AI for Anomaly Detection

Prerequisites

Professional Data Science Background with Python

Basics of Deep Learning – Have trained a DNN

Agenda

- Introduction to Anomaly Detection
- Supervised Learning with XGBoost
- Break
- Unsupervised Learning with Autoencoders
- Unsupervised Learning with GANs
- Assessment: Apply one technique to a new dataset

Introduction to Anomaly Detection

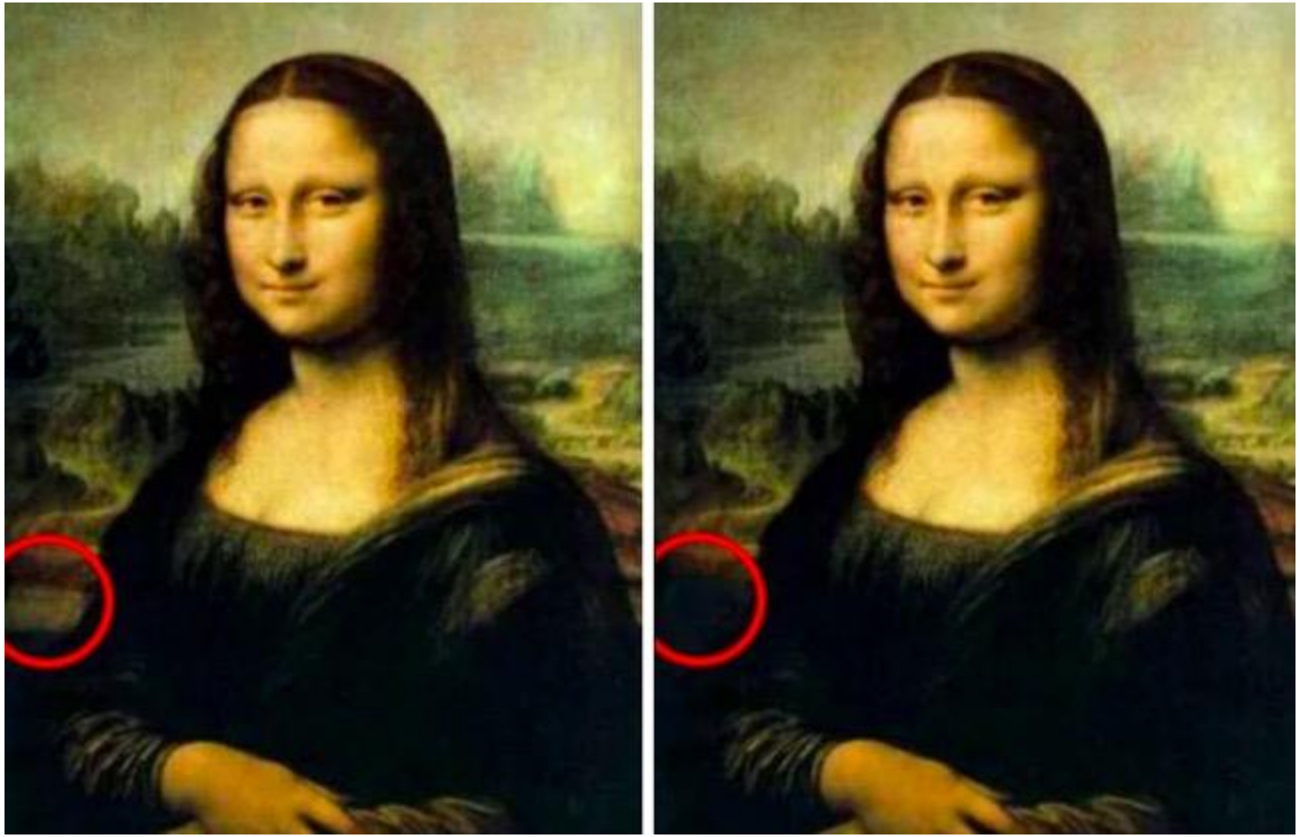
WHAT IS AN ANOMALY?

A ***data point*** which differs significantly from other ***data points***

- An observation that is likely generated by a different mechanism
- Finding anomalies can be useful in telecom/sp networks, cyber security, finance, industry, IOT, healthcare, autonomous driving, video surveillance, robotics.
- Many other problems can be framed as anomaly detection: customer retention, targeted advertising.

SPOT THE ANOMALY

SPOT THE ANOMALY



EXERCISE

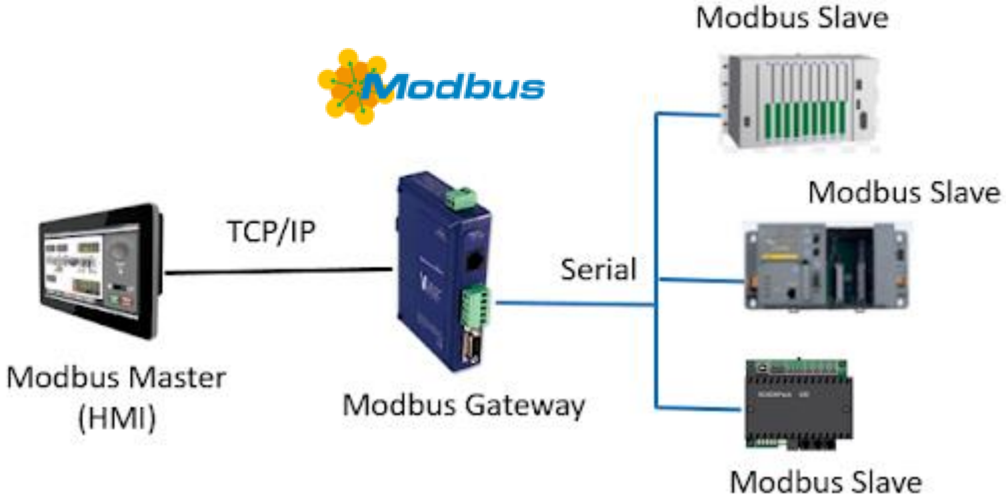
- What are some of the scenarios that produce anomalies in your organization/domain?
- What data sources might affect or record those anomalous activities?
- What kind of data analytics techniques could be applied or have been applied to detect those events?

Why is Anomaly Detection Important?

Case Study

Programmable Logic Controllers (PLCs)

Supervisory control and data acquisition (**SCADA**)



The Stuxnet Worm

Case Study

- A 500-kilobyte malicious computer worm that targets SCADA systems.
- Spread:
  - Through infected removable drives such as USB flash drives.
- Operation:
  - Analyzed and targeted Windows networks and computer systems.
  - Compromised the Step7 software, the worm gained access to 45 S7 to the PLCs.
  - Virus modified project communication configurations for the PLC’s Ethernet ports
- Result:
  - Infected over 100,000 computers & 22 Manufacturing sites
  - Appears to have impacted Natanz nuclear facility destroying 984 uranium enriching centrifuges.

DATASET

At a glance!

Name	KDD99 Intrusion Detection Dataset Publicly available at http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
Size	743 Mb
No. of Features	Numeric = 22 ; Categorical = 9
No. of Rows	18 Million
No. of Classes	23 (Including the Normal category)
Variable Types	Numeric & Categorical
Goal	**Detect Anomalies by studying Network Packet logs**

DATASET

Basic Features	Content Features	Traffic Features
duration	hot	count
protocol_type	num_failed_logins	serror_rate
service	logged_in	rerror_rate
src_bytes	num_compromised	same_srv_rate
dst_bytes	root_shell	diff_srv_rate
flag	su_attempted	srv_count
land	num_root	srv_serror_rate
wrong_fragment	num_file_creations	srv_rerror_rate
urgent	num_shells	srv_diff_host_rate
	num_access_files	
	num_outbound_cm	
	is_hot_login	
	is_guest_login	

