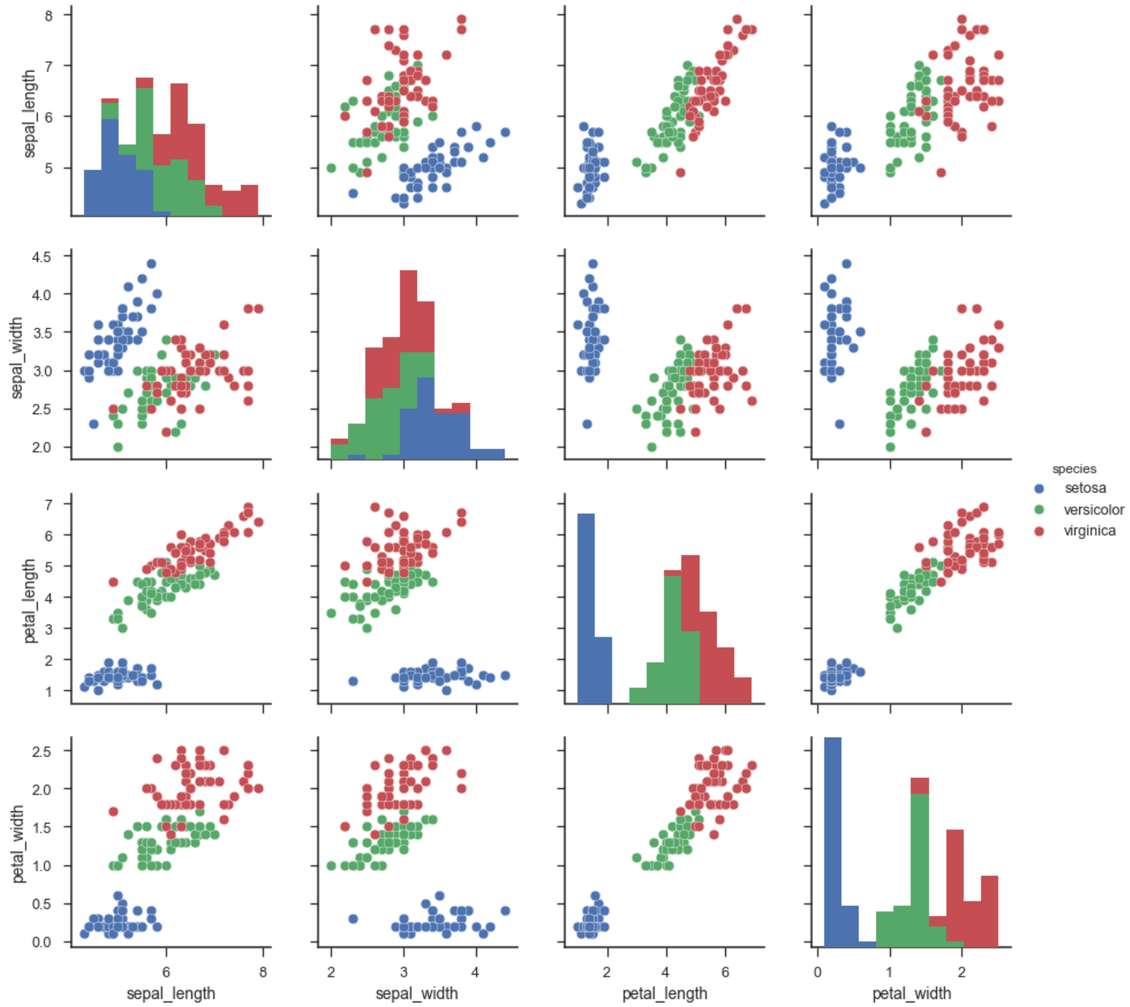
Choose data sources

- What is available?
- Is it enough?
- Is the data reliable/clean/consistent?
- What other data could you use?

Other Considerations

- Policies
- Legal contraints
- Biases in Data
- Latency
- Data size

Gather and Explore Your Data



Is the data good enough?

What are the rules governing its use?

Do I have enough?

Do problems or biases exist in the data that could cause problems?

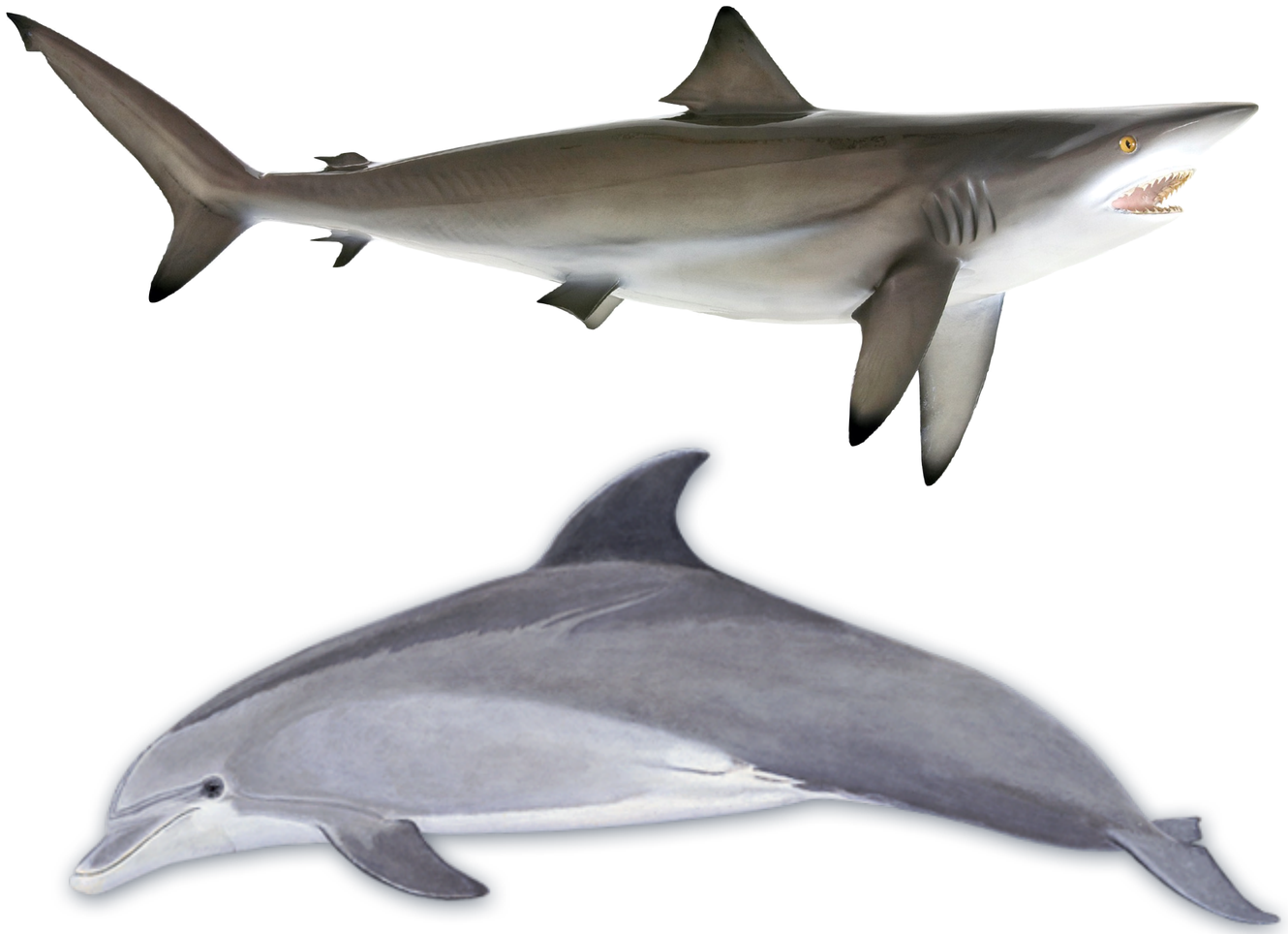
Feature Engineering

- Define what you are trying to measure. These will become the observations or rows of your final dataset
- Define how you will mathematically represent your data. This will be come the features or columns of your final dataset.

Feature Engineering

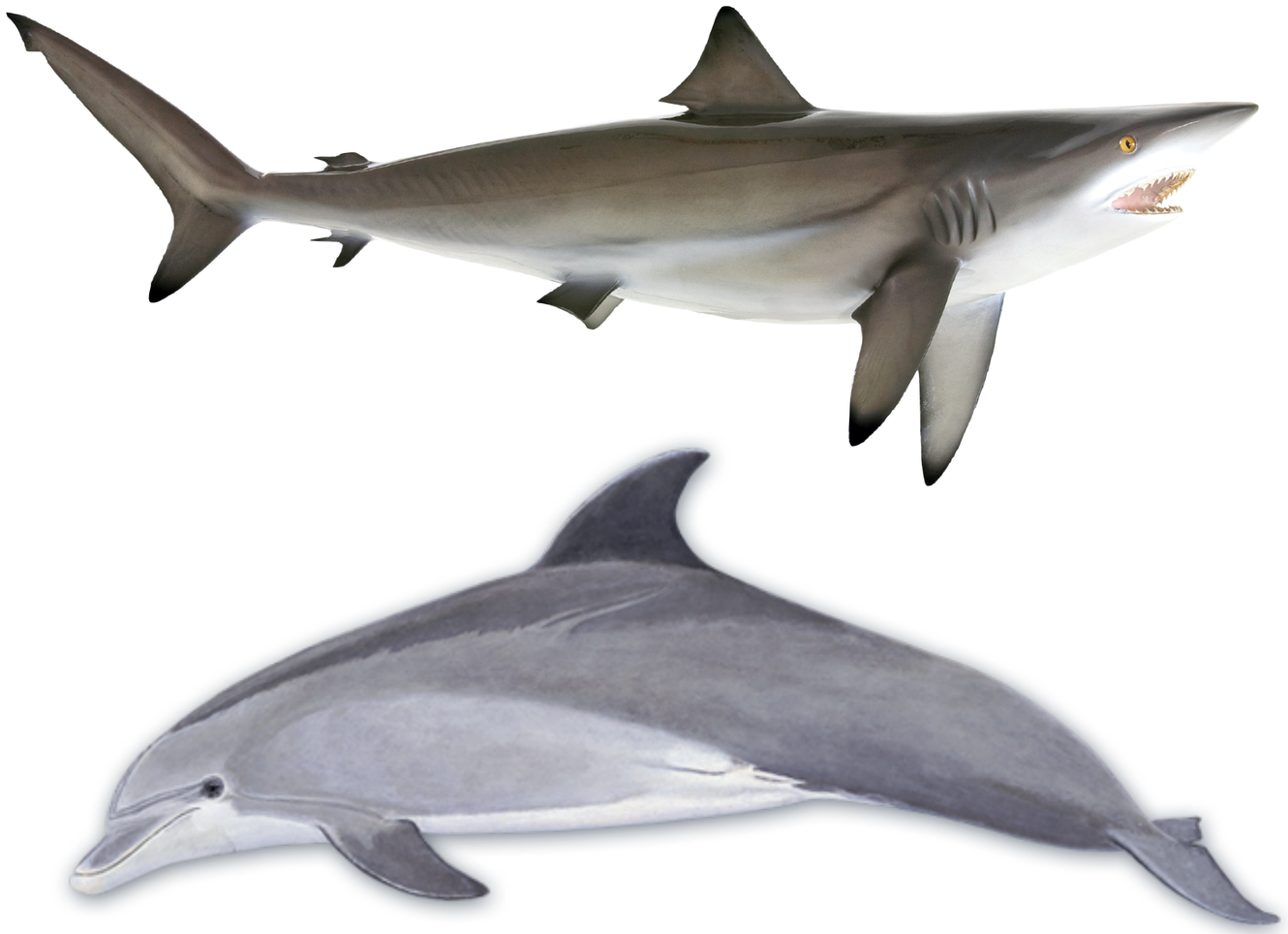
Feature	Value
Color	Gray
Fins	7
Predator	TRUE

Feature Engineering



Feature	Value
Color	Gray
Fins	7
Predator	TRUE

Feature Engineering



Feature	Value
Color	Gray
Fins	7
Predator	TRUE
Mammal	**TRUE**

Build and Tune your Model

- Believe it or not, this is the easy part.
- Most of this is **done using libraries** like scikit-learn, mllib, tensorflow, caret or keras, and **many steps can be automated**.
- You can even do it in Splunk or Elasticsearch.

Evaluate Performance

- Use various scoring methods, or write your own to determine model performance.
- Go back to step 1 and repeat! (Do not pass go, do not collect $200)

Discussion

- Consider that you are building a system to identify fraudulent credit card transactions. What are some features that you would want to capture?
- What are the challenges you could foresee in this problem?

The Python Data Science Ecosystem

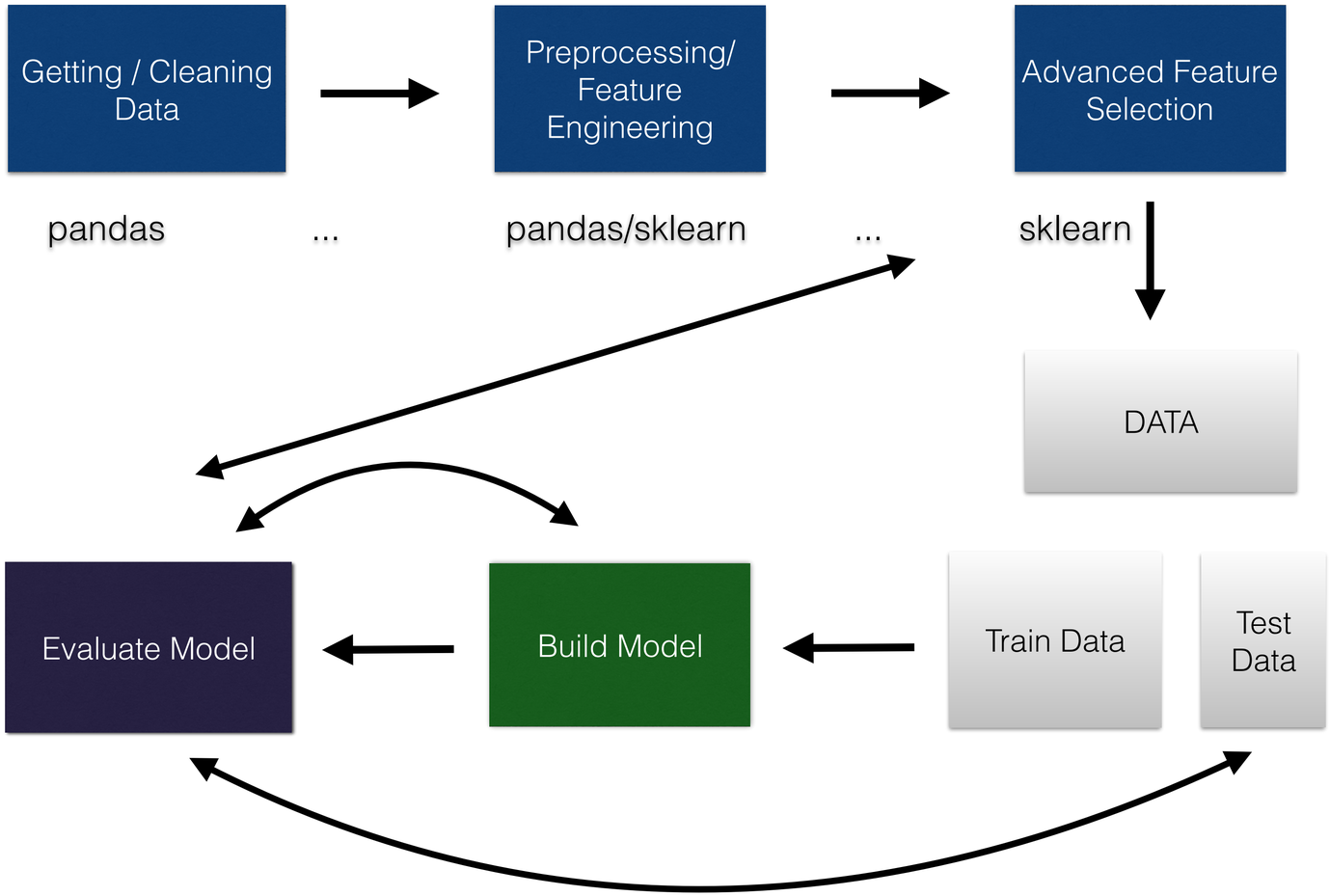
Machine Learning Ecosystem

- **Data Gathering:** Pandas, Drill, BeautifulSoup, PyDBAPI, PyDAL, Boto3
- **Feature Extraction:** Pandas, NumPy, Featuretools
- **Machine Learning**
  - **"Regular" ML:** Scikit-learn (sklearn), h2o, mllib (PySpark)
  - **Deep Learning:** Tensorflow, Keras, Theano, Caffe, PyTorch
- Visualization: Matplotlib, Seaborn, Yellowbrick, LIME, ggplot, plot.ly,

Machine Learning Part 1: Feature Engineering

From URL Strings to Features

Machine Learning Process



Machine Learning Terms

- **Features:** The mathematical representation of the original data. The features are the columns in your data set. Since the features will be a matrix, the are often written as X.
- **Observations:** The rows of your feature set.
- **Target:** The variable that you are trying to predict. Often represented as y.

Features

http://www.google.com

Features

http://www.google.com

domain_length	vowel_count	digit_count
6	3	0

Representation of URL Knowledge

- Come up with a **representation/set of knowledge** that has **enough complexity** to accurately describe the problem for the computer
- Knowledge here does not mean hard-coded knowledge or formal set of rules
- **The computer rather uses the knowledge we provide to extract patterns and acquire own knowledge**
- We should provide knowledge about reality that has **high variance about the problem** it describes (e.g. a feature that is high when it rains and low when it’s sunshine)

Malicious URL Detection Features (Literature)

1. **BlackList Features**: BlackLists suffer from a high false negative rate, but can still be useful as machine learning feature.
2. **Lexical Features**: Capture the property that malicious URLs tend to "look different" from benign URLs. **Contextual information** such as the length of the URL, number of digits, lengths of different parts, entropy of domain name.
3. **Host-based Features**: Properties of web site host. **"Where"** the site is hosted, **"who" owns it** and **"how" it is managed**. API queries are needed (WHOIS, DNS records). Examples: Date of registration, the geolocations, autonomous system (AS) number, connection speed or time-to-live (TTL).
4. **Content-based Features**: Less commonly used feature as it requires **execution of web-page**. Can be not only be not safe, but also increases the computational cost. Examples: HTML or JavaScript based.