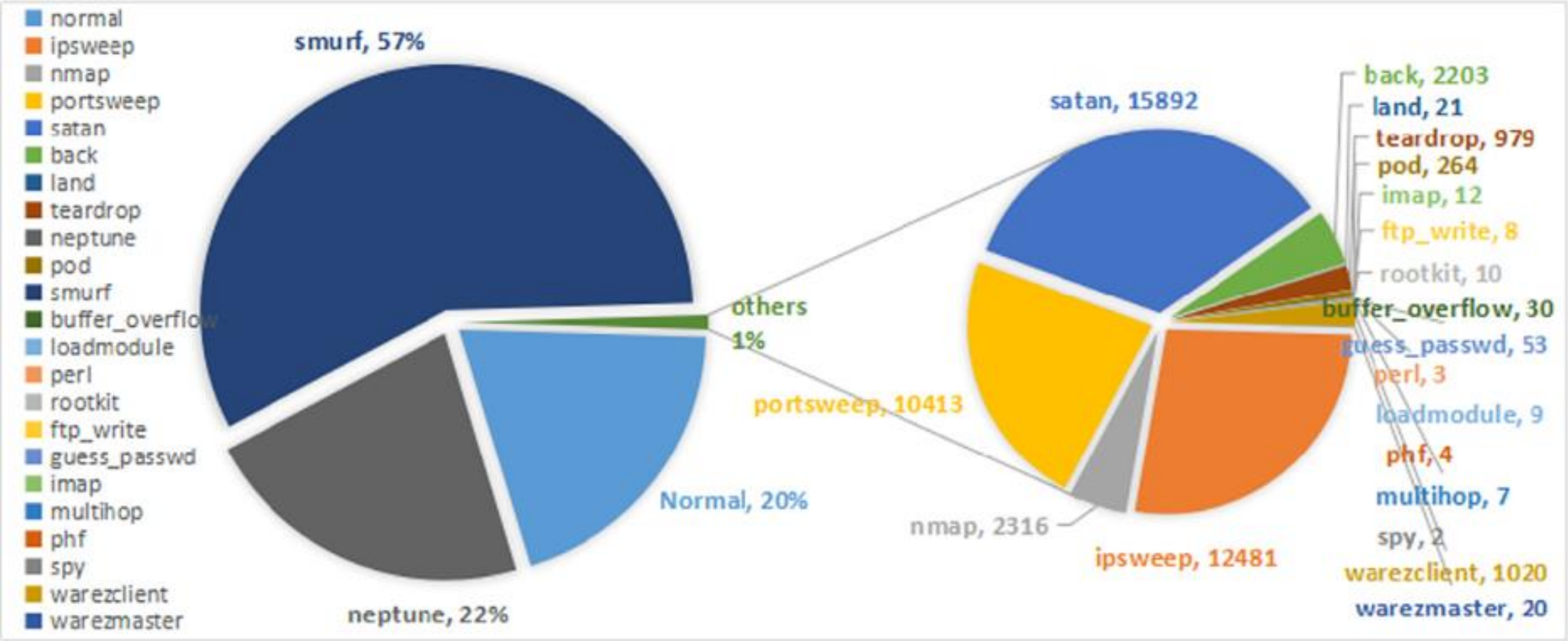
Numerical

Categorical

Detailed Description @
https://kdd.ics.uci.edu/databases/kddcup99/task.html

DATASET

Visualization by class



Handling Time Series Data

For Classification

Time	Feature 1	Feature 2	Feature 3
00:00:00	*Val_1*	*Val_2*	*Val_3*
00:00:01	*Val_4*	*Val_5*	*Val_6*
00:00:02	*Val_7*	*Val_8*	*Val_9*
00:00:03	*Val_10*	*Val_11*	*Val_12*

Averaging Features

Duration	Feature 1	Feature 2	Feature 3
1	*Avg(Val_1,Val_4)*	*Avg(Val_2,Val_5)*	*Avg(Val_3,Val_6)*
1	*Avg(Val_7,Val_10)*	*Avg(Val_8,Val_11)*	*Avg(Val_9,Val_12)*

Sampling Features

Duration	Feature 1	Feature 2	Feature 3
1	*Val_4*	*Val_5*	*Val_6*
1	*Val_10*	*Val_11*	*Val_12*

IN THE NEWS

Telecom

Operators beware: DDoS attacks—large and small—keep increasing

by Brian Santo | Jun 6, 2017 12:19pm

Telecoms industry and DNS attacks: attacked the most, slowest to fix

Networks are a prized target for hackers, as each attack costs £460,000 on average to remediate

https://www.information-age.com/telecoms-industry-dns-attacks-attacked-slowest-fix-123469037/

Telecom operators are not properly prepared for cyber-attacks: A10 Networks

Mobile network operators are not properly prepared for cyber attacks, and the core of 3G and 4G networks is generally not protected.

ETTelecom | Updated: January 15, 2018, 13:41 IST

https://telecom.economictimes.indiatimes.com/news/telecom-operators-are-not-properly-prepared-for-cyber-attacks-a10-networks/62504221

Hackers Are Tapping Into Mobile Networks' Backbone, New Research Shows

Parmy Olson Forbes Staff

AI, robotics and the digital transformation of European business.

https://www.forbes.com/sites/parmyolson/2015/10/14/hackers-mobile-network-backbone-ss7/#59d777f85142

Hack Attack: Sony Confirms PlayStation Network Outage Caused By 'External Intrusion'

Rip Empson @rpemp / 8 years ago

Comment

https://techcrunch.com/2011/04/23/hack-attack-sony-confirms-playstation-network-outage-caused-by-external-intrusion/

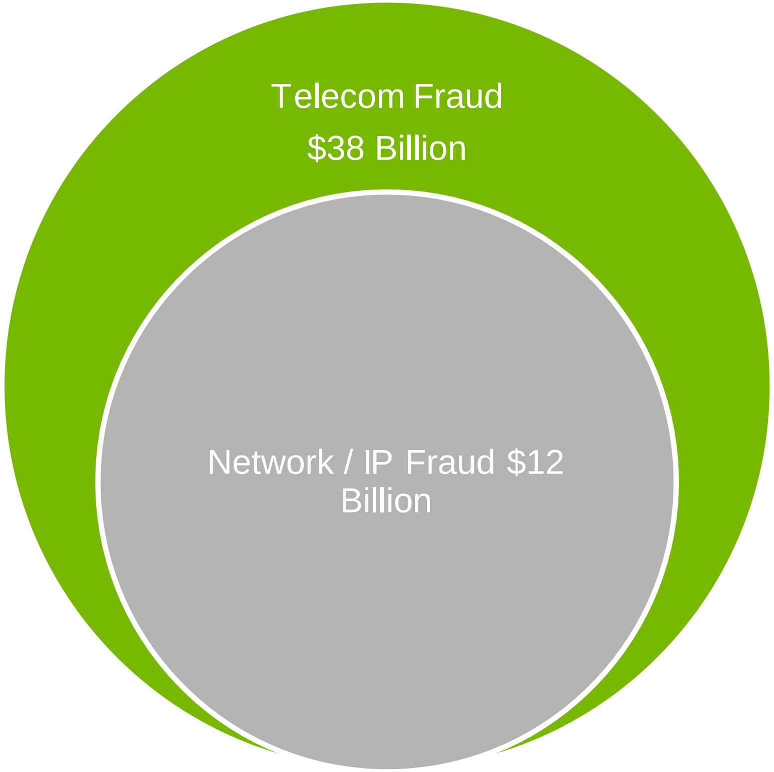
ANDY GREENBERG SECURITY 04.16.18 07:52 PM

THE WHITE HOUSE WARNS ON RUSSIAN ROUTER HACKING, BUT MUDDLES THE MESSAGE

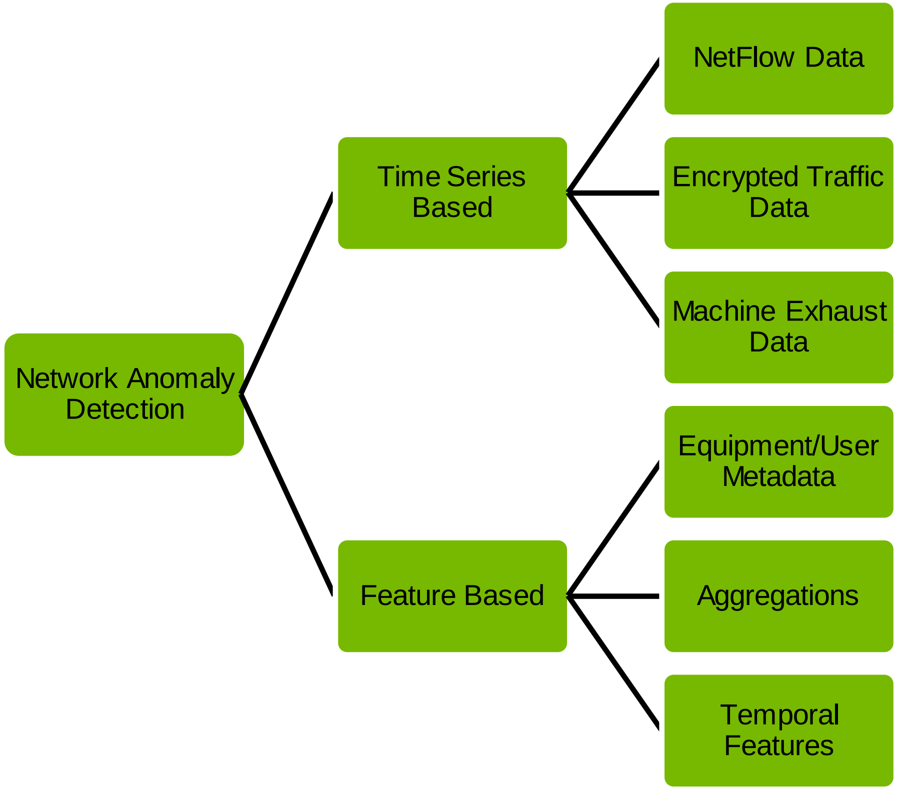
https://www.wired.com/story/white-house-warns-russian-router-hacking-muddles-message/

ANOMALY DETECTION IN NETWORKS

Why do we need it in Telecom ?



What sort of data can we leverage?



DETECTION METHODS IN THIS COURSE

Anomaly Detection

Supervised (When you have Labels)	Unsupervised (When you don't have labels for your data)	
XGBoost	Autoencoders	Generative Adversarial Networks
RAPIDS		

GPU ACCELERATED XGBOOST

XGBOOST

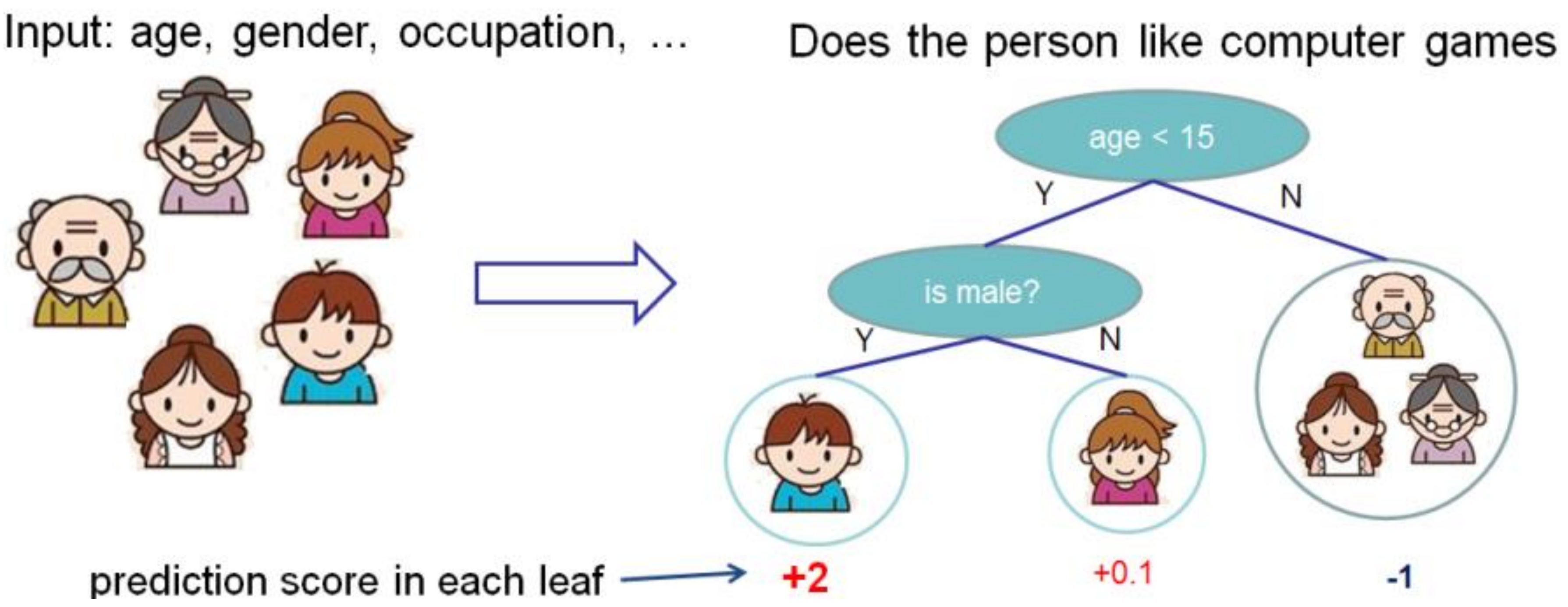
Definition

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

It is a powerful tool for solving classification and regression problems in a supervised learning setting.