Sahoo et al. 2017: Malicious URL Detection using Machine Learning: A Survey

Data Set (Features and Target)

	url	isMalicious	domain	created
56675	jeita.biz/w/google/drive/document.html?ssl=yes	1	jeita.biz	2012-04-11 17:08:19
73229	sosnovskoe.info/layouts/plugins/mailbox	1	sosnovskoe.info	2011-09-19 09:53:07
60112	teothemes.com/html/mp3pl/blue-preview.jpg	1	teothemes.com	2011-09-08 21:43:00
66946	kfj.cc:162/17852q	1	kfj.cc	2013-08-18 05:52:47
81906	verapdpf.info/db/6d1b281b5c4bbcfe3b99228680c232fa	1	verapdpf.info	2016-08-18 07:09:03

Features or X

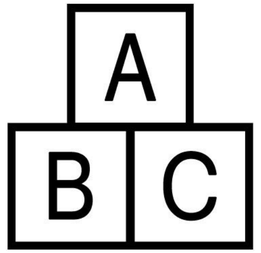
Target or y

	isMalicious	isIP	Length	LengthDomain	DigitsCount	EntropyDomain	FirstDigitIndex	com	org	net	...	w	waset
73820	1	0	27	21	0	3.558519	0	0	1	0	...	0	0
30785	0	0	77	11	14	3.095795	22	1	0	0	...	0	0
60789	1	0	141	11	5	3.459432	103	0	1	0	...	0	0
19495	0	0	59	13	20	3.546594	31	1	0	0	...	0	0
45022	1	0	23	11	7	3.277613	13	0	0	0	...	0	0

Preprocessing - an Art Work!

- Imputing missing values
- Scaling/Normalization
- One-Hot Encoding (Encoding categorical features)
- Embedding (e.g. word2vec)
- Binarizing (e.g. needed for Deep Learning multi-class target vector encoding)
- Encoding strings as int
- Dimensionality Reduction (e.g. PCA)
- Augmentation (e.g. tild/zoom images)
- Feature selection based on classifier
- Variance threshold

Data Types



01

Primary Python libraries: **`pandas, sklearn, scipy`**

http://scikit-learn.org/stable/modules/preprocessing.html

Dealing with Imbalanced Classes

Dealing with Imbalanced Classes

- A lot of real-world security data will have very imbalanced targets.
- Fortunately, there is a library called imbalanced-learn which can assist.
- Imbalanced-learn provides a series of options to resample the data so that you have more balanced classes
- Docs available here: http://imbalanced-learn.org/

Dealing with Imbalanced Classes

```
from imblearn.over_sampling import RandomOverSampler

ros = RandomOverSampler(random_state=0)
X_resampled, y_resampled = ros.fit_resample(X, y)
```

Selecting Features

Should we use all of them?

How do we know which features to use and which to discard?

How do we know which features to use and which to discard?

Select *k* features according to the highest score

```
best_features = SelectKBest(score_func=chi2,k=3).fit_transform(features,target)
```

Select all features above a given threshold in the scoring function

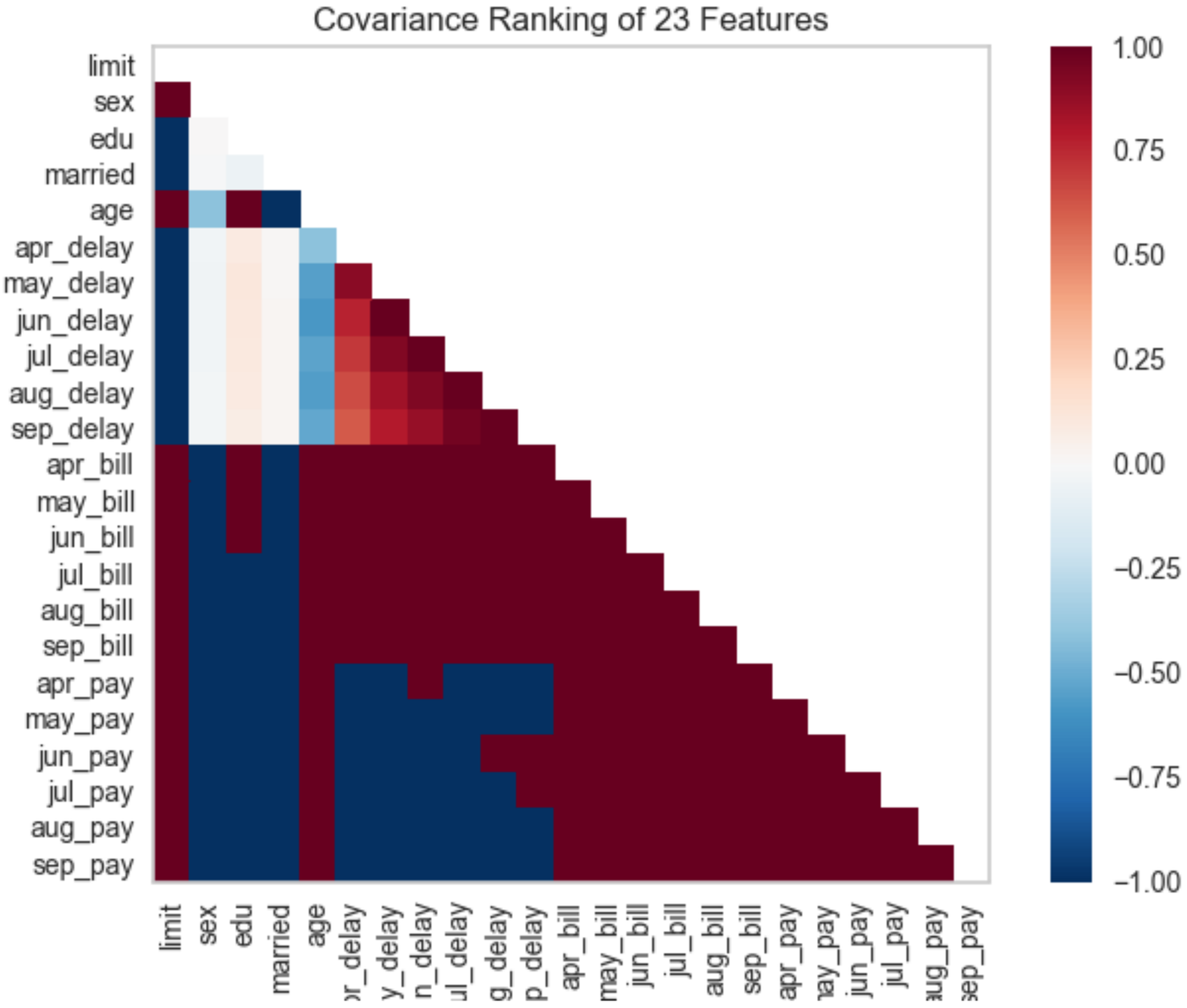
```
best_features =
SelectPercentile(score_func=chi2,percentile=3).fit_transform(features,target)
```

Available Scoring Functions:

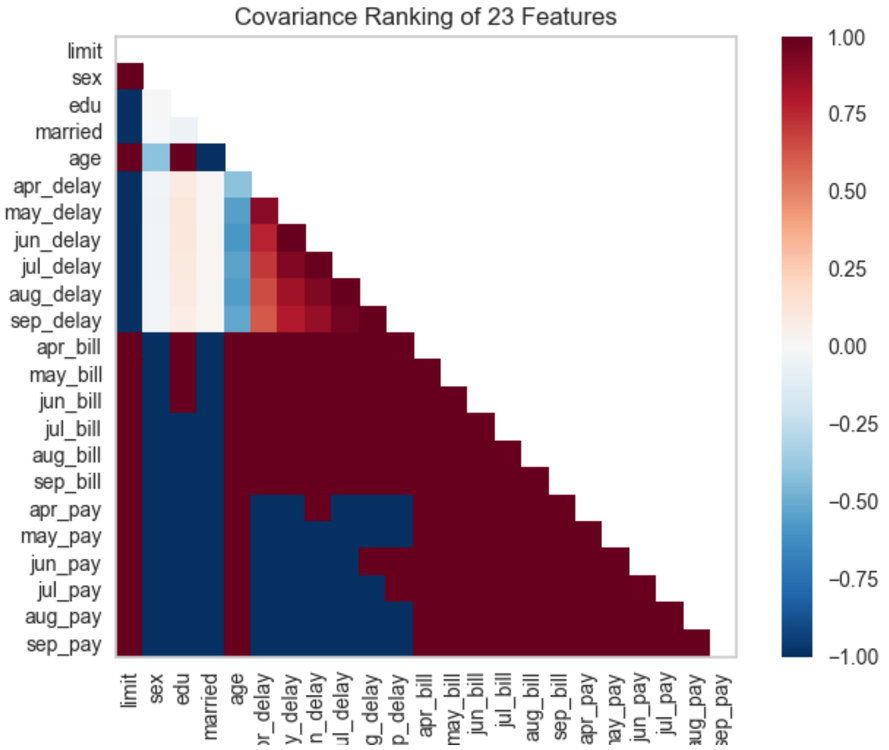
- **For regression:** f_regression, mutual_info_regression
- **For classification:** chi2, f_classif, mutual_info_classif

References:
http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html
http://scikit-learn.org/stable/modules/feature_selection.html#univariate-feature-selection

Introducing Yellowbrick

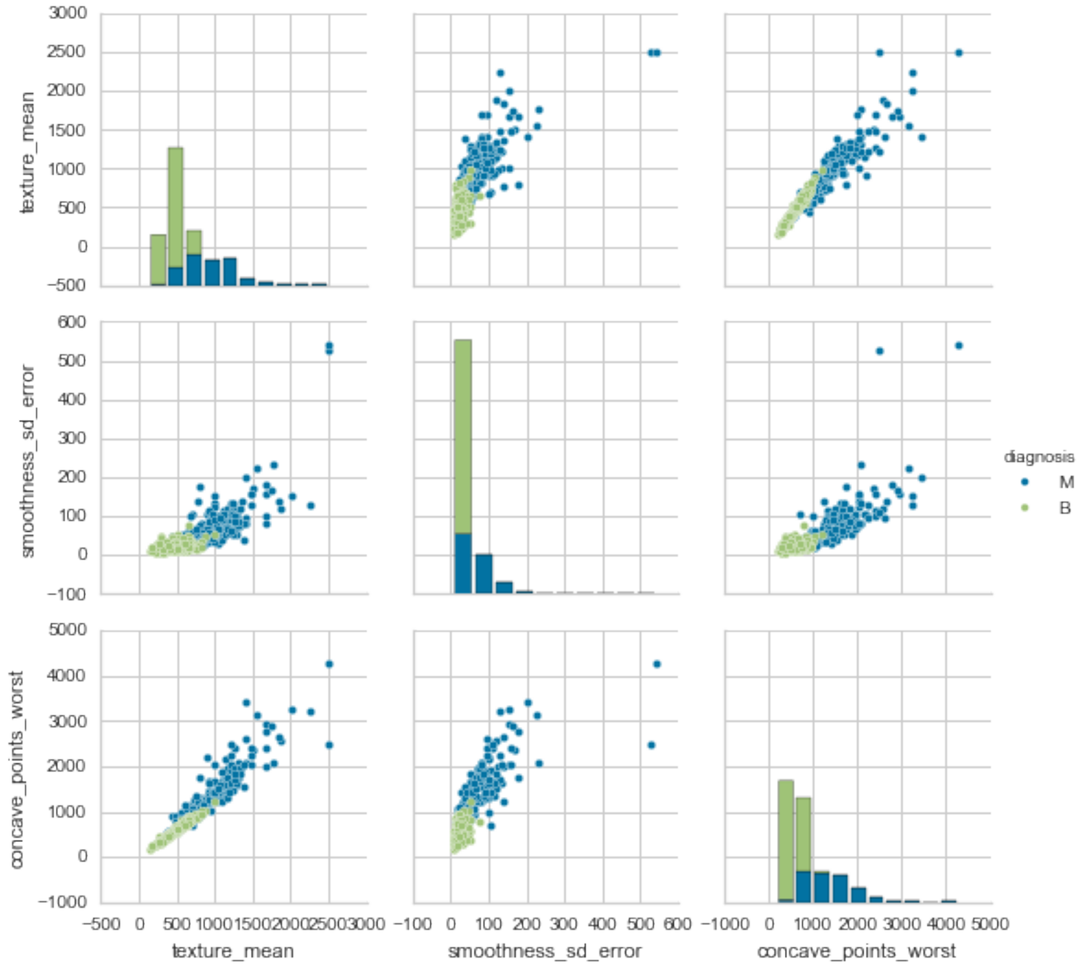


http://www.scikit-yb.org/en/latest/api/features/rankd.html

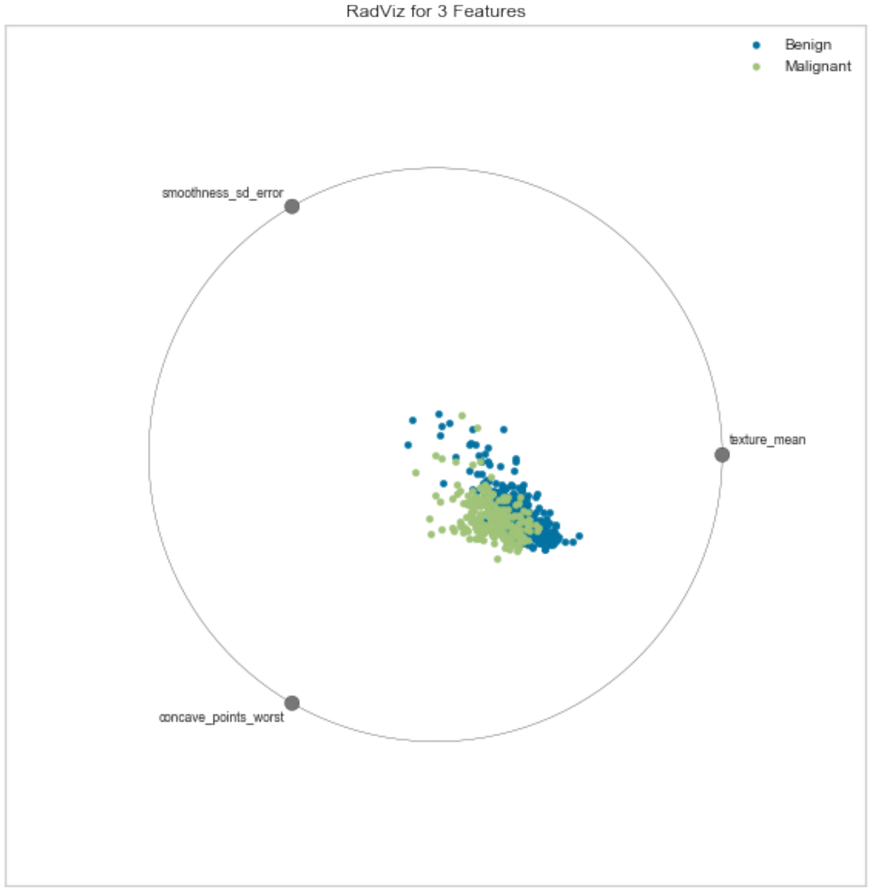



```python
import pandas as pd
import seaborn as sns
from yellowbrick.features.rankd import Rank2D

# Extract the numpy arrays from the data frame
features = df[features]
target = df['diagnosis']

# Instantiate the visualizer with the Covariance ranking algorithm
visualizer = Rank2D(features=features, algorithm='covariance')

visualizer.fit(features, target)
visualizer.transform(features)
visualizer.poof()
```

http://www.scikit-yb.org/en/latest/api/features/rankd.html




```
import seaborn as sns
sns.pairplot(<features>, hue='<target>' )
```

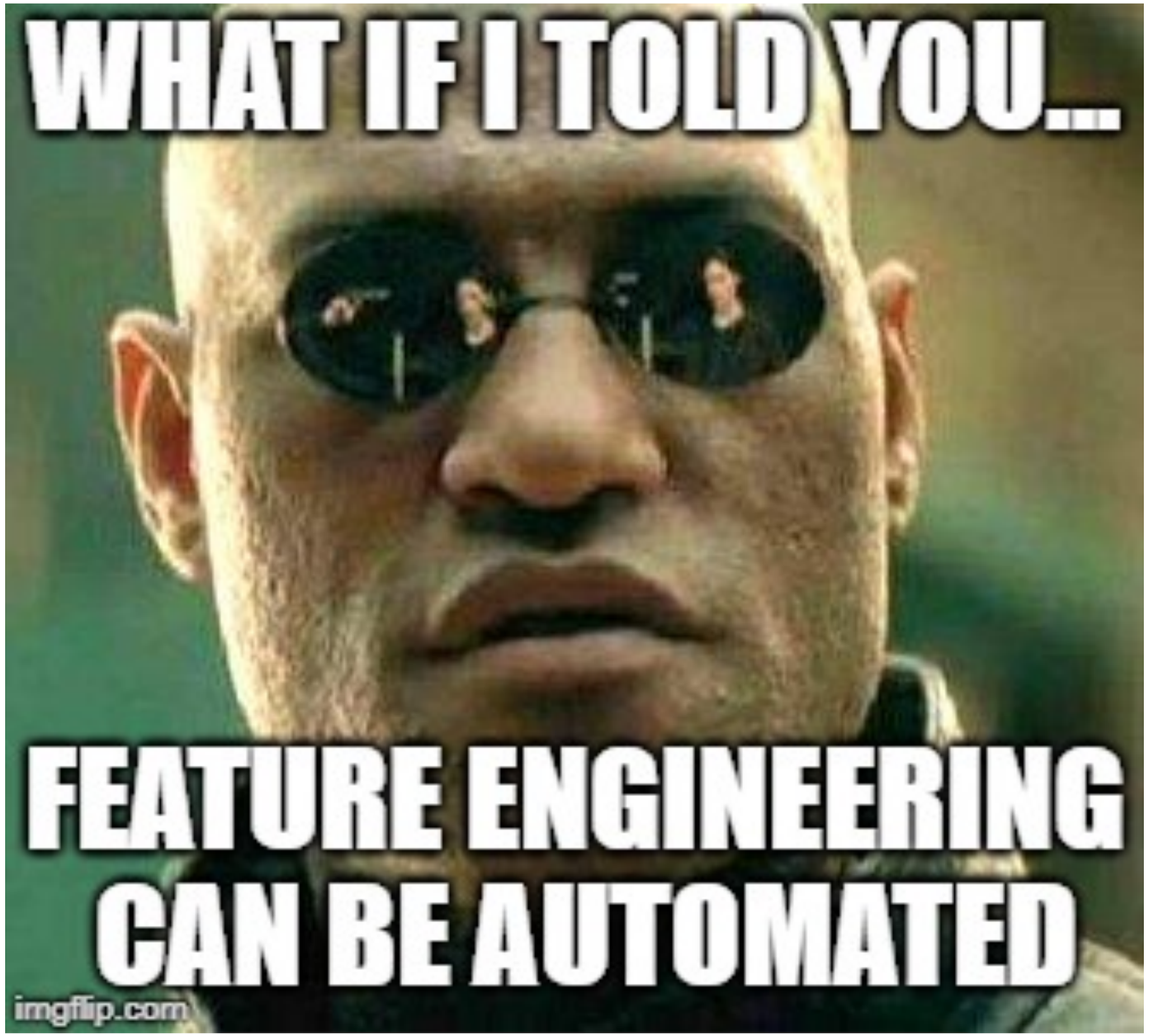



```
from yellowbrick.features.radviz import RadViz
…
visualizer = RadViz(classes=<target classes>, features = <features>)
visualizer.fit(features, target)
visualizer.transform(features)
visualizer.poof()
```