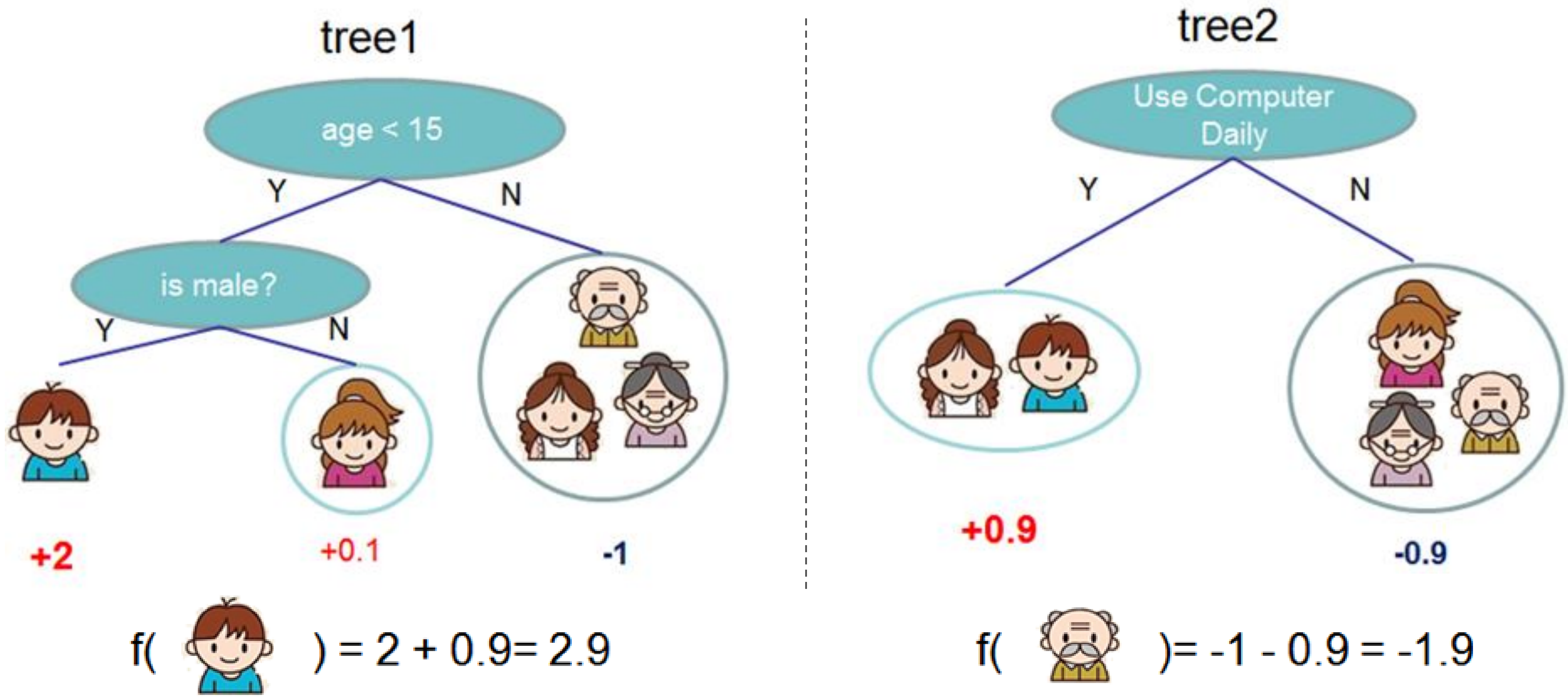
PREDICT: WHO ENJOYS COMPUTER GAMES

Example of Decision Tree



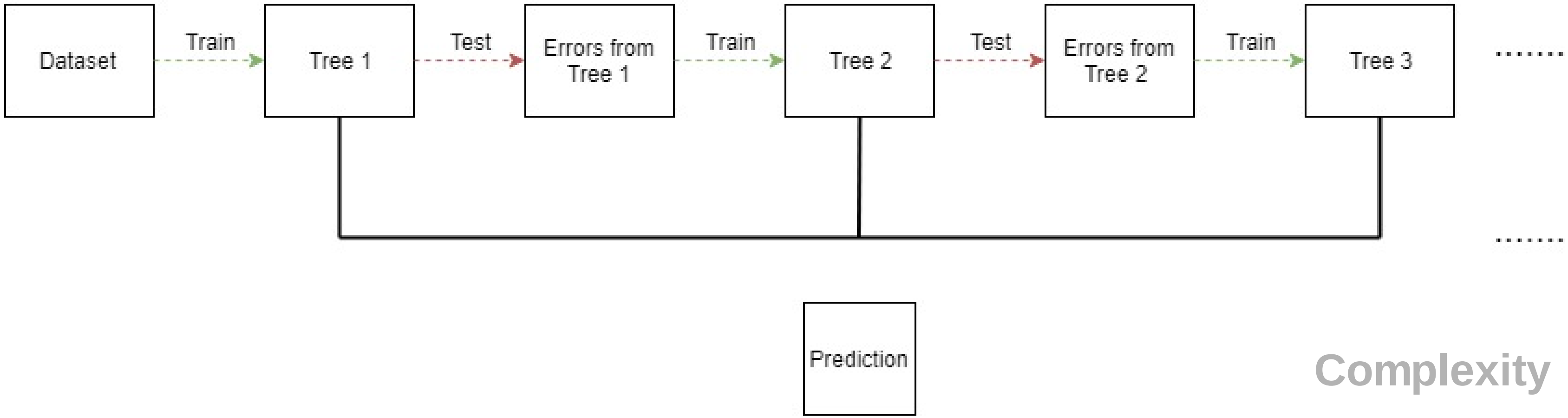
Source: https://goo.gl/eTxVtA

ENSEMBLED DECISION TREES



Source: https://goo.gl/eTxVtA

GRADIENT BOOSTED TREES FOR STRONGER PREDICTIONS



Build trees one at a time, where each new tree helps to correct errors made by previously trained tree.

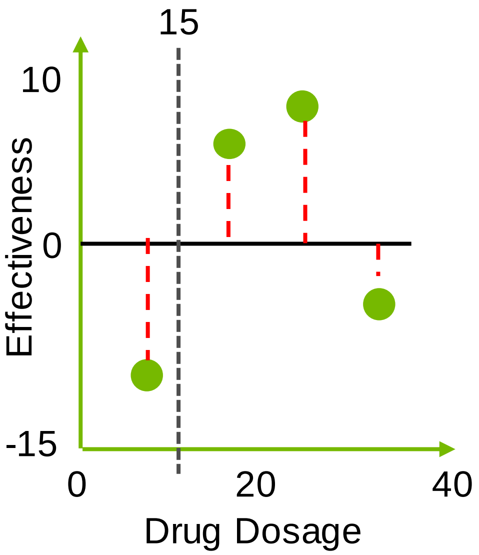
Complexity of the Trees

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Training Loss

XgBoost

Intuitive Example for Tree Construction



Step 1: Start as a single leaf
Input all residuals

-10.5, 6.5, 7.5, -7.5

Step 2: Calculate similarity score
For all residuals

$$\frac{\text{Sum of residuals squared}}{\text{No. of residuals} + \text{Regularization}}$$

Set Threshold @ Arbitrary Drug Dosage 15

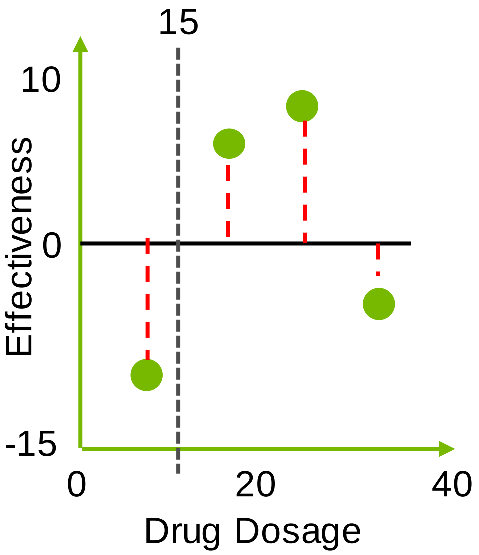
-10.5, 6.5, 7.5, -7.5

S0 = 4
Setting Reg = 0

https://www.youtube.com/watch?v=OtD8wVaFm6E

XgBoost

Intuitive Example for Tree Construction



Step 3: Calculate Similarity Score

For Left and Right Sub Trees

-10.5, 6.5, 7.5, -7.5

S10 = 110.25

Setting Reg = 0

-10.5

6.5, 7.5, -7.5

S11 = 14.08

Setting Reg = 0

Step 4: Calculate Gain

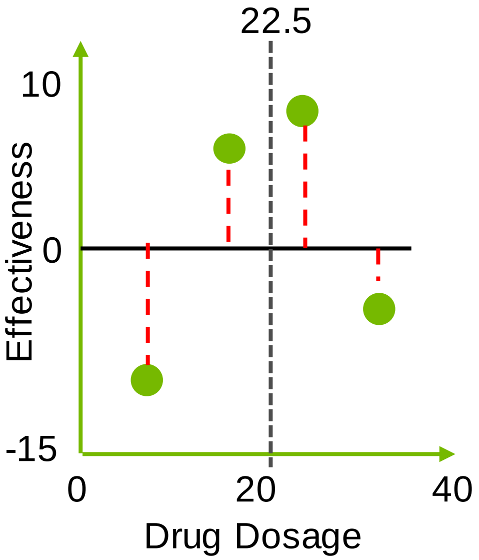
Lower Leaves cluster similar residuals than root ?

Gain = Left Similarity + Right Similarity – Root Similarity

G1 = 120.33

XgBoost

Intuitive Example for Tree Construction



Step 5: Calculate Similarity Score
For Left and Right Sub Trees

Dosage < 22.5

S10 = 8
Setting Reg = 0

-10.5 , 6.5

7.5, -7.5

S11 = 0
Setting Reg = 0

Step 6: Calculate Gain
Lower Leaves cluster similar residuals than root ?

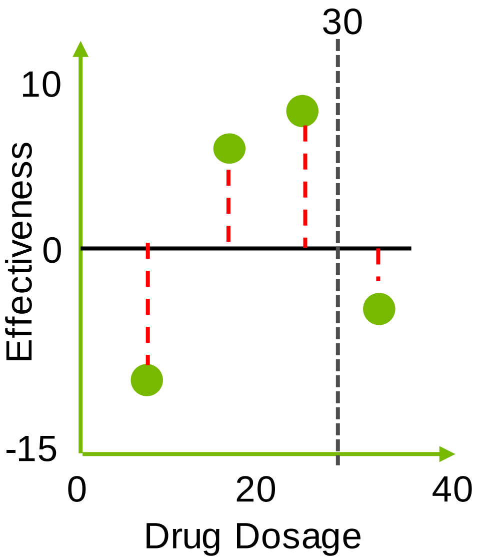
Gain = Left Similarity + Right Similarity – Root Similarity

G2 = 4

Since G2 = 4 < G1 = 120.33
Tree 1 had better split

XgBoost

Intuitive Example for Tree Construction



Step 7: Calculate Similarity Score
For Left and Right Sub Trees

Dosage < 30

S10 = 4.08
Setting Reg = 0

-10.5 , 6.5, 7.5

-7.5

S11 = 56.25
Setting Reg = 0

Step 6: Calculate Gain
Lower Leaves cluster similar residuals than root ?

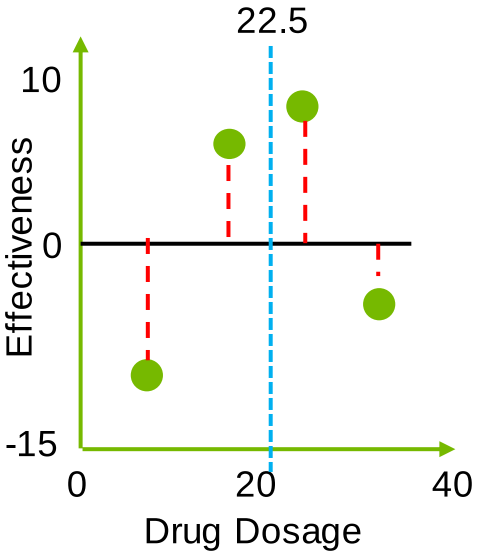
Gain = Left Similarity + Right Similarity – Root Similarity

G3 = 56.33

Since G3 = 56.33 < G1 = 120.33
Tree 1 had better split

XgBoost

Intuitive Example for Tree Construction



Dosage <15

S10 = 110.25
Setting Reg = 0

-10.5

Dosage < 22.5

S11 = 14.08
Setting Reg = 0

Step 8: Calculate Similarity Score
For Left and Right Sub Trees (2^{nd} Level)

S21 = 42.25
Setting Reg = 0

6.5

7.5, -7.5

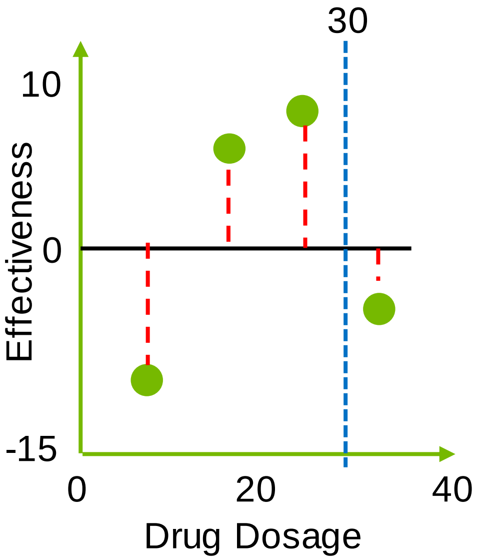
S21 = 0
Setting Reg = 0

Step 9: Calculate Gain
Lower Leaves cluster similar residuals than root ?

G12 = 42.25 – 14.0 = 28.17

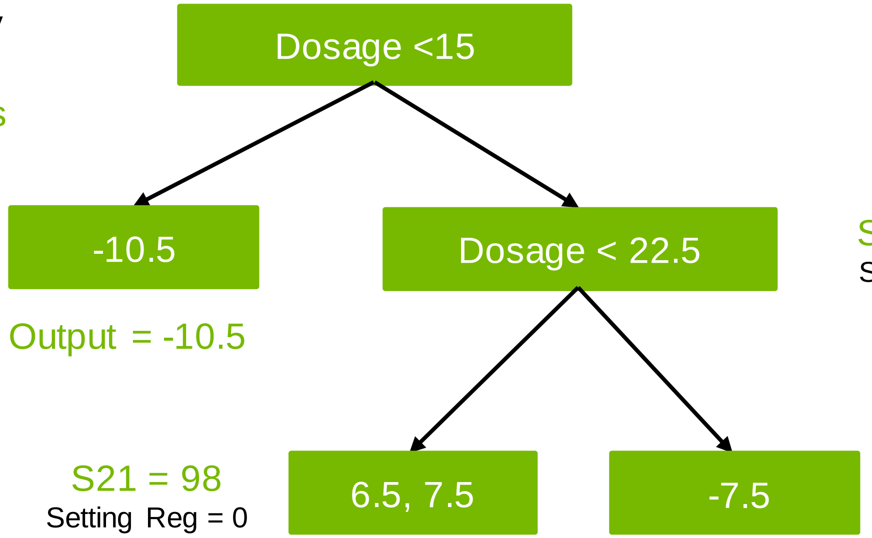
XgBoost

Intuitive Example for Tree Construction



Step 10: Calculate Similarity Score

For Left and Right Sub Trees



S10 = 110.25
Setting Reg = 0

Output = -10.5

S11 = 14.08
Setting Reg = 0

Step 11: Calculate Gain

Lower Leaves cluster similar residuals than root ?

S21 = 98
Setting Reg = 0

S21 = 56.25
Setting Reg = 0

G21 = 140.17 > G12

Output = 7

Output = -7.5

Step 12:
Calculate Output Value = $\dfrac{\text{Sum of residuals squared}}{\text{No. of residuals} + \text{Regularization}}$

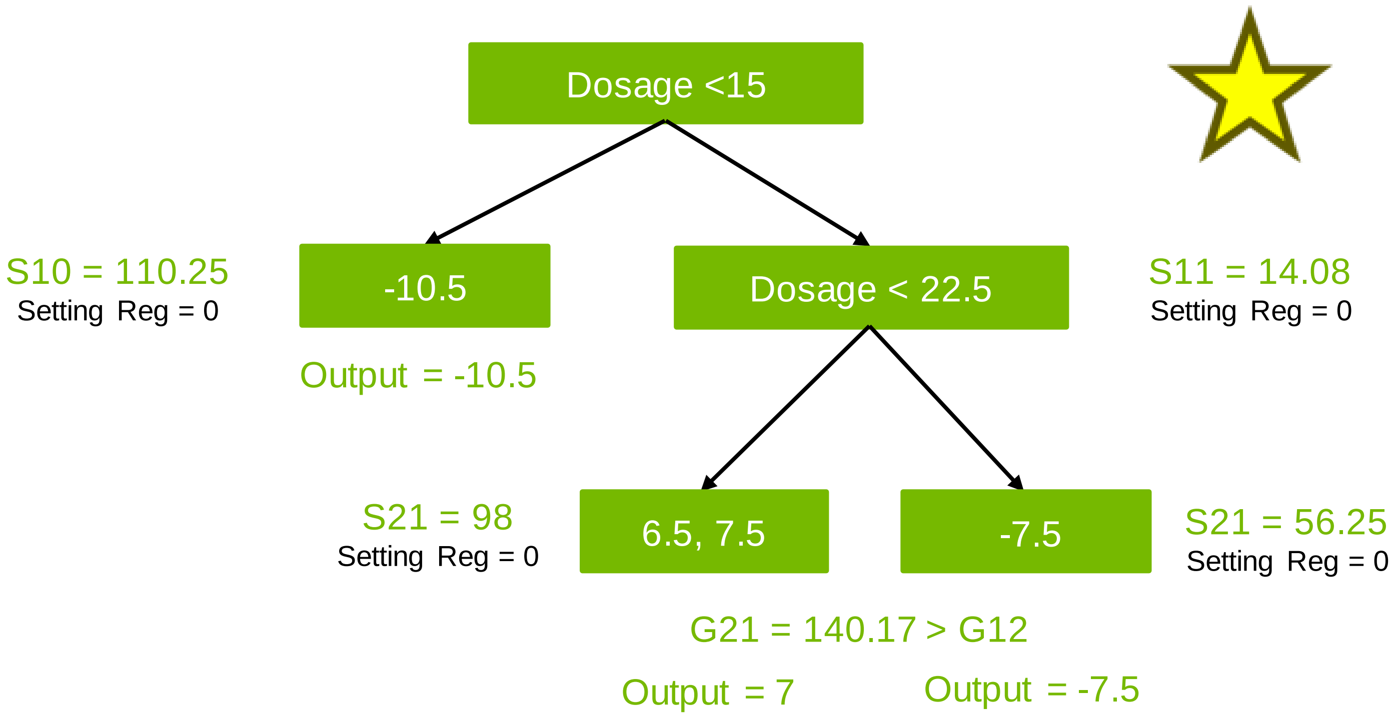
XgBoost

Intuitive Example for Tree Construction

Re - Calculate Residuals :
Assuming Gradient Multiplier = 0.3

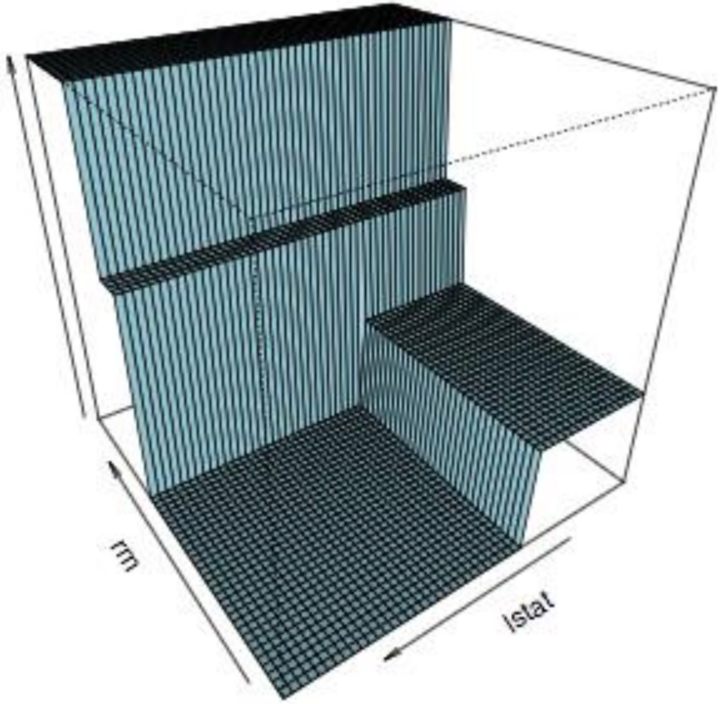
- R1 : 0.5 + 0.3 (-10.5) = - 2.65
- R2 : 0.5 + 0.3(7) = 2.6
- R3 : 0.5 + 0.3(7) = 2.6
- R4 : 0.5 + 0.3(-7.5) = -1.75