In Class Exercise

Please take 45 minutes and complete

Worksheet 5.1 - Feature Engineering


WHAT IF I TOLD YOU...
FEATURE ENGINEERING
CAN BE AUTOMATED
imgflip.com

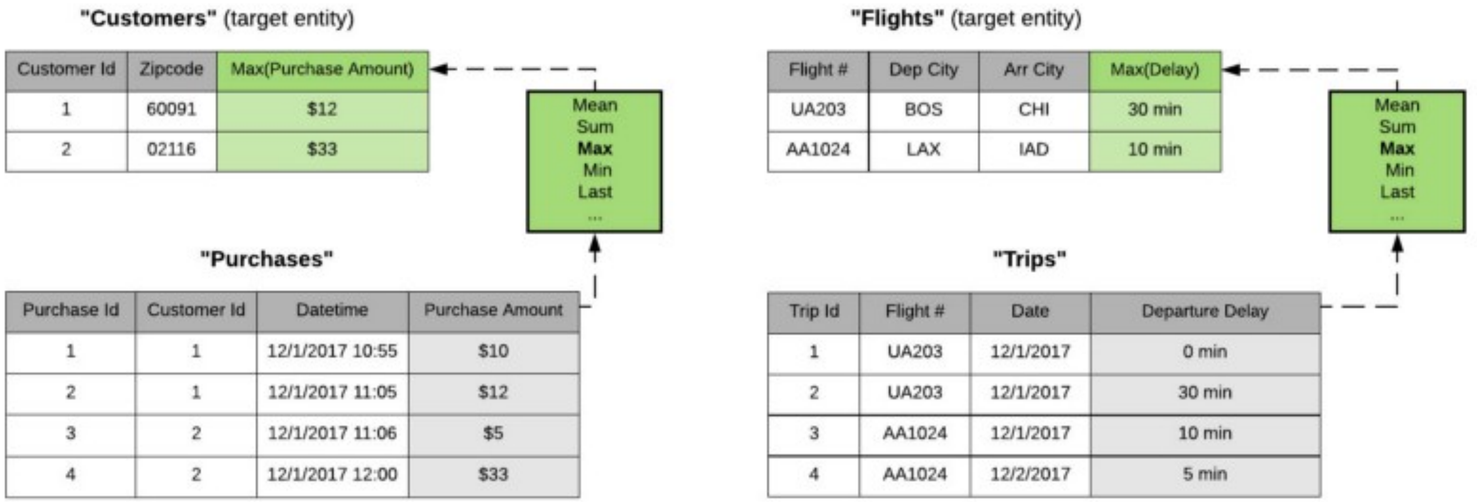
Feature Tools

- Open Source
- Automated Feature Engineering

https://www.featuretools.com/

Feature Tools

1. Features are derived from relationships between the data points in a dataset.

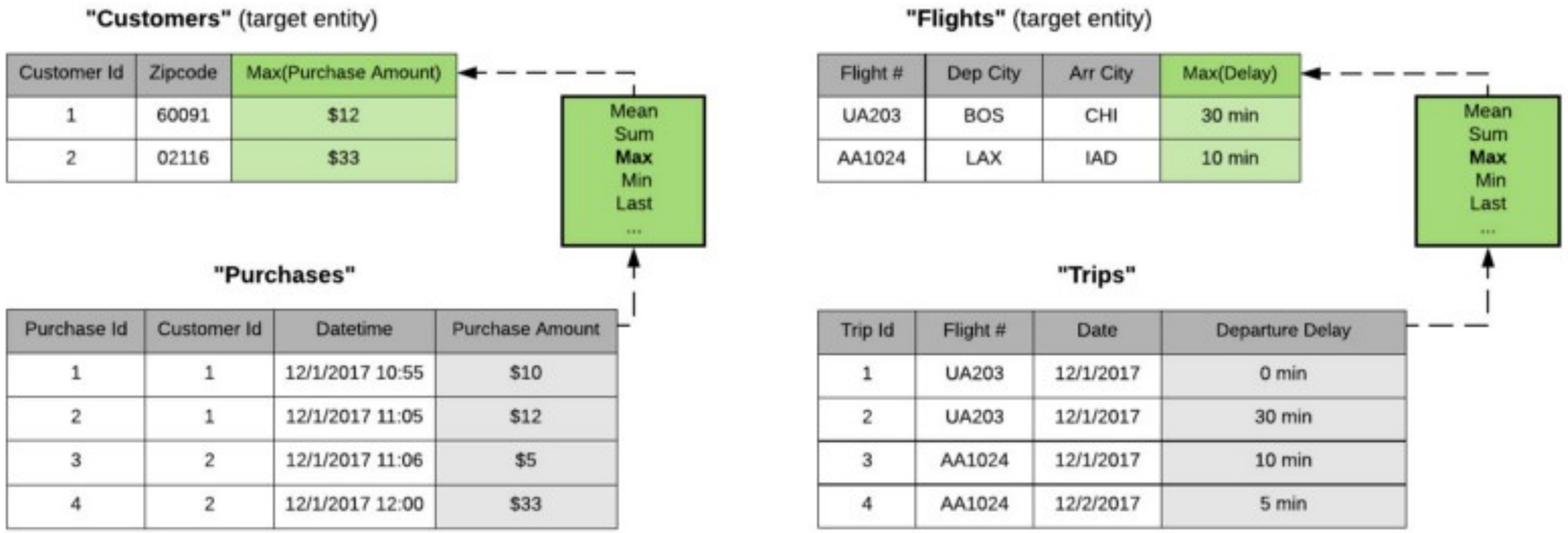


To calculate a customer's most expensive purchase, we apply the **Max** primitive to the purchase amount field in all related purchases. When we perform the same steps to a dataset of airplane flights, we calculate "the longest flight delay".

https://www.featuretools.com/

Feature Tools

2. Across datasets, many features are derived by using similar mathematical operations.

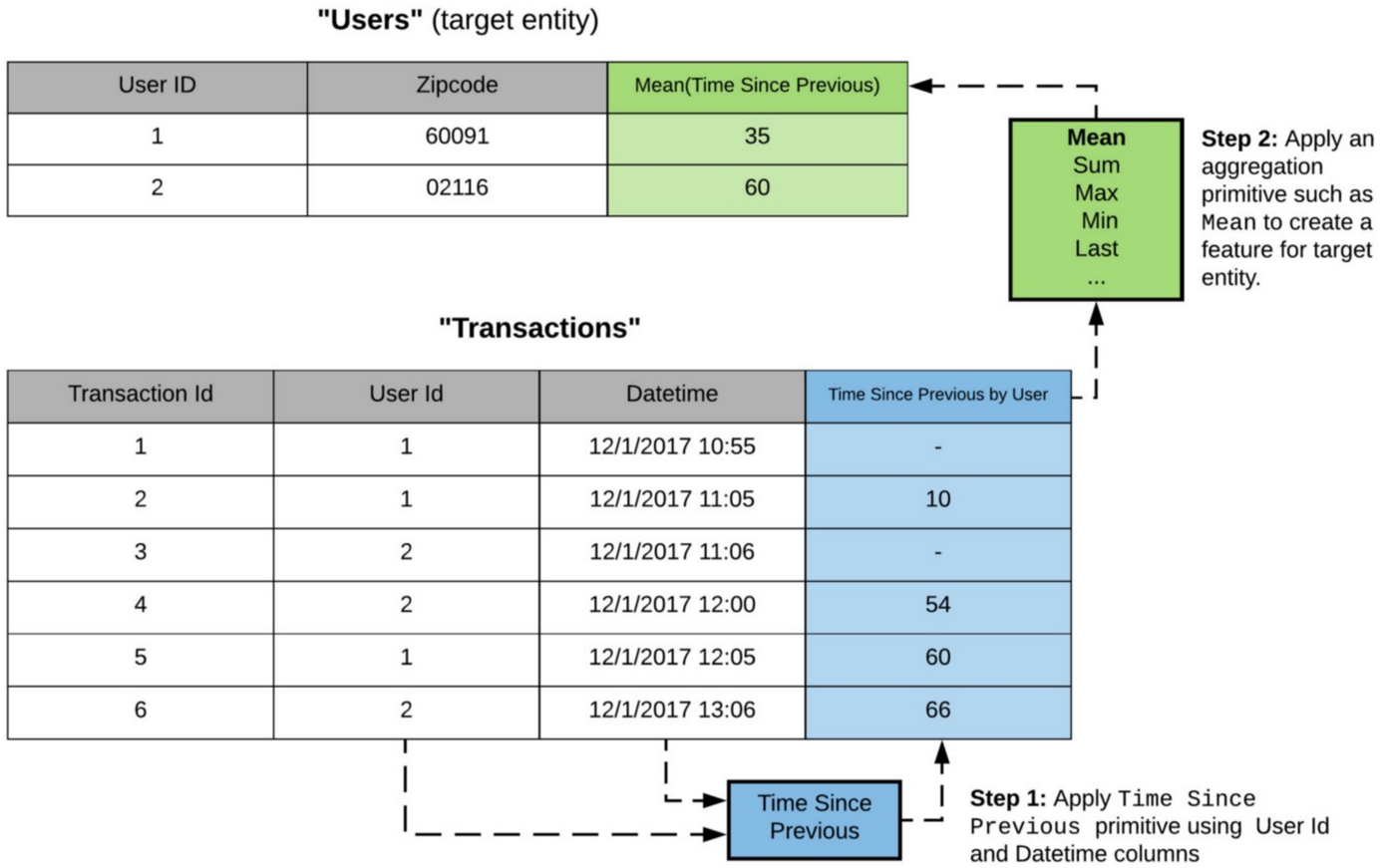


To calculate a customer's most expensive purchase, we apply the **Max** primitive to the purchase amount field in all related purchases. When we perform the same steps to a dataset of airplane flights, we calculate "the longest flight delay".

https://www.featuretools.com/

Feature Tools

3. New features are often composed from utilizing previously derived features.

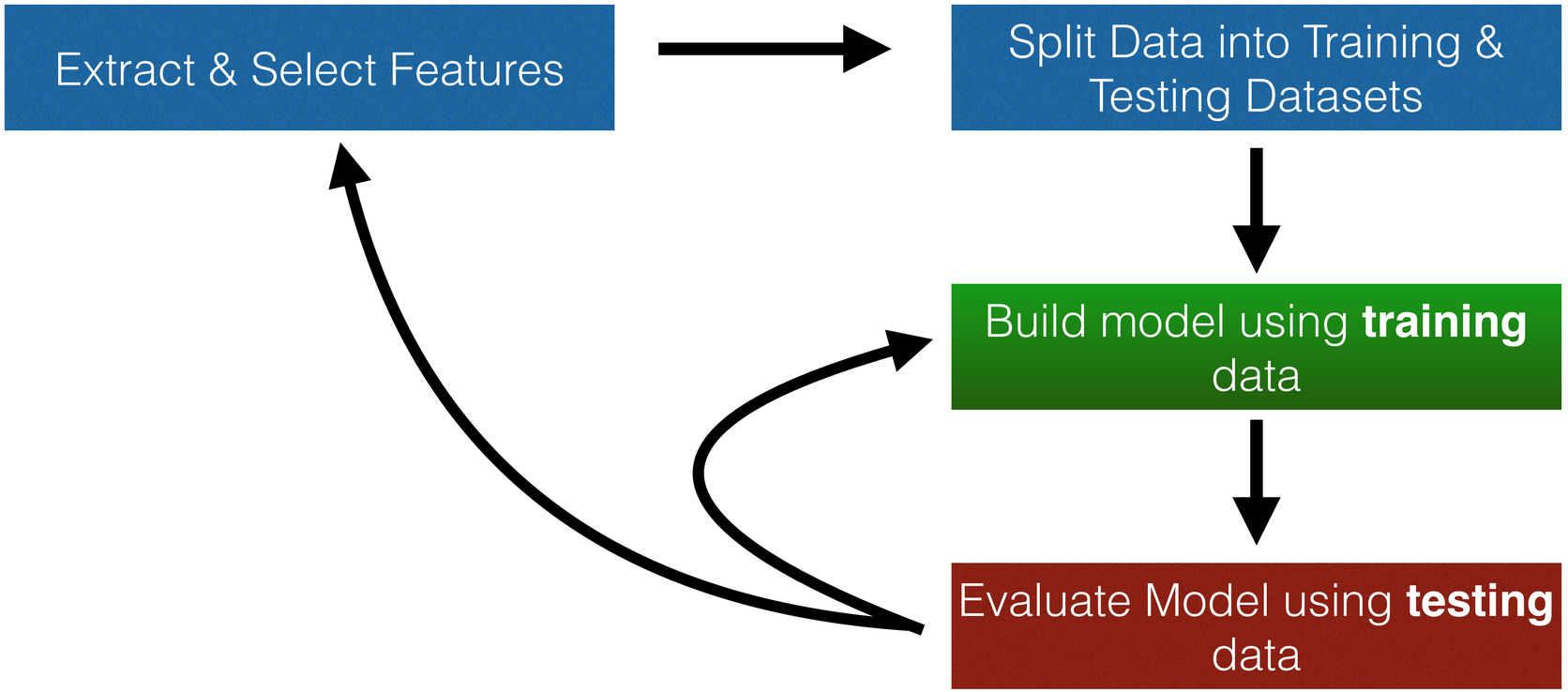


Building a Classifier

Supervised Learning

- The field which you are trying to predict is known as the **target**.
- Supervised learning depends on having a dataset of known data, known as **training data**.
- Supervised learning falls into two categories: Classification and Regression.
- **Many security problems are classification problems.**

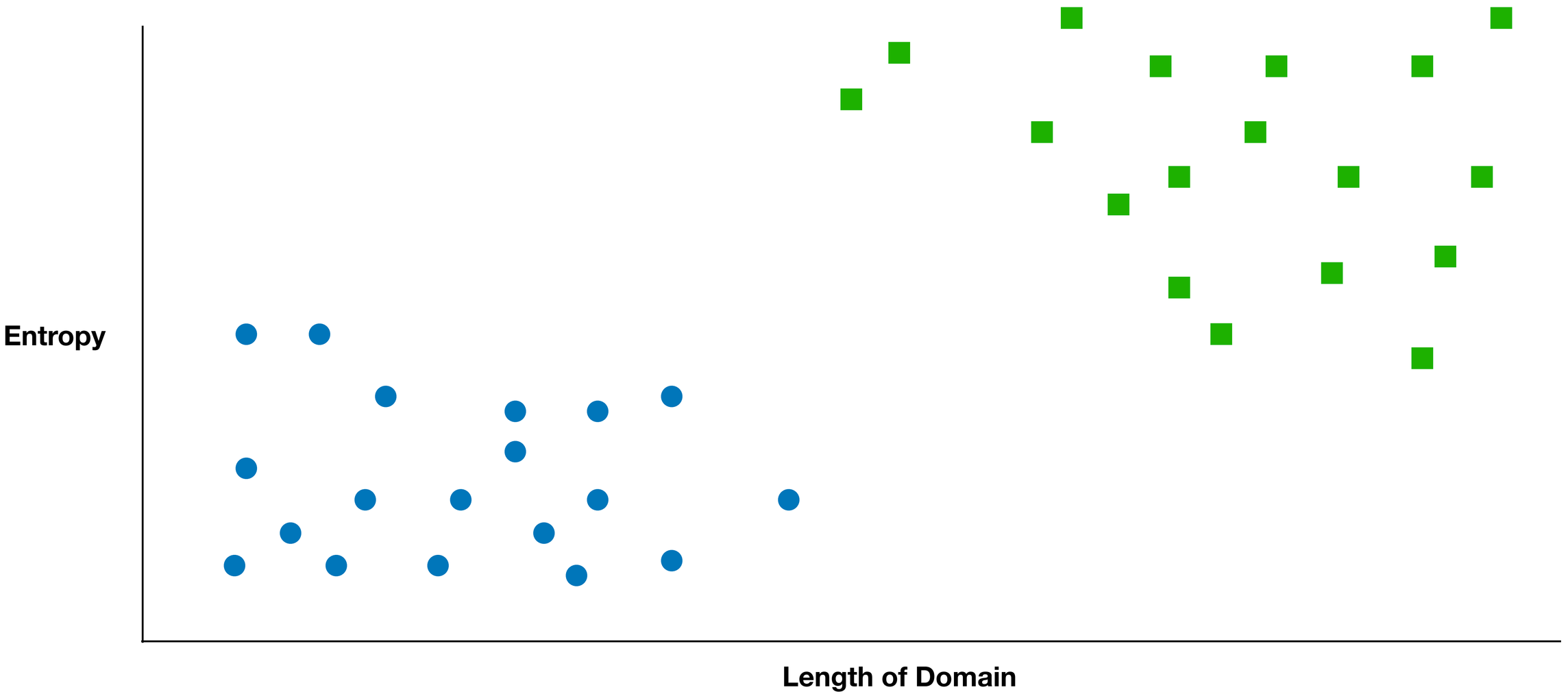
Supervised ML Process



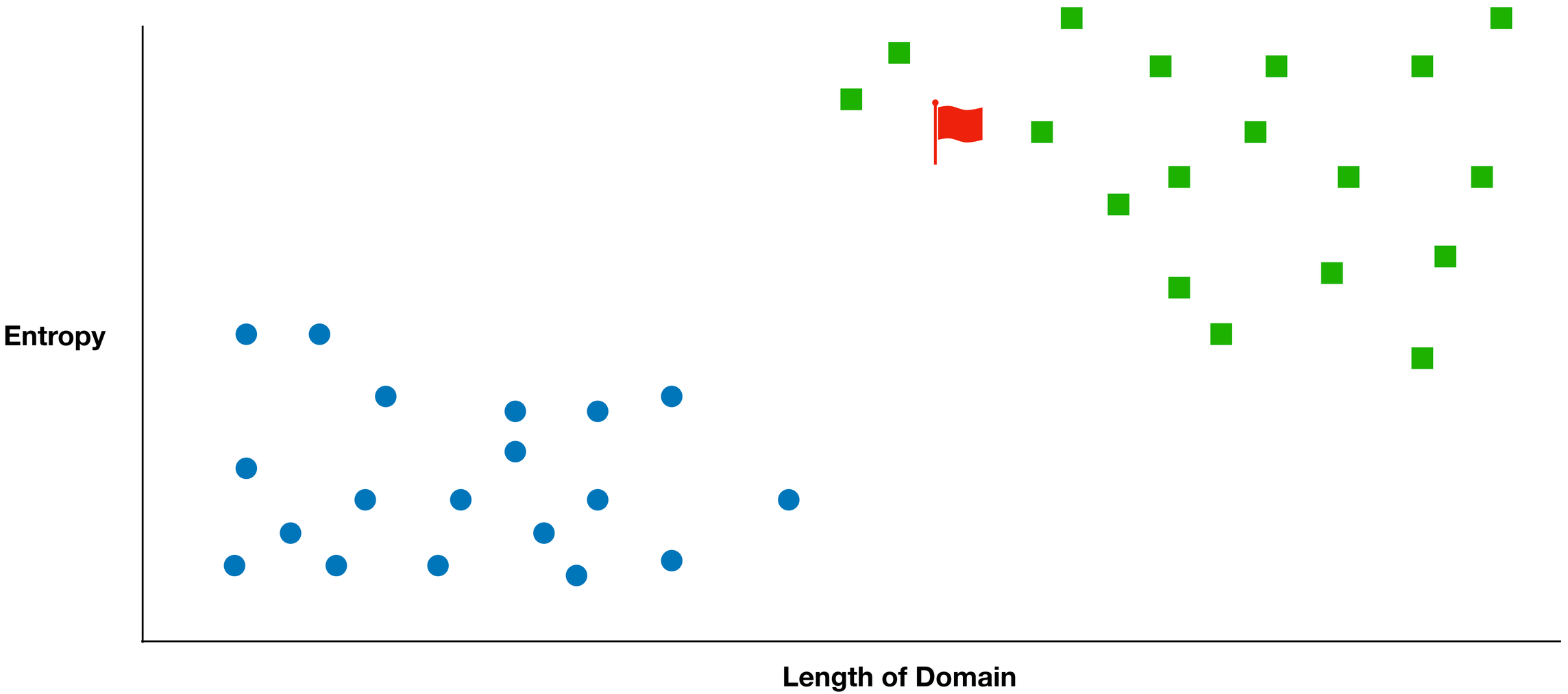
Classification Models

- K-Nearest Neighbors (K-NN) Classifiers
- Random Forest
- Decision Trees
- Support Vector Machine (SVM)
- Logistic Regression

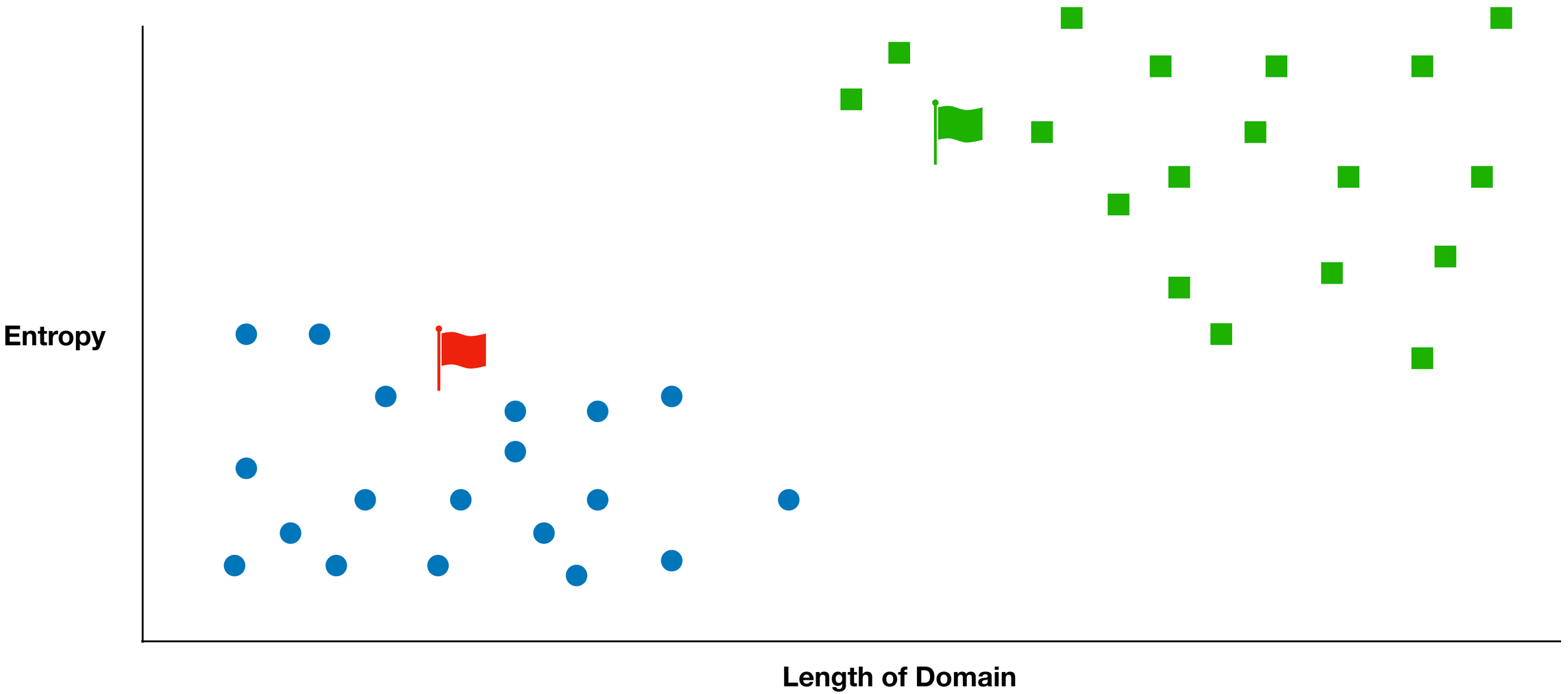
How do they work?



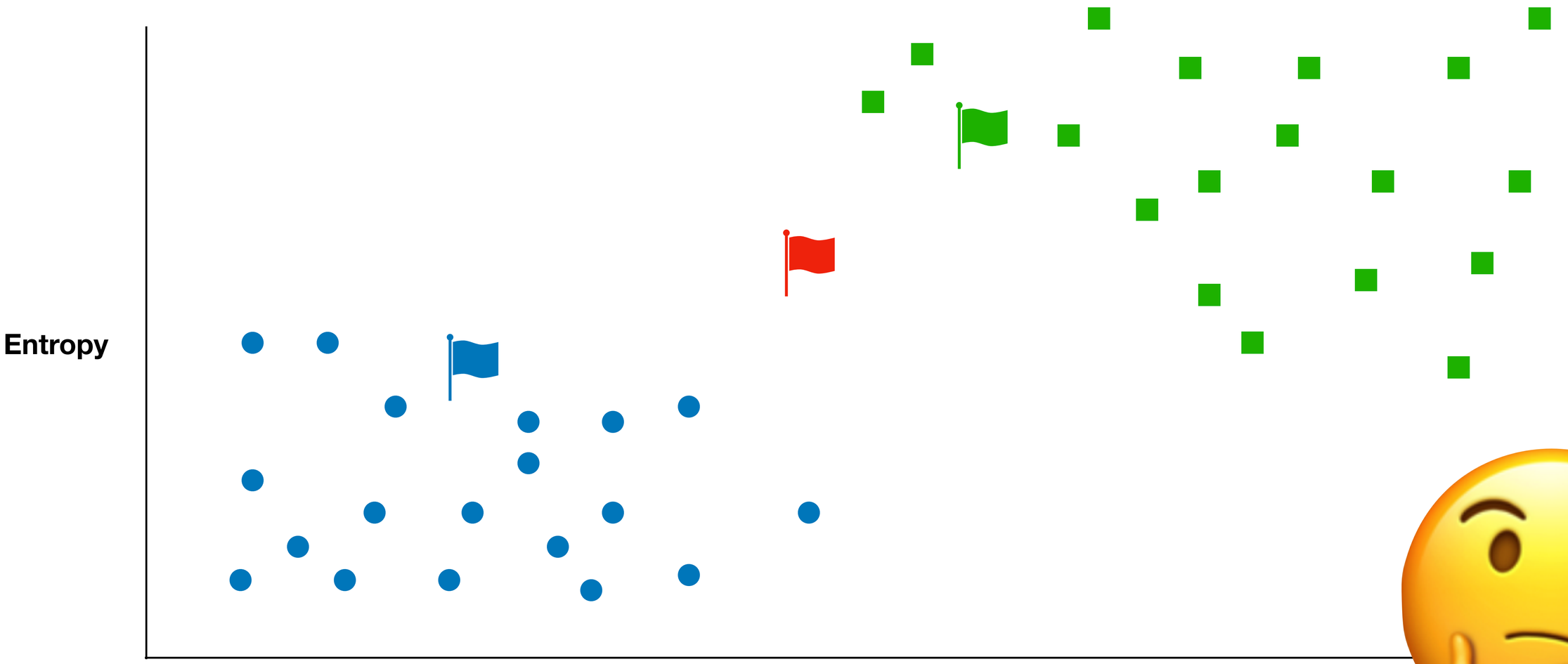
How do they work?



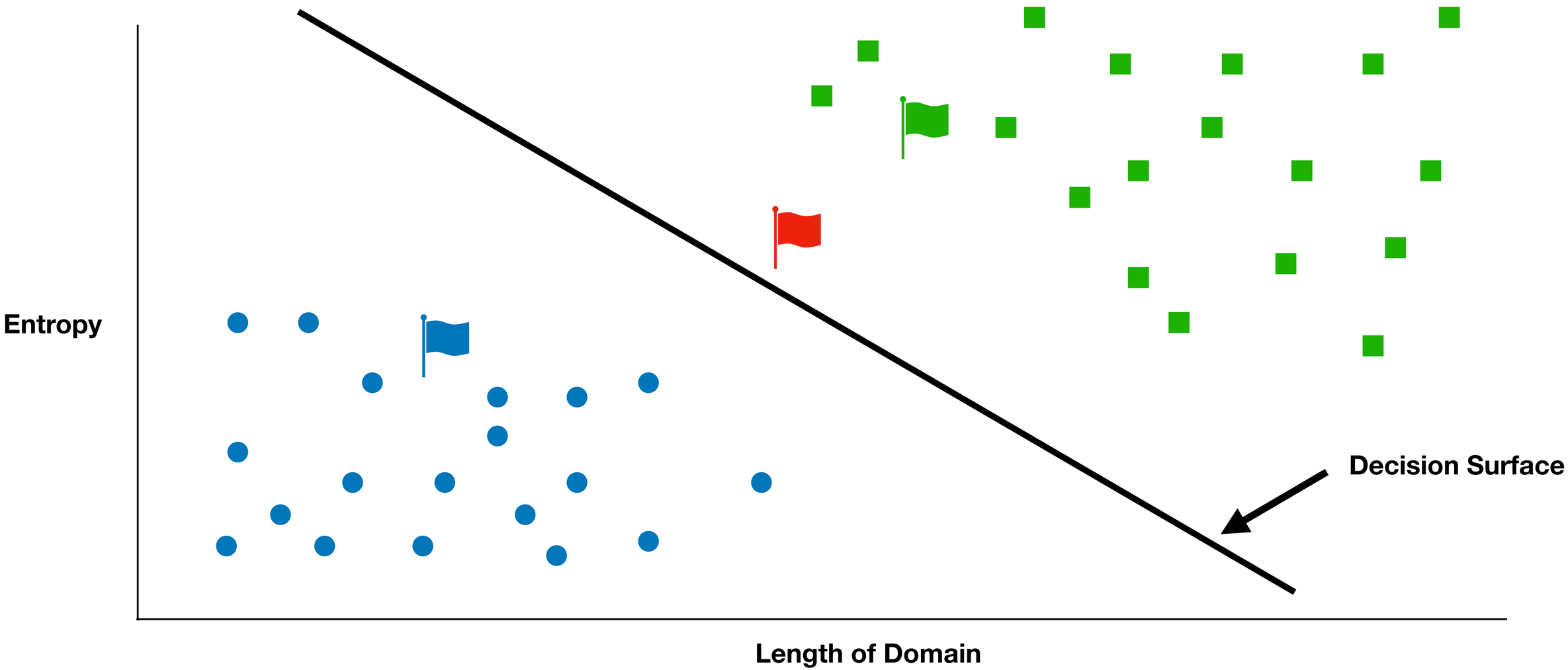
How do they work?

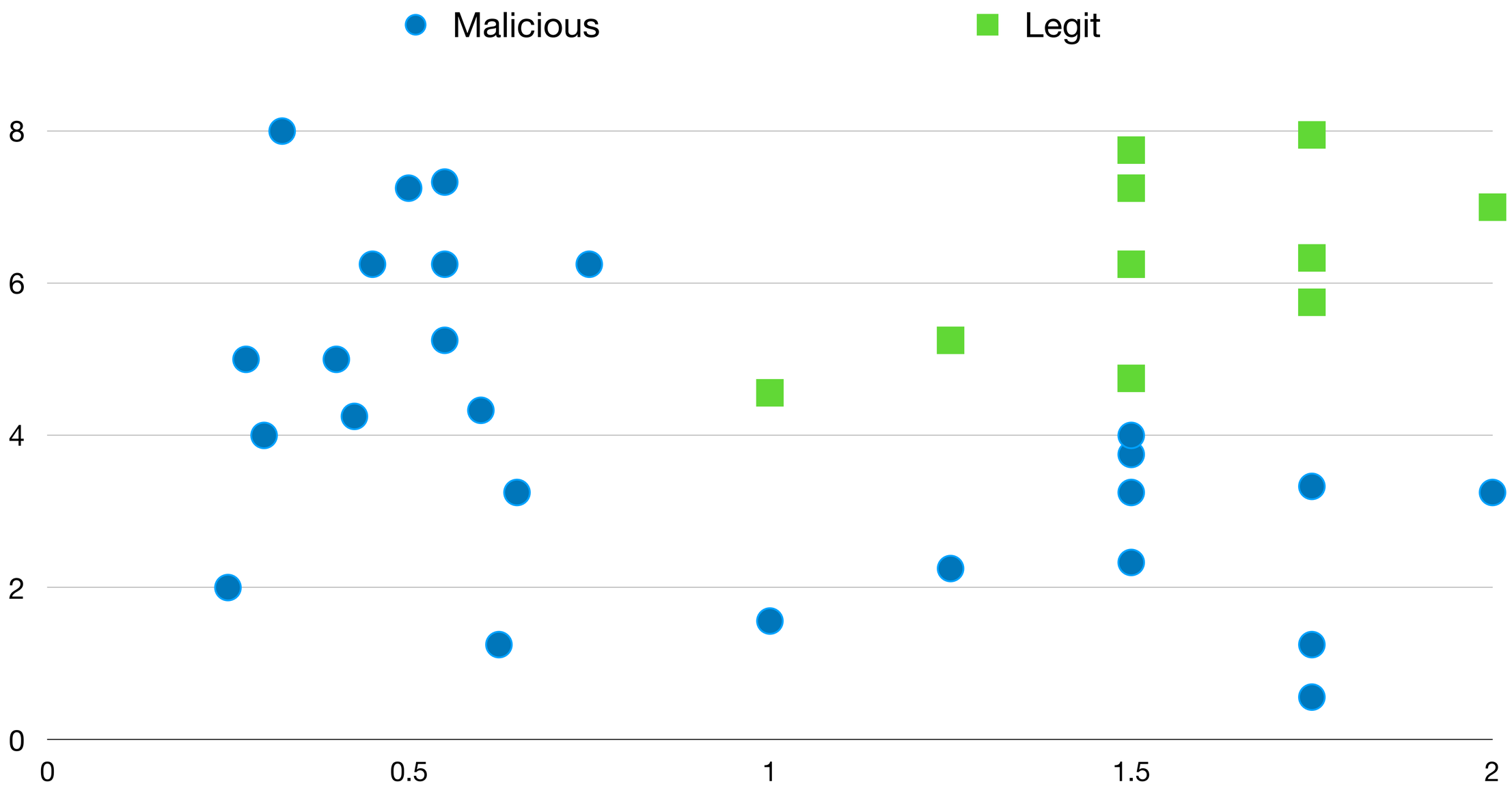


How do they work?