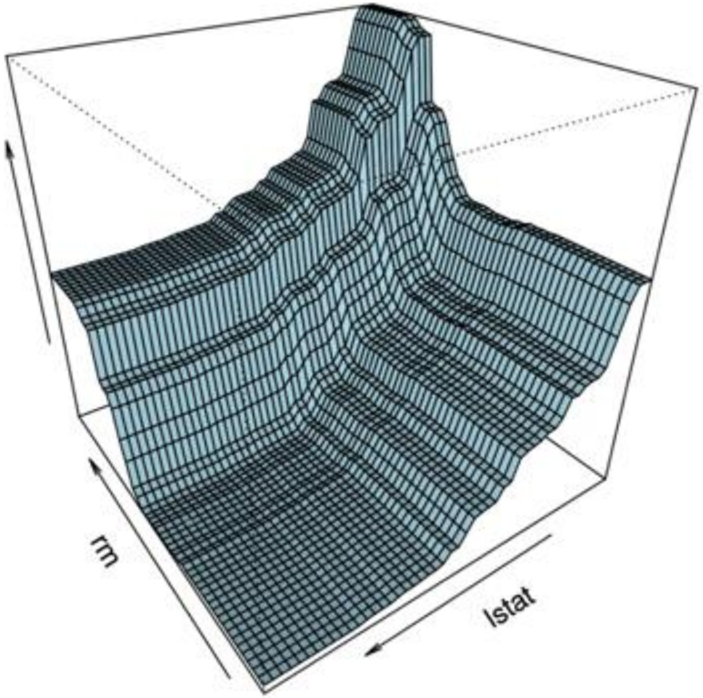
Step 13:
Construct new tree with updated Residuals



TRAINED MODELS VISUALIZATION

Single Decision Tree vs Ensembled Decision Trees





Models fit to the *Boston Housing* Dataset

Source: https://goo.gl/GWNdEm

XgBoost

Building up from a Decision Tree

Optimized version of GBT incorporating parallelism, tree pruning and regularization.

Utilize Gradient Descent to minimize errors in the sequentially built trees.

Trees built sequentially minimizing errors from previous trees and weighing better performing ones more.

Utilize random subsets of a dataset to build multiple decision trees

Ensemble of multiple decision trees to arrive at decision through majority voting

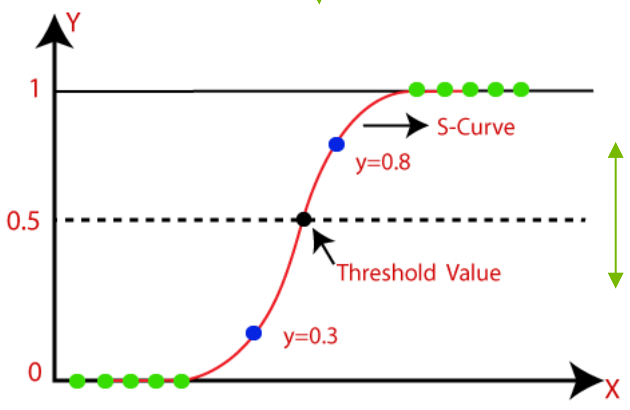
Tree based algorithm that outputs decisions based on certain conditions.

WHY XGBOOST?

ROC CURVE

Construction

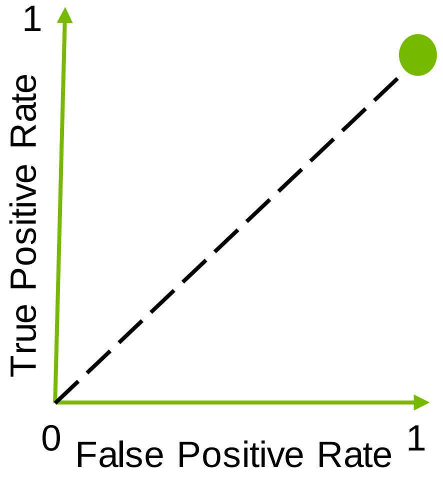
Move
Threshold



		Actual	
		Anomaly	Not Anomaly
Predicted	Anomaly	TP	FP
	Not Anomaly	FN	TN

True Positive Rate
TP / (TP+FN) = Sensitivity

False Positive Rate
FP / (FP +TN)

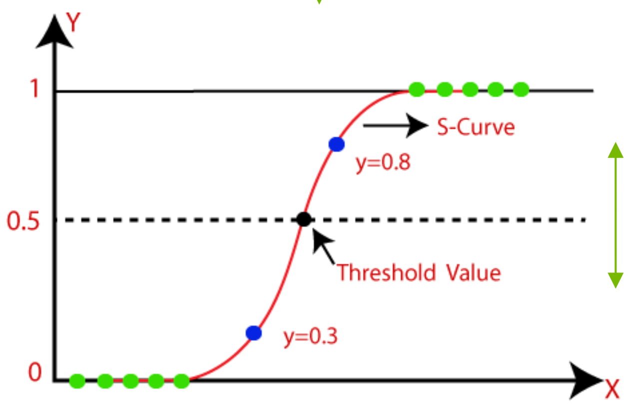


https://www.youtube.com/watch?v=4jRBRDbJemM

ROC CURVE

Construction

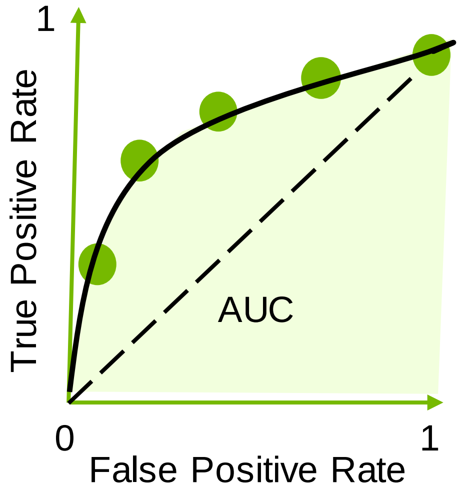
Move Threshold



		Actual	
		Anomaly	Not Anomaly
Predicted	Anomaly	TP	FP
	Not Anomaly	FN	TN

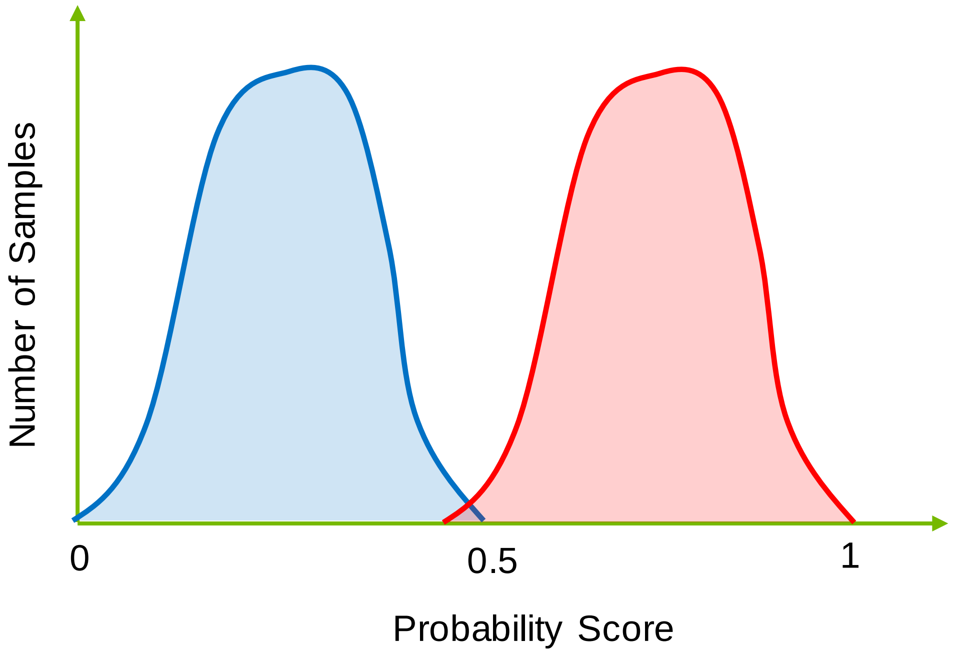
True Positive Rate
TP / (TP+FN) = Sensitivity

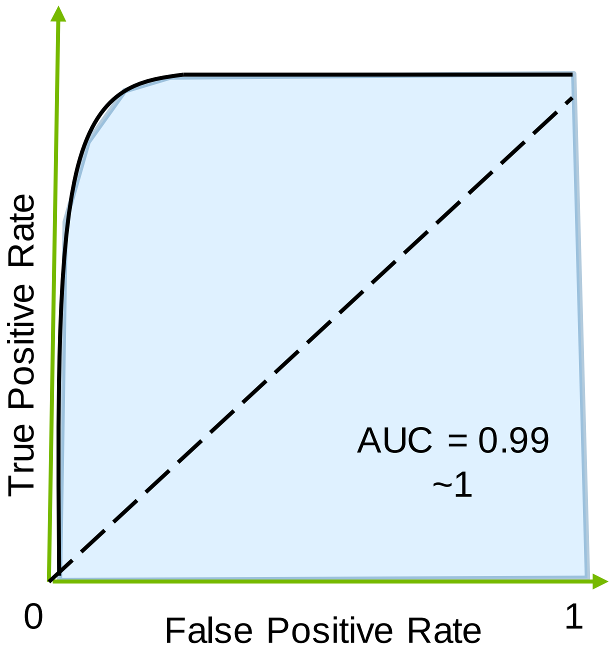
False Positive Rate
FP / (FP +TN)