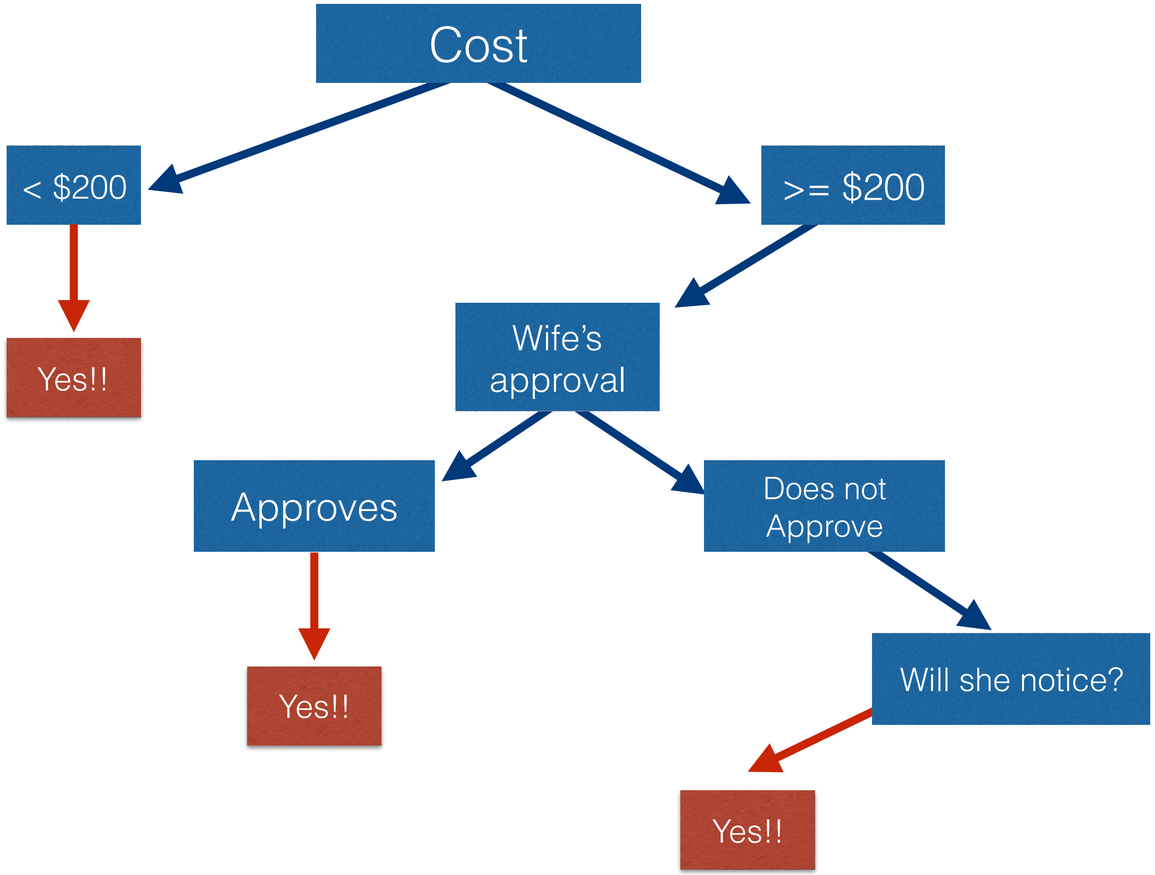
How do they work?

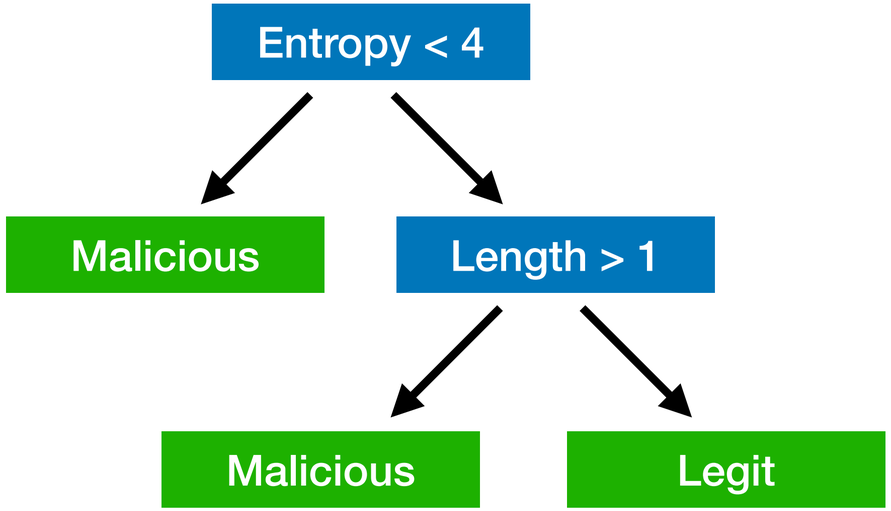


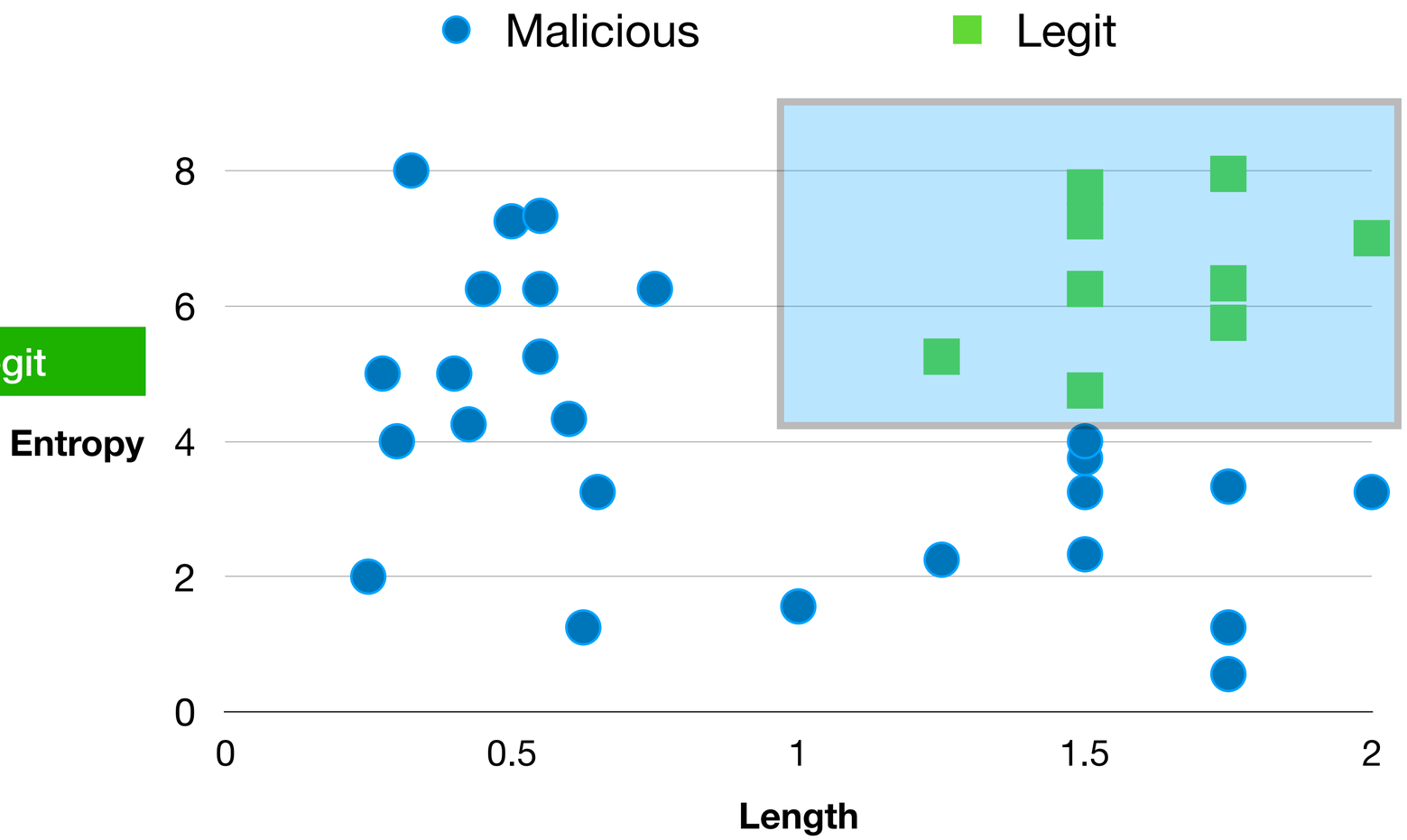

Malicious
Legit
8
6
4
2
0
0
0.5
1
1.5
2

Simple Decision Tree (DT)

Should I Buy a Tech Gadget?




Entropy < 4
Malicious
Length > 1
Malicious
Legit


Malicious
Legit
8
6
4
2
0
Entropy
0
0.5
1
1.5
2
Length

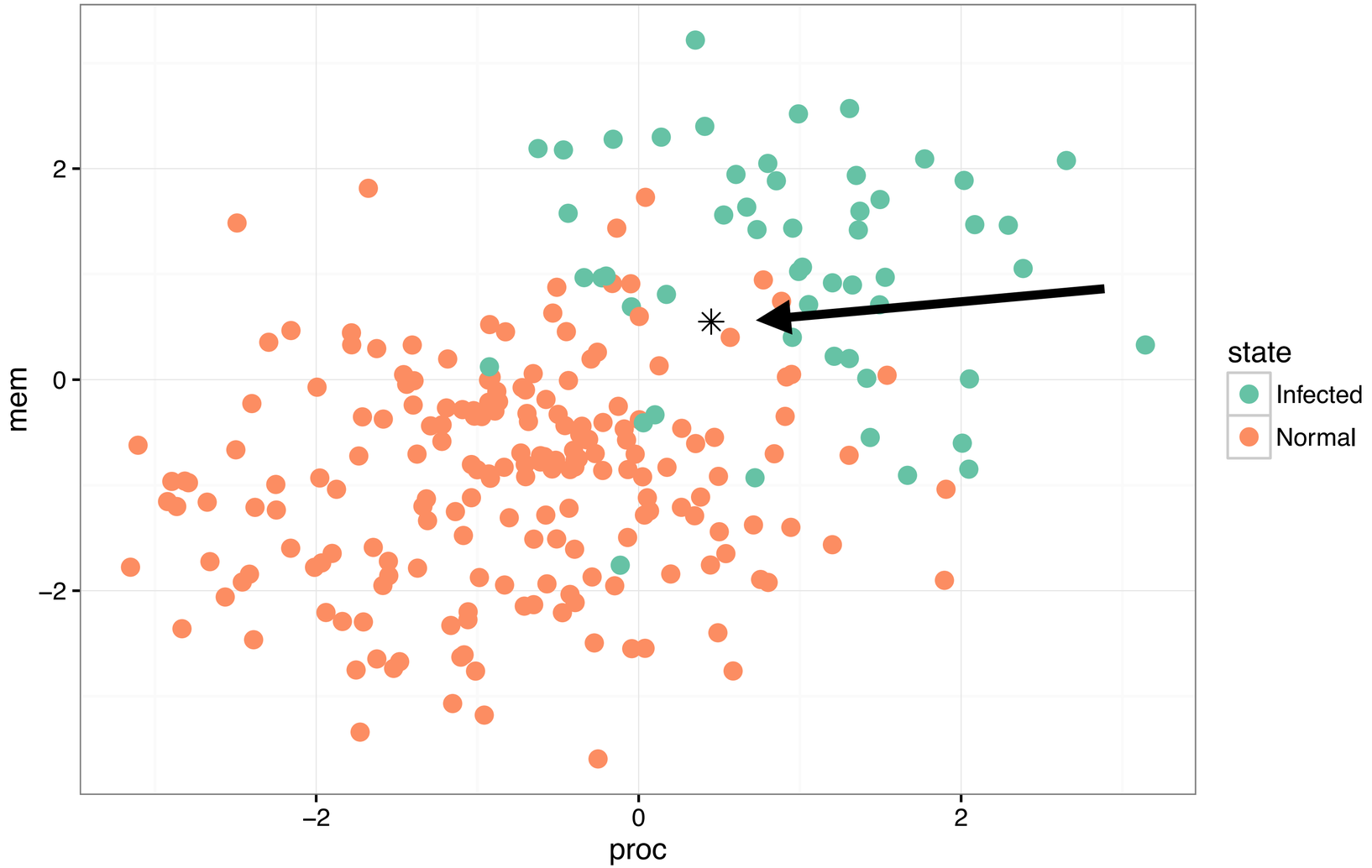
K-Nearest Neighbors Classifier

K-Nearest Neighbors

Define K, and for every unknown observation:

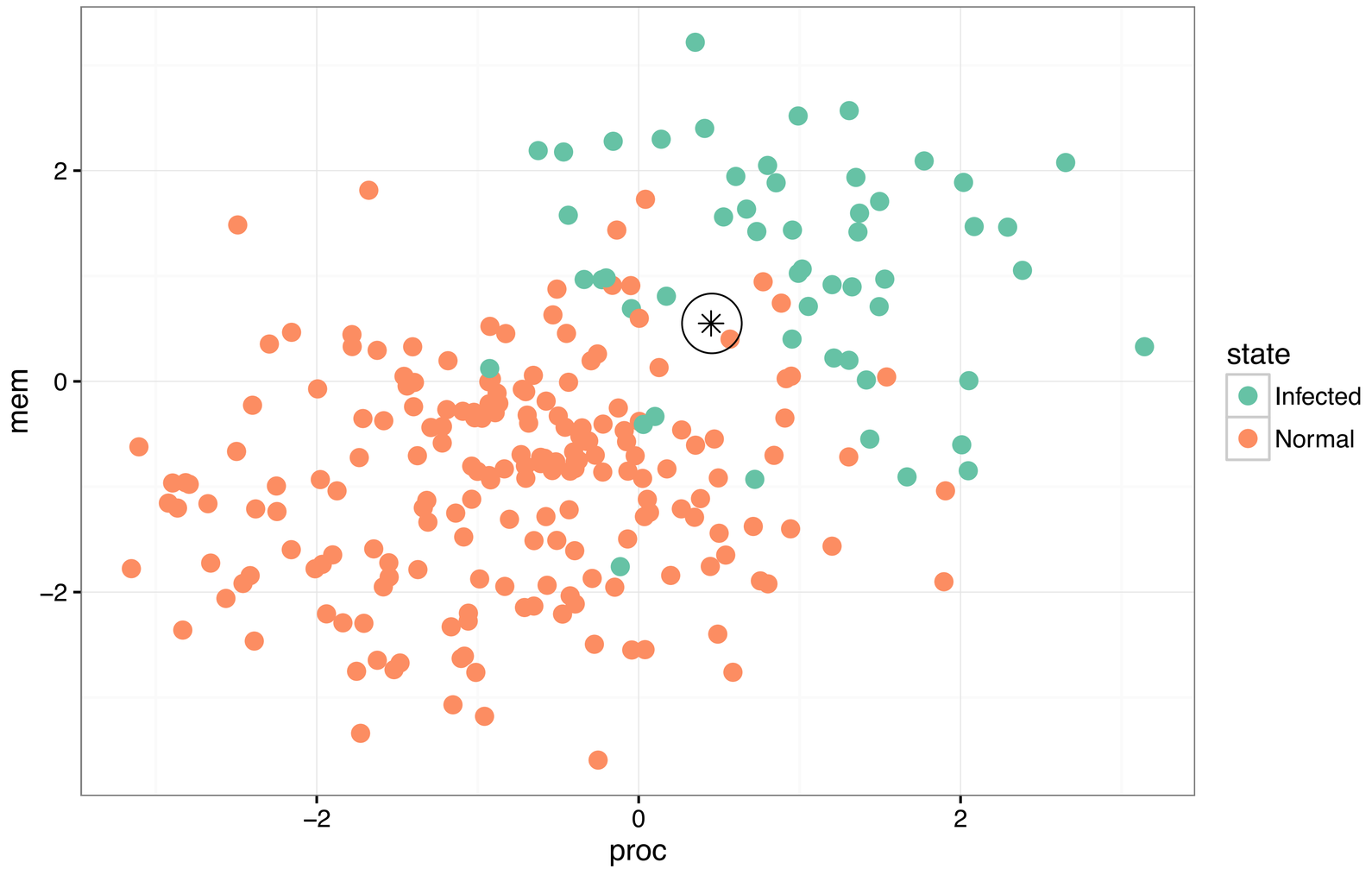
1. Find k-nearest neighbors (by *distance*)
2. Assign class based on dominate class
3. Optional weight by distance