ROC CURVE

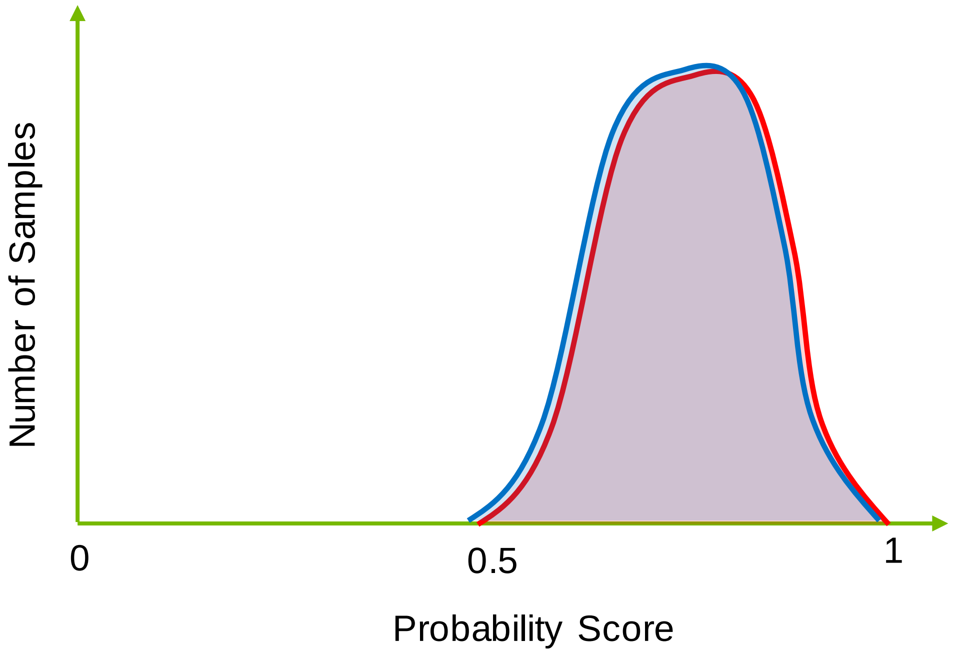
Interpretation

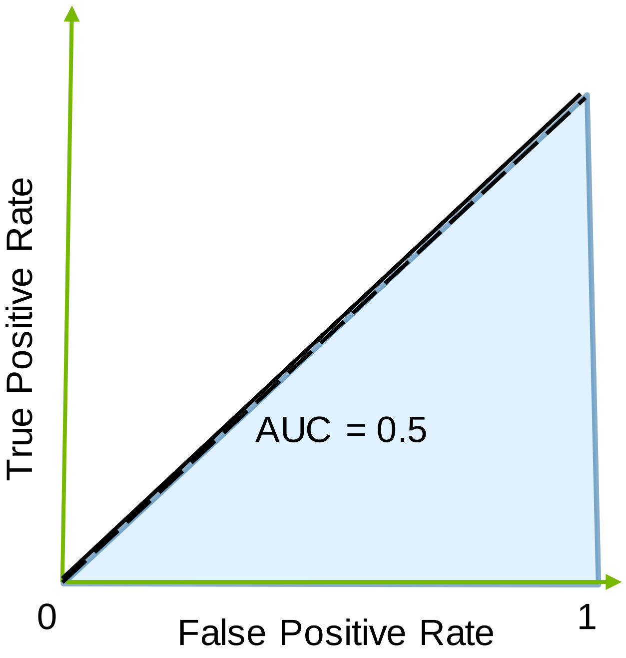




ROC CURVE

Interpretation

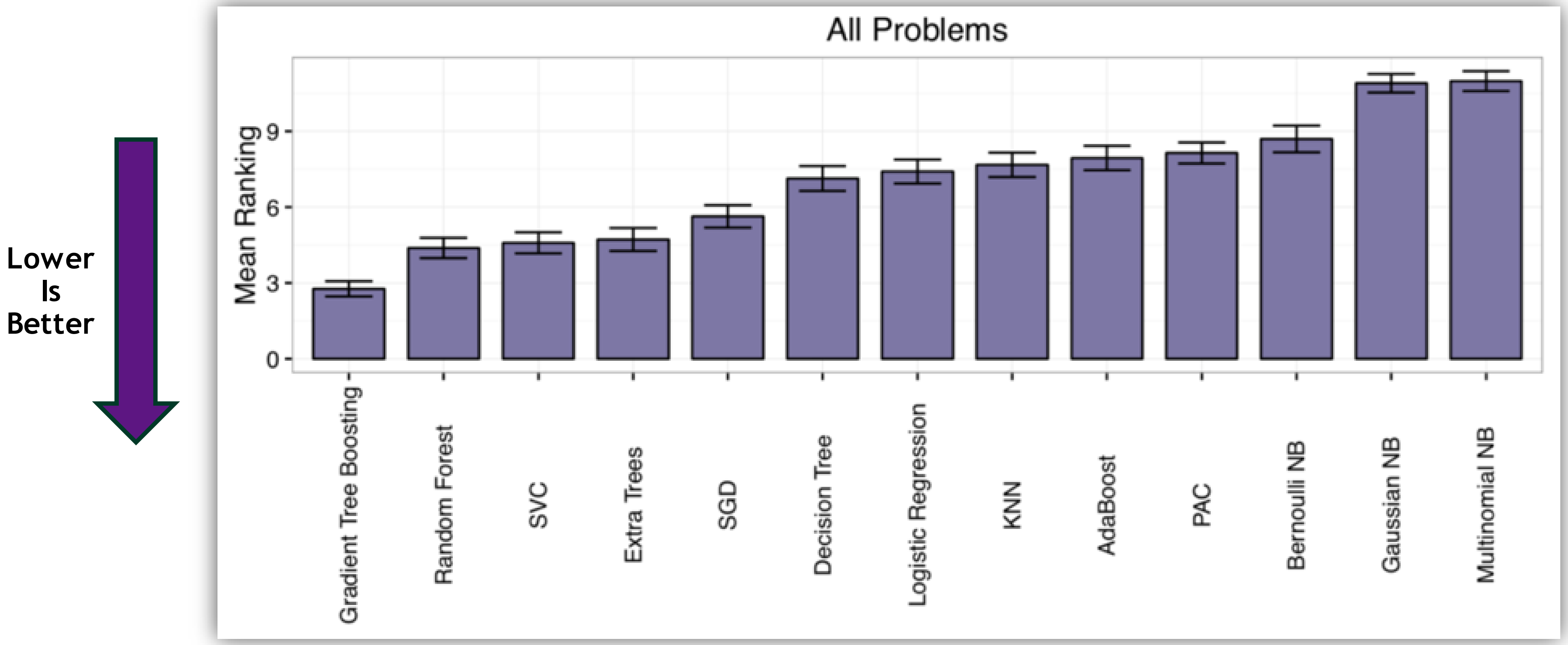




Better the separation between the classes = Better Model / Classifier

WHICH ML ALGORITHM PERFORMED BEST

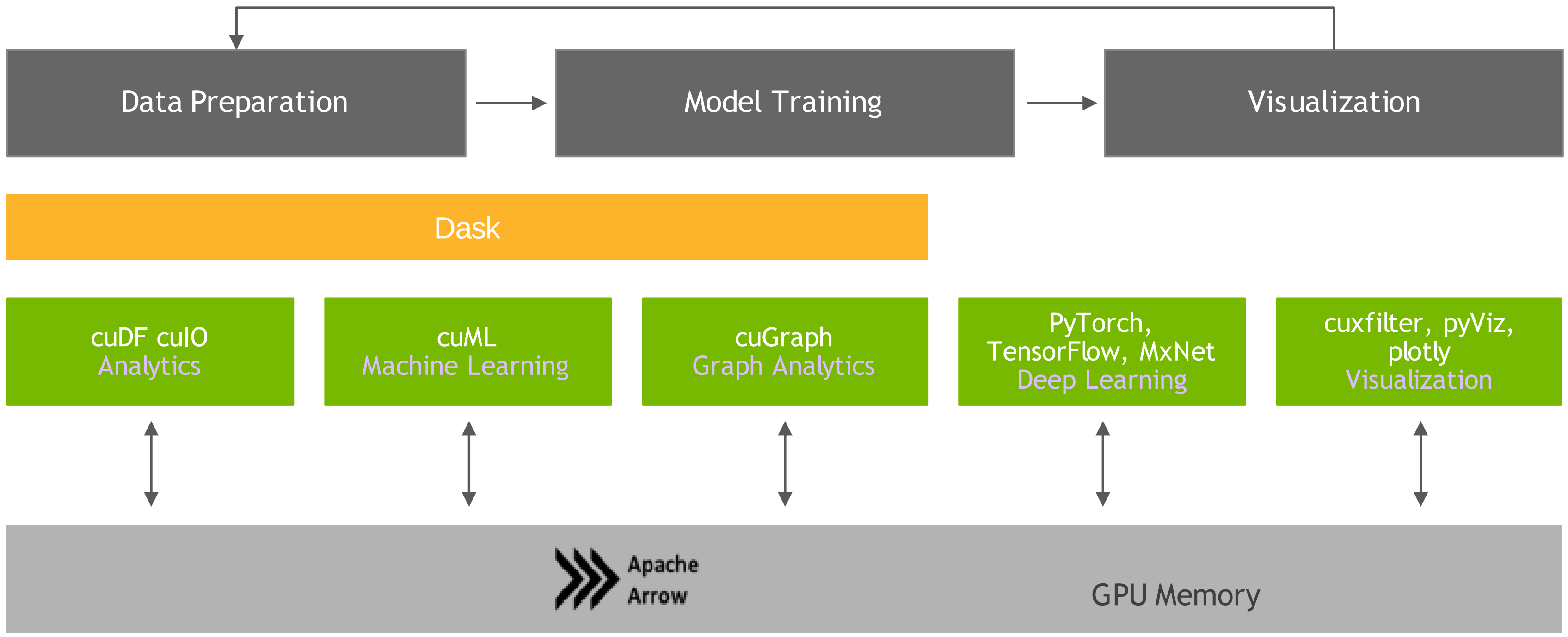
Average rank across 165 ML datasets



Source: https://goo.gl/R8Y8Pp

RAPIDS

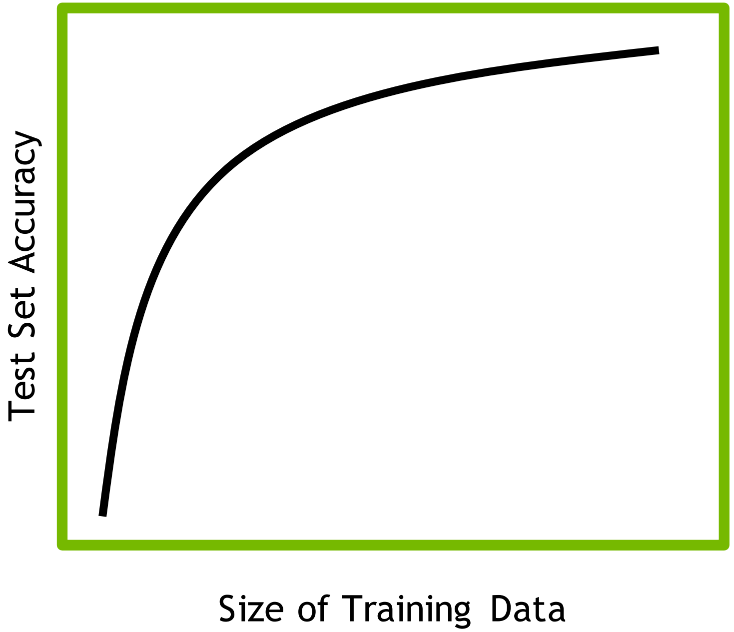
End-to-End Accelerated GPU Data Science



WHY RAPIDS + XGBOOST?

TIME TO TRAIN

Rapid Data Science



Model Selection and Hyper-Parameter Tuning

```
best_model = init_model

for (m,h) in zip(models, hyperparams):

    my_model = train(m,h)

    if acc(my_model) > acc(best_model):

        best_model = my_model
```

RAPIDS WITH XGBOOST

Multi-GPU, Multi-Node, Scalability

- XGBoost:
  - Algorithm tuned for eXtreme performance and high efficiency
  - Multi-GPU and Multi-Node Support
- RAPIDS:
  - End-to-end data science & analytics pipeline entirely on GPU
  - User-friendly Python interfaces
  - Relies on CUDA primitives, exposes parallelism and high-memory bandwidth
  - Benefits from DGX system designs (NVLINK, NVSWITCH, dense compute platform)
  - Dask integration for managing workers & data in distributed environments

Work through the first reflection

1.2 Dataset Modification

Notice that the dataset has more anomalies than normal data. Reflect for a moment about the implications of having more anomalies might be. Reflect either here in the notebook, on a piece of paper, or with a peer sitting next to you.

```
Reflection:
```

We'll come back to test your hypothesis shortly.

Section 3: Impact of Skewed Data

As we prepared our data, we pointed out that there were more anomalies than normal data and considered the implications of this dataset skew that doesn't match the real world. Take a moment now see how adjusting our dataset impacts performance.

In [2]:
```
def reduce_anomalies(df, pct_anomalies=.01):
    labels = df['label'].copy()
    is_anomaly = labels != 'normal.'
    num_normal = np.sum(~is_anomaly)
    num_anomalies = int(pct_anomalies * num_normal)
    all_anomalies = labels[labels != 'normal.']
    anomalies_to_keep = np.random.choice(all_anomalies.index, size=num_anomalies, replace=False)
    anomalous_data = df.iloc[anomalies_to_keep].copy()
    normal_data = df[~is_anomaly].copy()
    new_df = pd.concat([normal_data, anomalous_data], axis=0)
    return new_df
```

In [ ]:
```
df = reduce_anomalies(df)
```

Let's see what anomalies we have after the reduction.

In [ ]:
```
pd.DataFrame(df['label'].value_counts())
```

Return to [data preprocessing] and rerun cells to this point, comparing and contrasting performance. Again, reflect below, on paper, or with a peer. Reflect on *why* the reduction of anomalies had the impact that it did.

What was the impact of reducing anomalies in the dataset and why do you think that is?

```
Answer:
```

Multi-Class Classifier Challenge

In the field below, set up `dtrain`, `dtest`, `evals`, and `model` as exemplified when we trained our binary classifier.

Note: Multiclass labels are in y_train and y_test. Hint: Control F will help you find `dtrain`, `dtest`, `evals` and `model`.

You can see how adding multiple classes doesn't increase the complexity in training this type of model.

In [ ]:

```
%%time


dtrain = ##SEE BINARY CLASSIFIER FOR HINT##
dtest = ##SEE BINARY CLASSIFIER FOR HINT##
evals = ##SEE BINARY CLASSIFIER FOR HINT##
model = ##SEE BINARY CLASSIFIER FOR HINT##
```