K-Nearest Neighbors



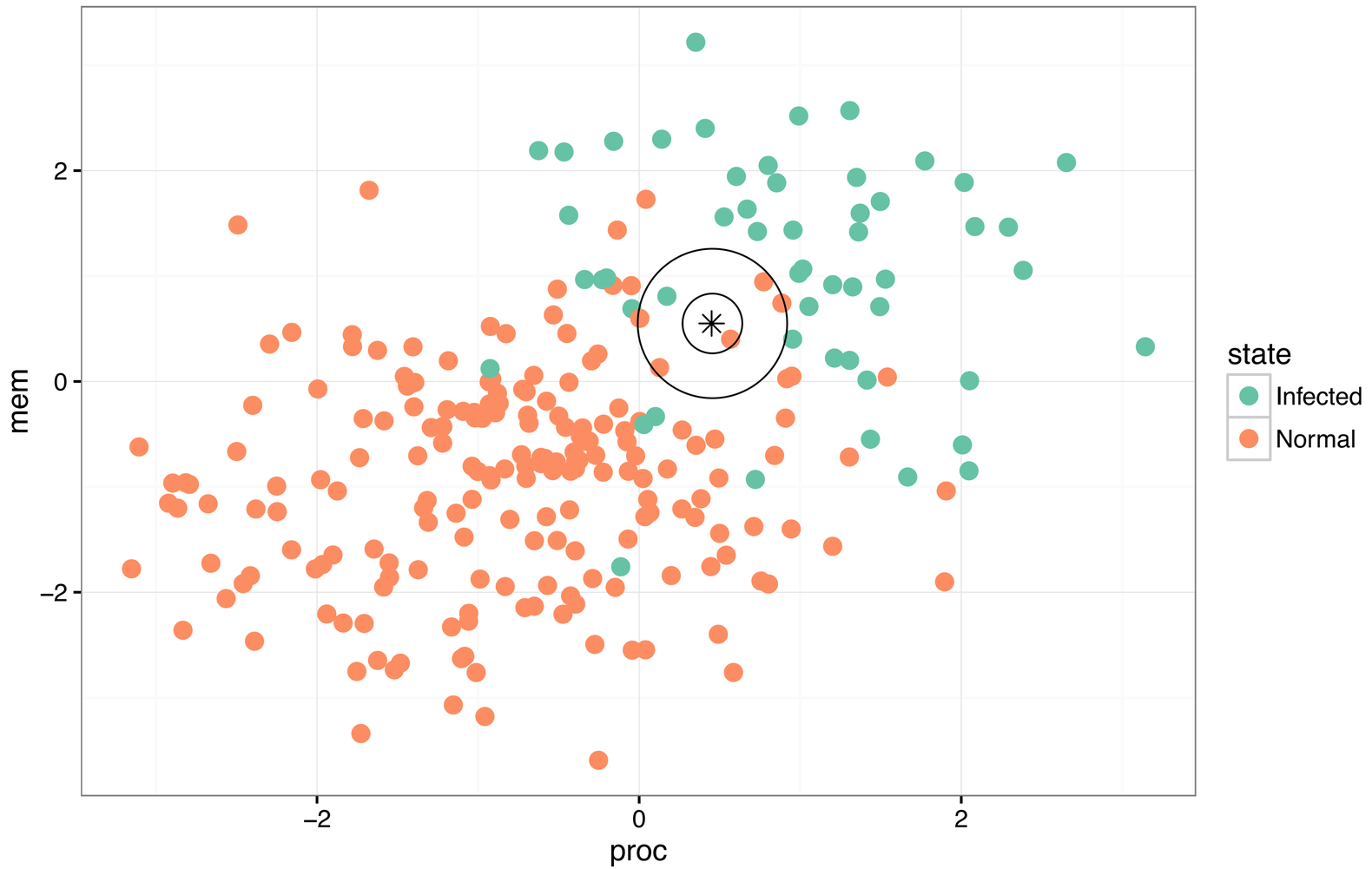
Data Driven Security: Jay Jacobs, Bob Rudis

K-Nearest Neighbors



k = 1
1 Normal
0 Infected
Assign “Normal”

K-Nearest Neighbors



k = 5
4 Normal
1 Infected
Assign “Normal”

K-Nearest Neighbors

Further reading: https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

K-NN Classifiers

Advantages of K-NN Classifiers

- Robust to noisy, non-linear training data
- Flexible to feature / distance choices
- Works with large datasets
- Naturally handles multi-class cases

Disadvantages of K-NN Classifiers:

- Need to determine the value of K
- Training is fast but predictions are slow. Indexing can help
- Must have meaningful distance function

Ensembles

Ensembles

Ensemble learning is the process of combining several predictive models in order to produce a combined model that is more accurate than any individual model.

- **Regression:** take the average of the predictions
- **Classification:** take a vote and use the most common prediction, or take the average of the predicted probabilities

Ensembles

For ensembling to work well, the models must have the following characteristics:

- **Accurate:** they outperform random guessing
- **Independent:** their predictions are generated using different processes

The big idea: If you have a collection of individually imperfect (and independent) models, the "one-off" mistakes made by each model are probably not going to be made by the rest of the models, and thus the mistakes will be discarded when averaging the models.

Note: As you add more models to the voting process, the probability of error decreases, which is known as Condorcet's Jury Theorem.

Random Forest

Random Forests are a **slight variation of bagged trees** that have even better performance:

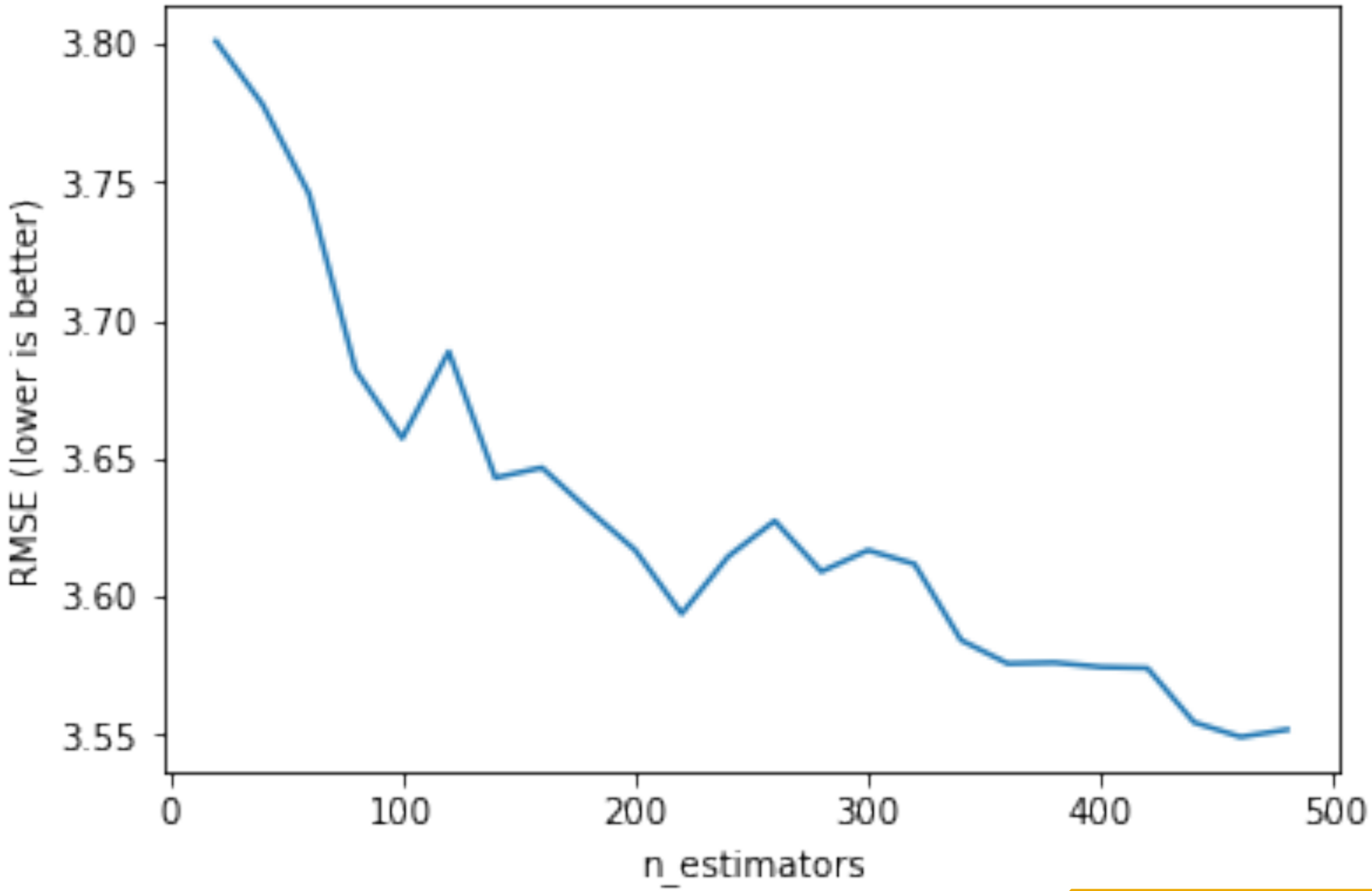
- Just like bagging, we create an ensemble of decision trees using bootstrapped samples of the training set.
- However, when building each tree, each time a split is considered, a **random sample of *m* features** is chosen as split candidates from the **full set of *p* features**. The split is only allowed to use **one of those *m* features**.
    - A new random sample of features is chosen for **every single tree at every single split**.
    - For **classification**, *m* is typically chosen to be the square root of *p* (the total number of features).
    - For **regression**, *m* is typically chosen to be somewhere between *p*/3 and *p*.

Tuning a Random Forest

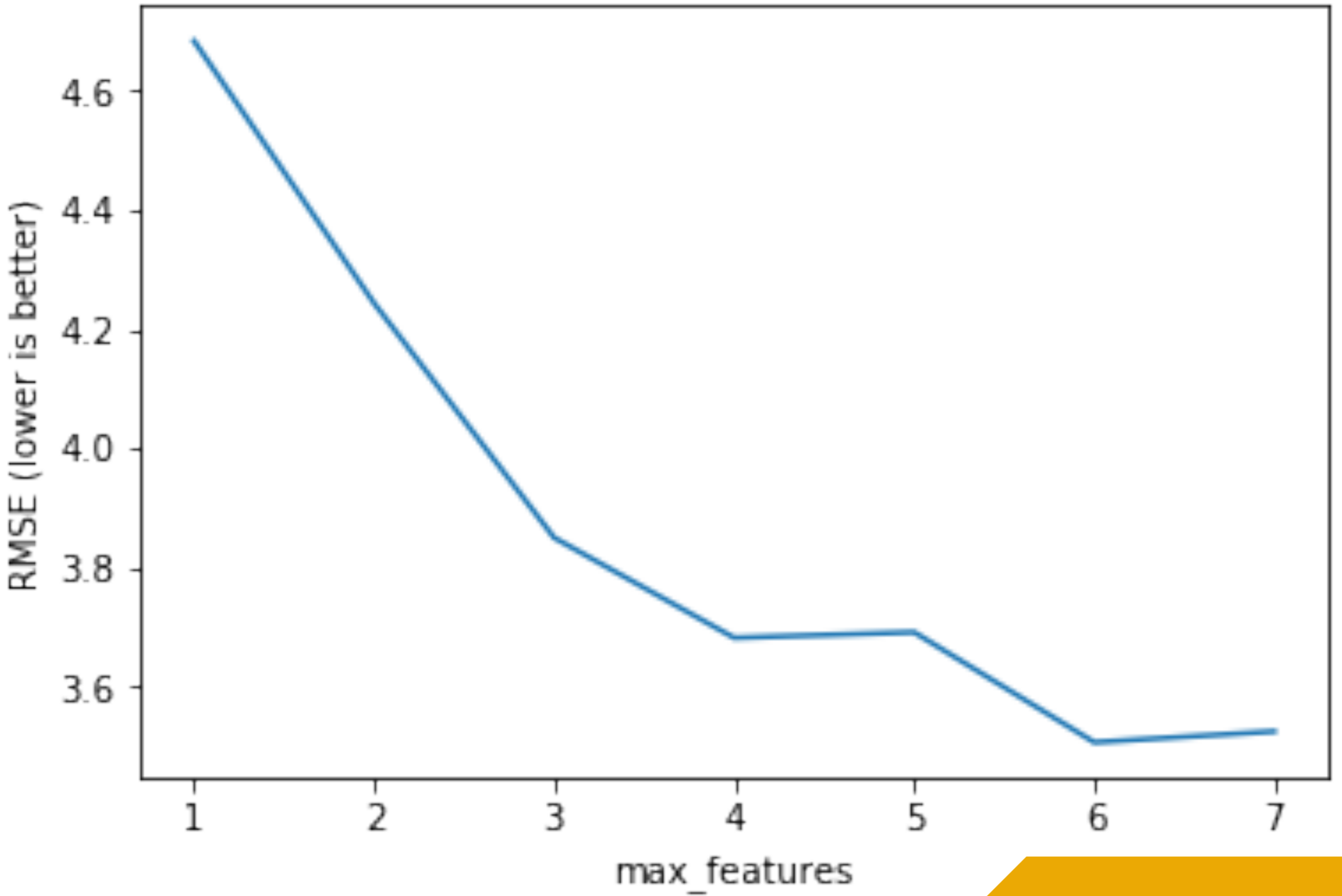
2 important parameters that should be tuned when creating a random forest model are:

- The number of trees to grow (called **n_estimators** in scikit-learn)
- The number of features that should be considered at each split (called **max_features** in scikit-learn)

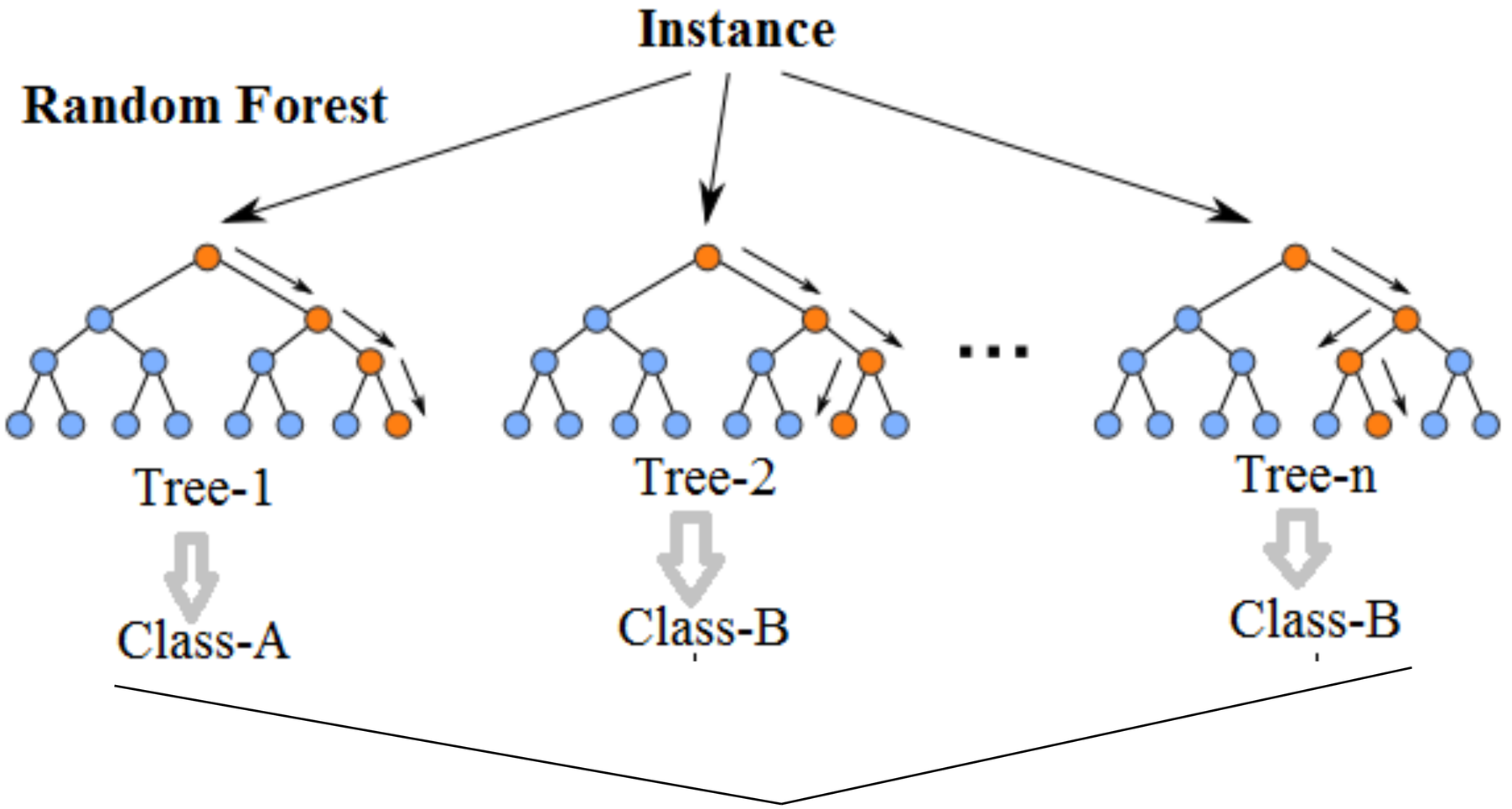
Tuning a Random Forest



Tuning a Random Forest



Tuning a Random Forest



Class A: p% Class B: 1-p%

Majority Win

Random Forests

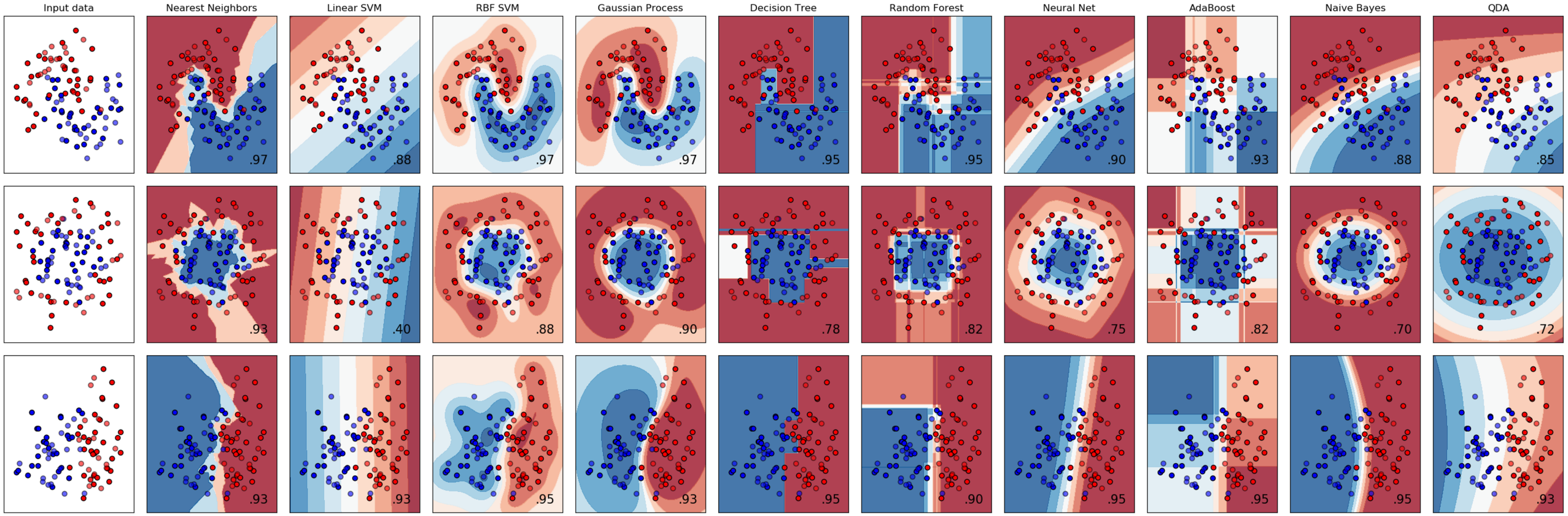
Advantages of Random Forests:

- Performance is competitive with the best supervised learning methods
- Provides a more reliable estimate of feature importance
- Allows you to estimate out-of-sample error without using train/test split or cross-validation

Disadvantages of Random Forests:

- Less interpretable
- Slower to train
- Slower to predict

Classification Algorithms in Python's Scikit-Learn

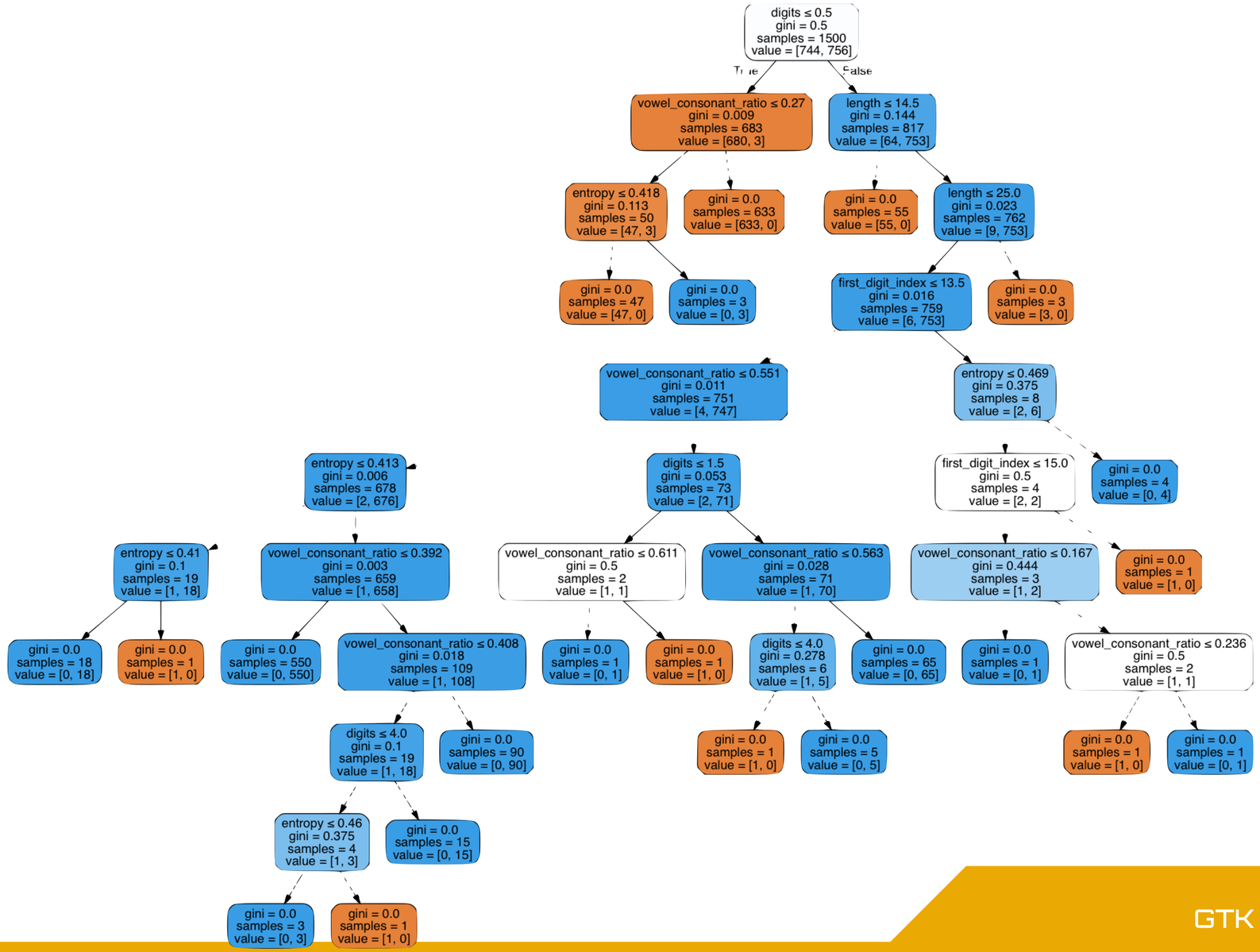


http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Example: DGA Classification

Legitimate Domains	Suspicious Domains
facebook.com	dmvyqbvclnupvsjpysr.com
db.com	nqsb0g1nn5efb1ma036d1hx1yxy.org
oreilly.com	btoovqsxlmlkgb.com
thedataist.com	wxkbfcxjxffr.org
lemonde.fr	1m5zs931330zfe1tba6ov1ur4zut.ru
1800flowers.com	1363c0g18y9x9j1xjeuvull2d4o.com

Decision Tree for DGA



Implementation

Scikit-Learn Process

The basic process for any supervised learning is the same for any supervised learning in Python.

1. Create the Classifier or Regressor object.

```
clf = RandomForestClassifier()
```

2. Call the **`.fit()`** method using the **training feature matrix (X)** and the **training target vector (y)**.

```
clf.fit( training_features, training_target )
```

3. Call the **`.predict()`** method using a feature matrix

#Returns vector of predictions

```
clf.predict( testing_features )
```

Train-Predict in sklearn

```
## Support Vector Machine Classification

clf = svm.SVC()
clf = clf.fit(X, target)
target_pred = clf.predict(X)
```

clf means
classifier

Train-Predict in sklearn

```
## Decision Tree Classification

clf = tree.DecisionTreeClassifier()
clf = clf.fit(X, target)
target_pred = clf.predict(X)
```

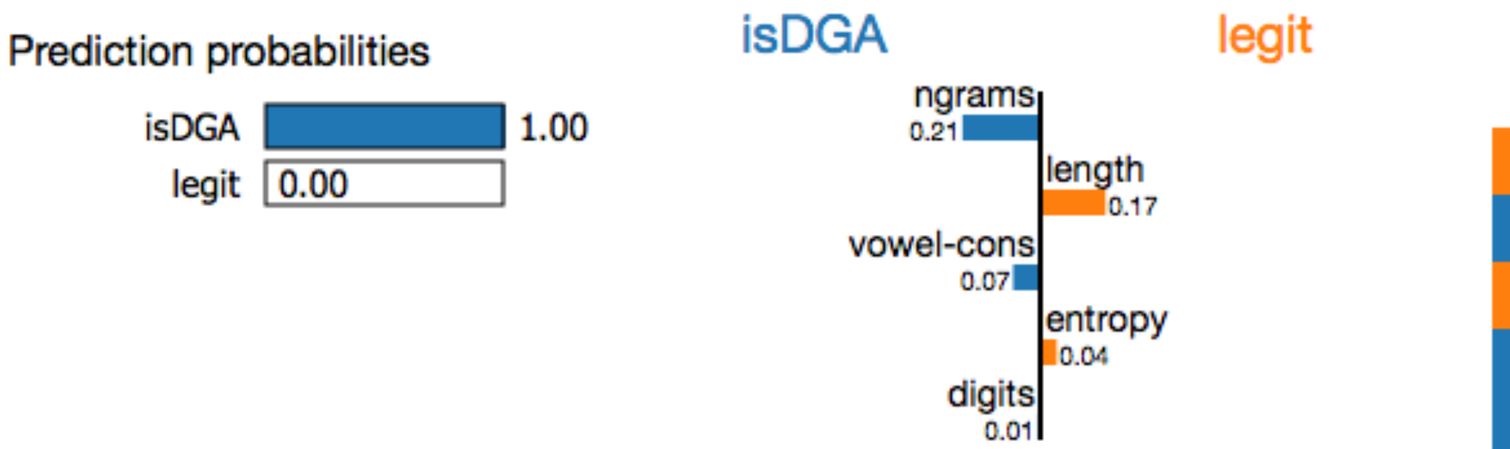
clf means classifier

Classification Algorithms in Scikit-Learn

- Generalized Linear Models
- Kernel Ridge
- Support Vector Machines
- Nearest Neighbors
- Gaussian Processes
- Naive Bayes
- Trees
- Neural Networks
- AdaBoost
- Gradient Tree Boosting
- Ensemble methods

"Why should I trust you?"
Explaining the predictions of any classifier

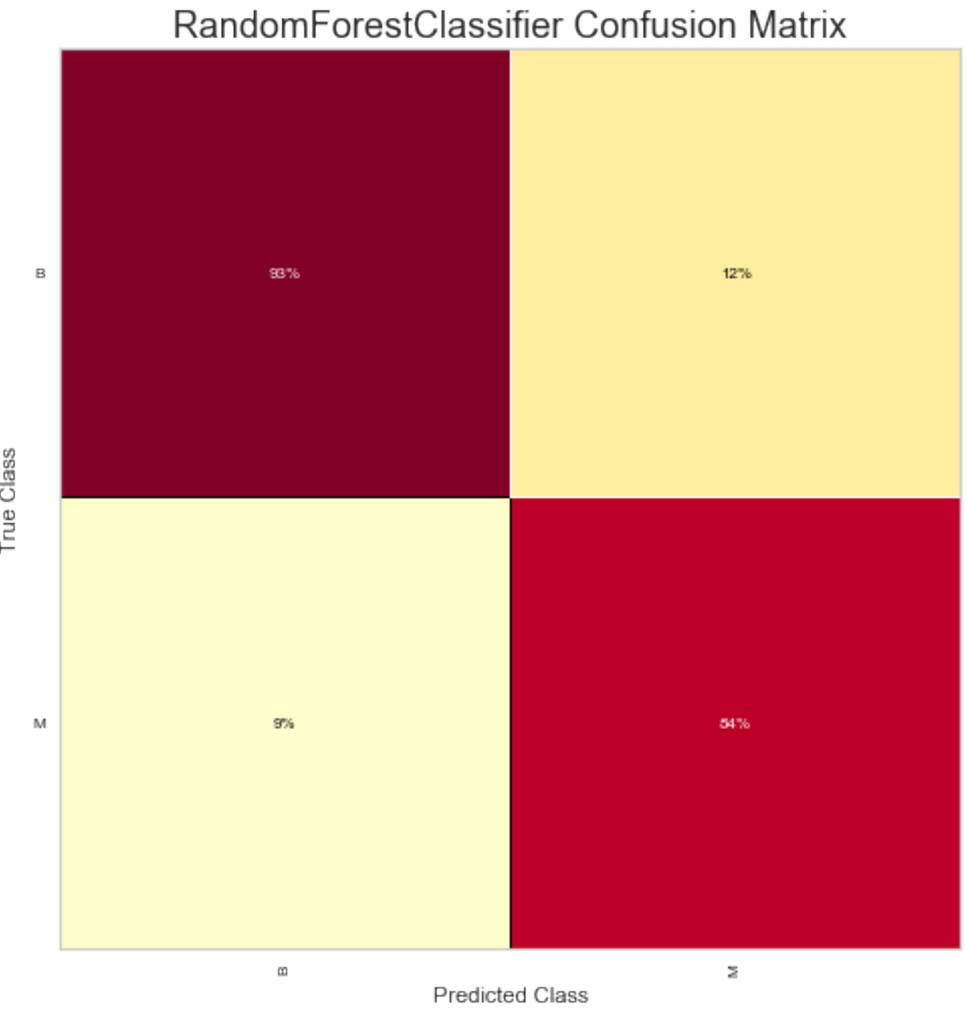
```
import lime
import lime.lime_tabular
explainer = lime.lime_tabular.LimeTabularExplainer(<training_data>,
        feature_names=<feature names>,
        class_names=<class names>,
        discretize_continuous=False)

exp = explainer.explain_instance( <test_row>,
   <model>.predict_proba,
   num_features=5,
   top_labels=1)

exp.show_in_notebook(show_table=True, show_all=False)
```



https://github.com/marcotcr/lime

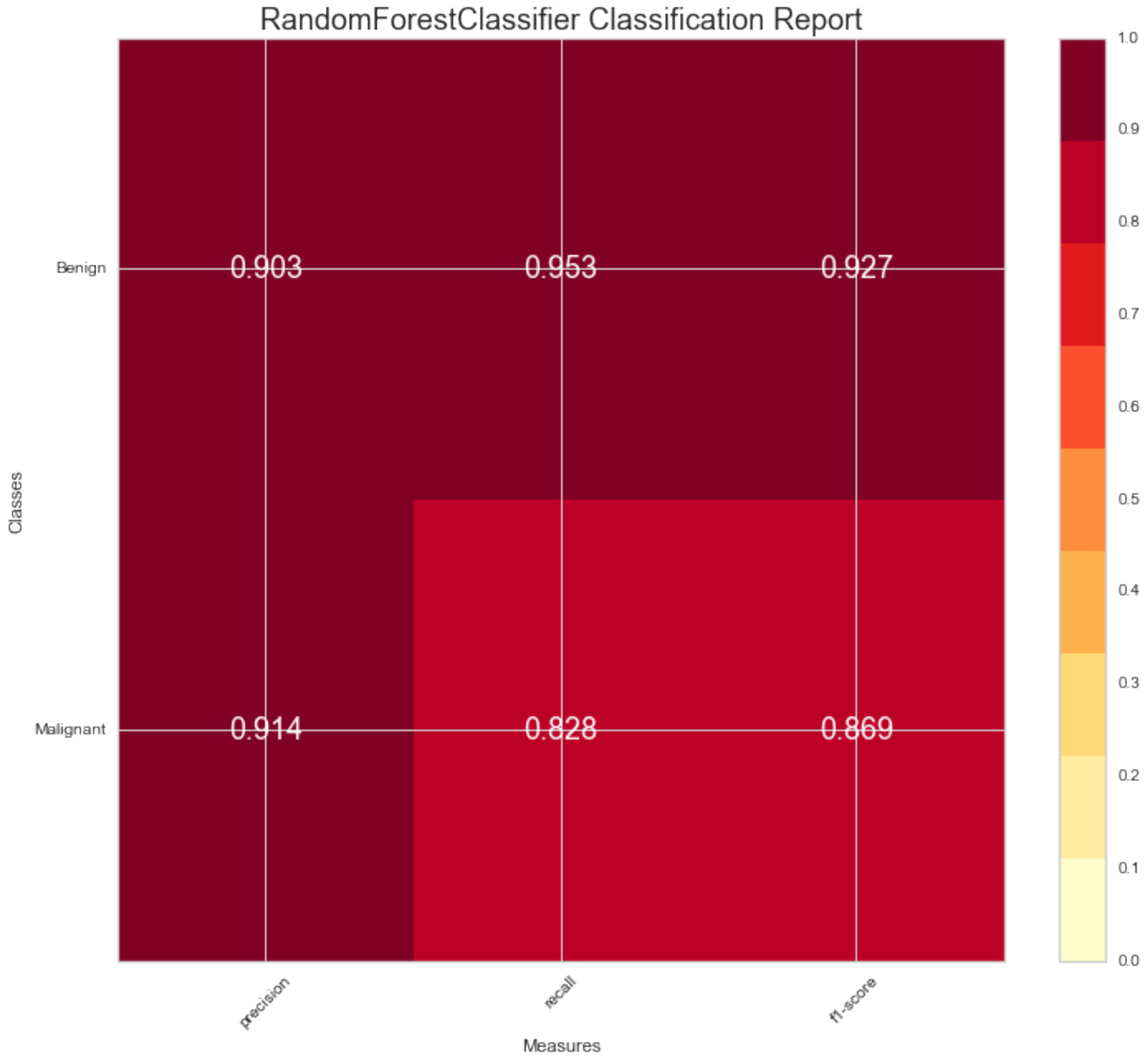
Assessing Model Performance

Visualizing a Confusion Matrix

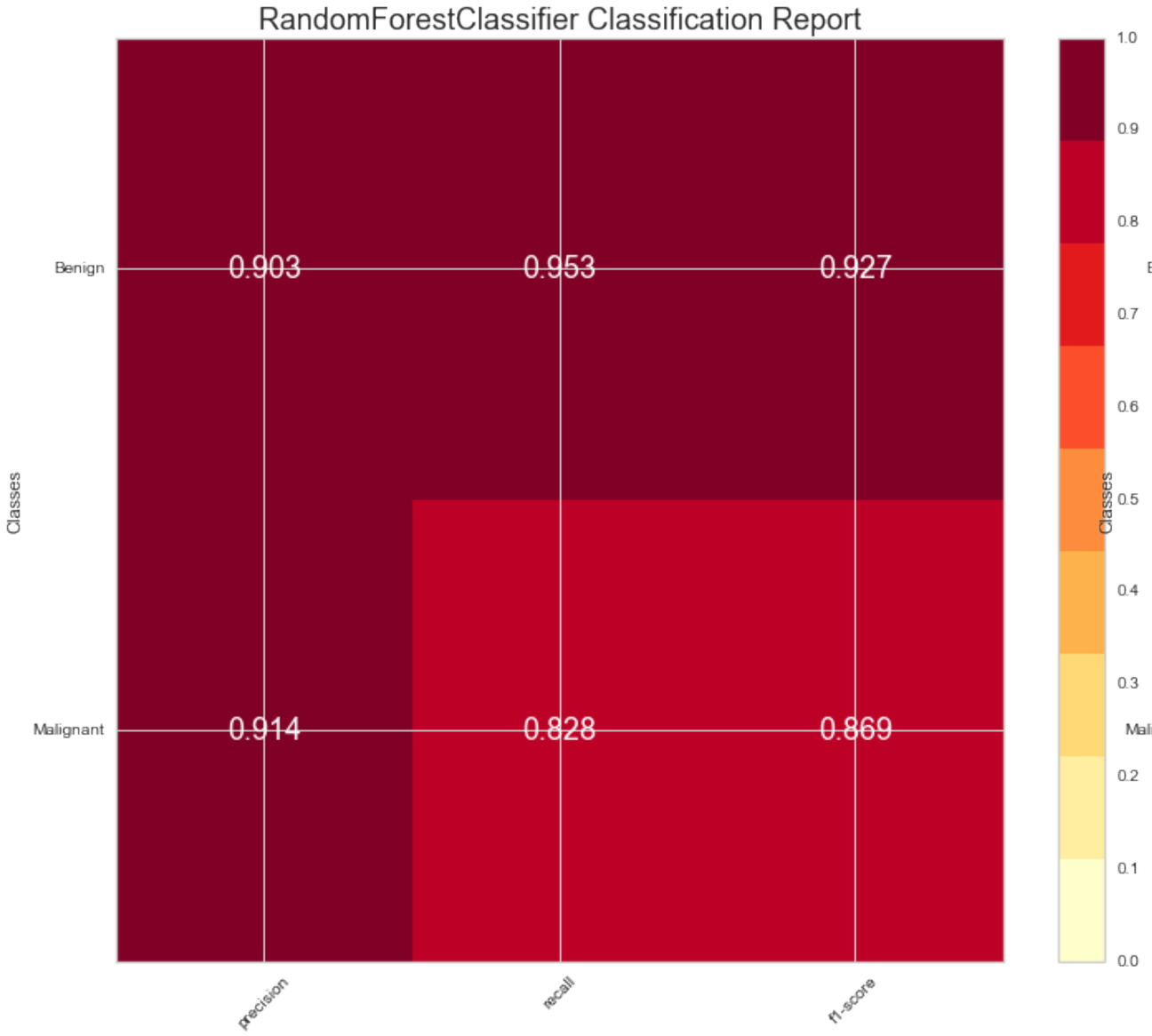


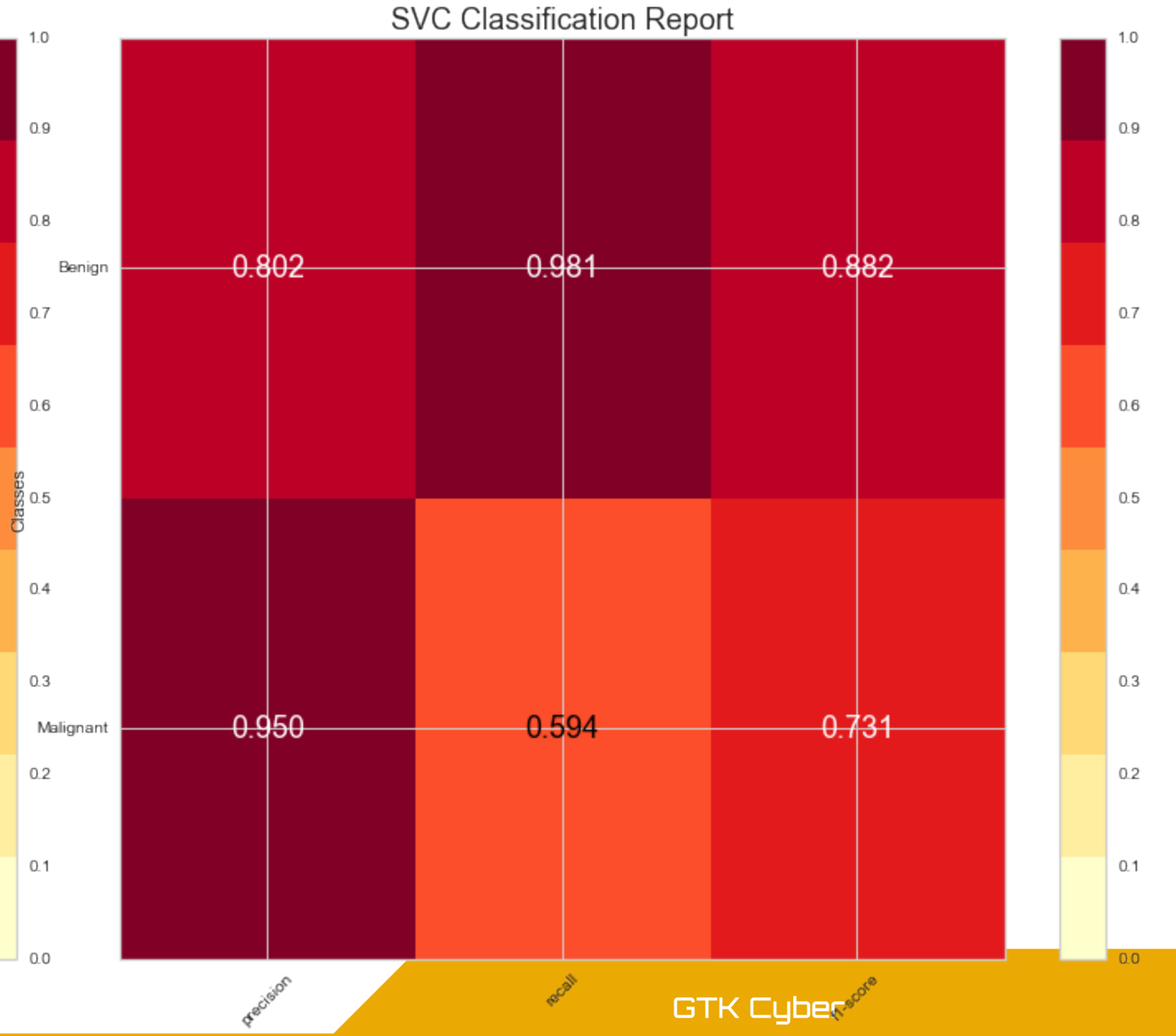

```
from yellowbrick.classifier import ConfusionMatrix

#Create the models for both the Random Forest
random_forest_model = RandomForestClassifier()

random_forest_conf_matrix = ConfusionMatrix(
                                    random_forest_model )
random_forest_conf_matrix.fit( X_train, y_train )
random_forest_conf_matrix.score( X_test, y_test )
random_forest_conf_matrix.poof()
```

Visualizing a Classification Report

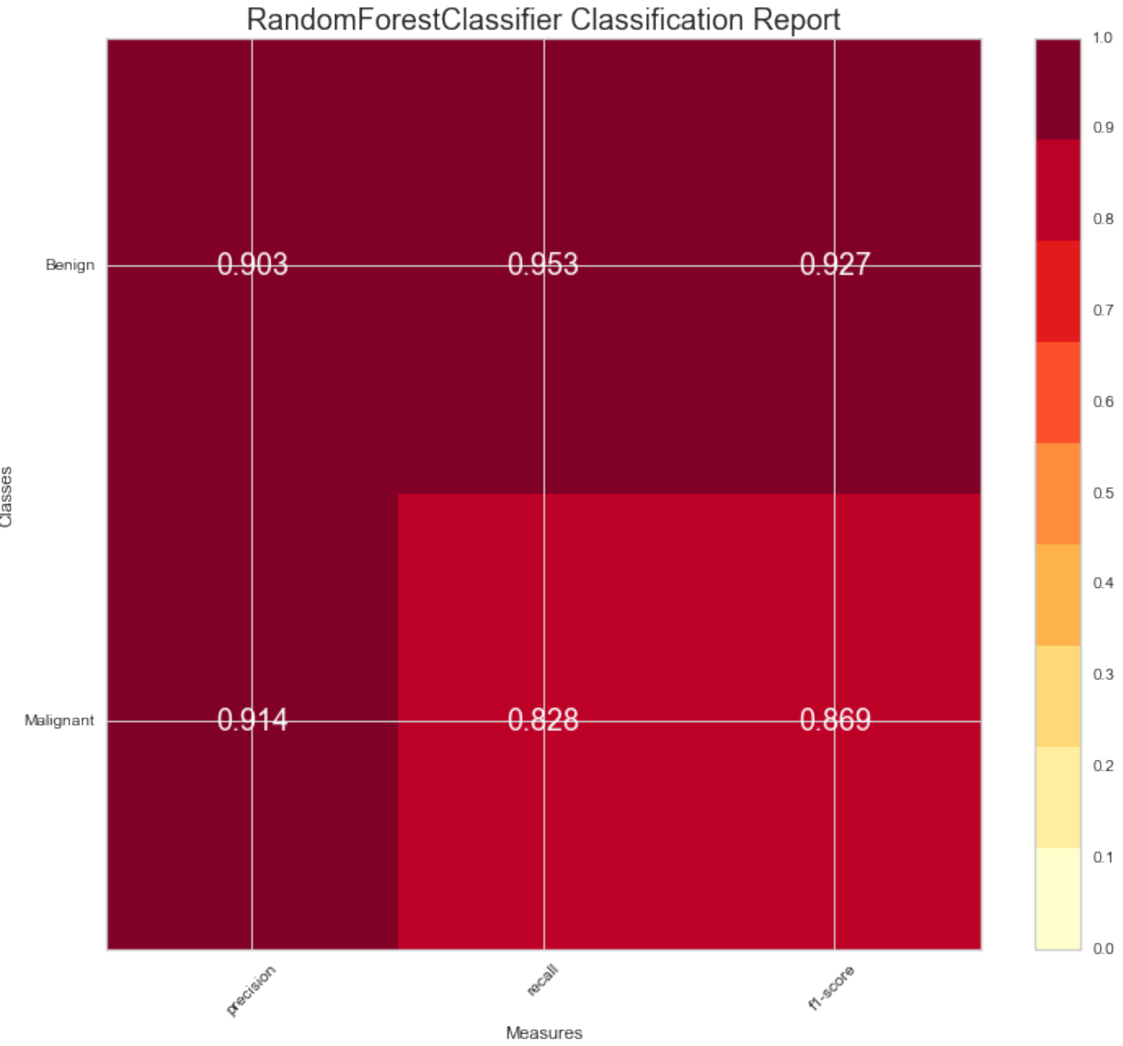


Visualizing a Classification Report





Visualizing a Classification Report




```
from yellowbrick.classifier import ClassificationReport

random_forest_class_report = ClassificationReport( random_forest_model,
                                classes=['Benign', 'Malignant'])
random_forest_class_report.fit(X_train, y_train)
random_forest_class_report.score(X_test, y_test)
random_forest_class_report.poof()
```

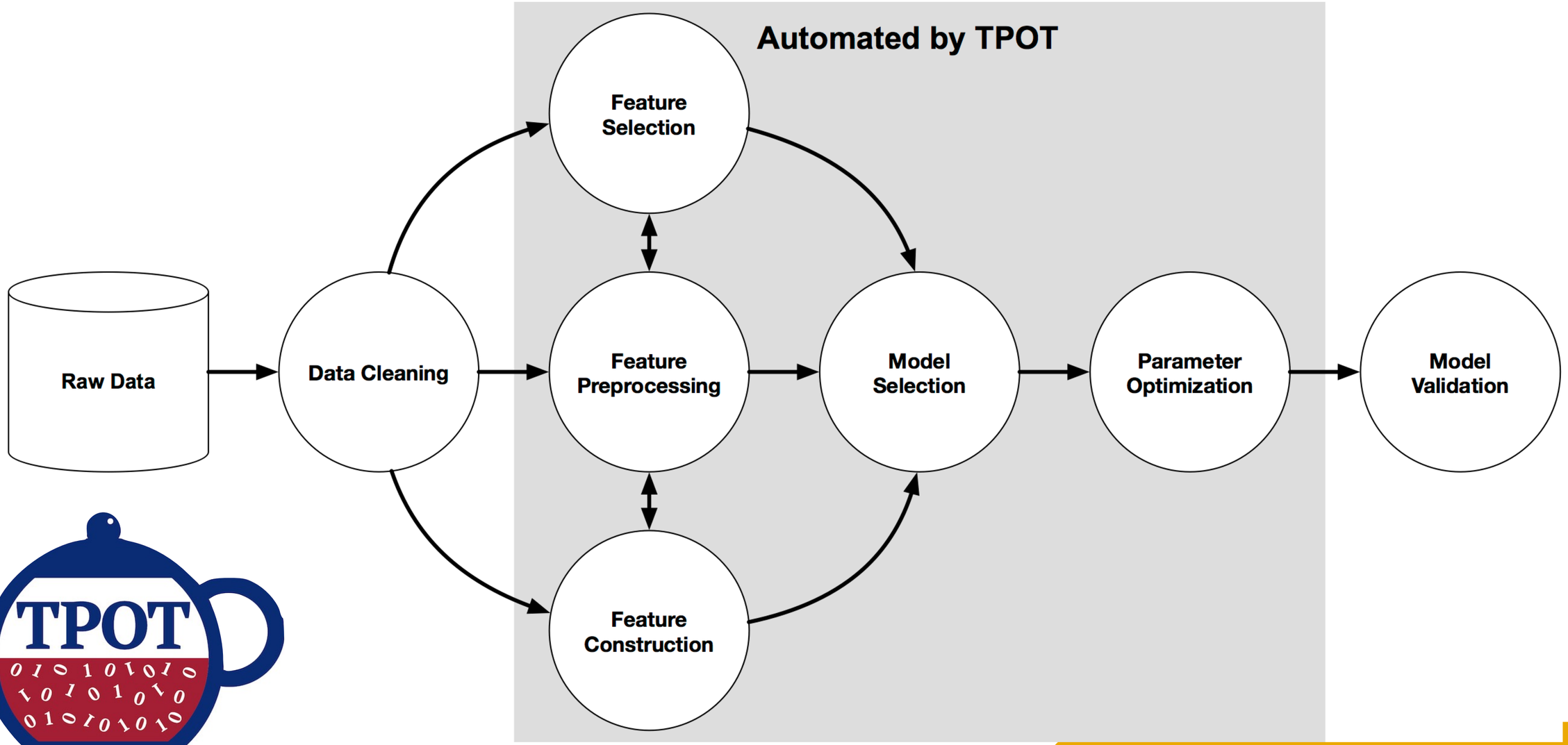
What can you automate?

Auto ML

What else can you automate?
It turns out, **almost everything**
with TPOT.

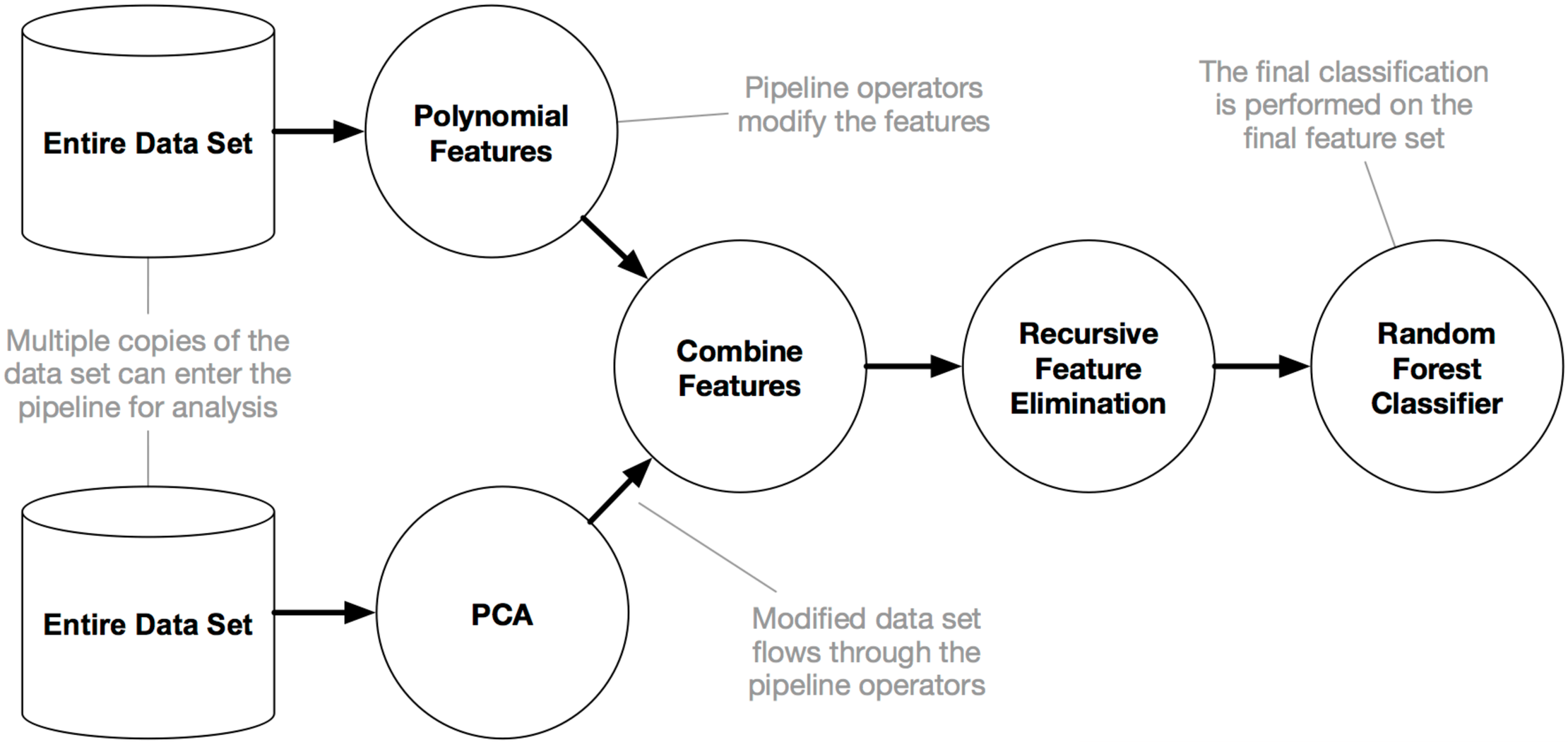
TPOT will automate the most tedious part of machine learning by intelligently exploring thousands of possible pipelines to find the best one for your data.

Auto ML



https://github.com/EpistasisLab/tpot

Auto ML



https://github.com/EpistasisLab/tpot

Auto ML

```
from tpot import TPOTClassifier

tpot = TPOTClassifier(generations=5, population_size=20, verbosity=2)
tpot.fit(features_train, target_train)
print(tpot.score(features_test, target_test))
tpot.export('tpot_pipeline.py')
```

https://github.com/EpistasisLab/tpot

Auto ML

Docs available here: http://epistasislab.github.io/tpot/

Questions?

Shameless Plug...

http://bit.ly/cyberds-md

Thank You!

GTKCYBER.COM | 855-GTK-CYBER | info@gtkcyber.com