#ODSC

@ODSC

Tutorial : How to Build High Performing Weighted XGBoost ML Model for Real Life Imbalance Dataset.

OPEN DATA SCIENCE CONFERENCE

London| September 19th- 22nd 2018

Alok Singh
IBM Center for Open Source Data & AI Technologies

About Presenter

Alok Singh

Alok Singh is a Principal Engineer at the IBM CODAIT (Center for Open-Source Data and AI Technologies). He has built and architected multiple analytical frameworks and implemented machine learning algorithms along with various data science use cases. His interest is in creating Big Data and scalable machine learning software and algorithms. He has also created many Data Science based applications.

- Interests: Big Data, Scalable Machine Learning Software, and Algorithms.

- Contact:

url- http://codait.org

email- singh_alok@hotmail.com

github -https://github.com/aloknsingh

Agenda

- IBM and Data Science
- Why this workshop?
- Tools and Technology
- Overview and flow of workshop
    - Scikit-learn based ML pipeline
    - Various Model evaluation techniques
    - Iterative Model training
- Hands on tutorial for building predictive models for imbalance datasets
    - Setup
    - Tutorial
- Q/A

IBM Commitments to Open Source AI and Data Science

Center for Open Source Data and AI Technologies

CODAIT aims to make AI solutions dramatically easier to create, deploy, and manage in the enterprise

Relaunch of the Spark Technology Center (STC) to reflect expanded mission

CODAIT

codait.org

codait (French)

= coder/coded

https://m.interglot.com/fr/en/codait

IBM CODE

Center for Open Source Data and AI Technologies

Code - Build and improve practical frameworks to enable more developers to realize immediate value

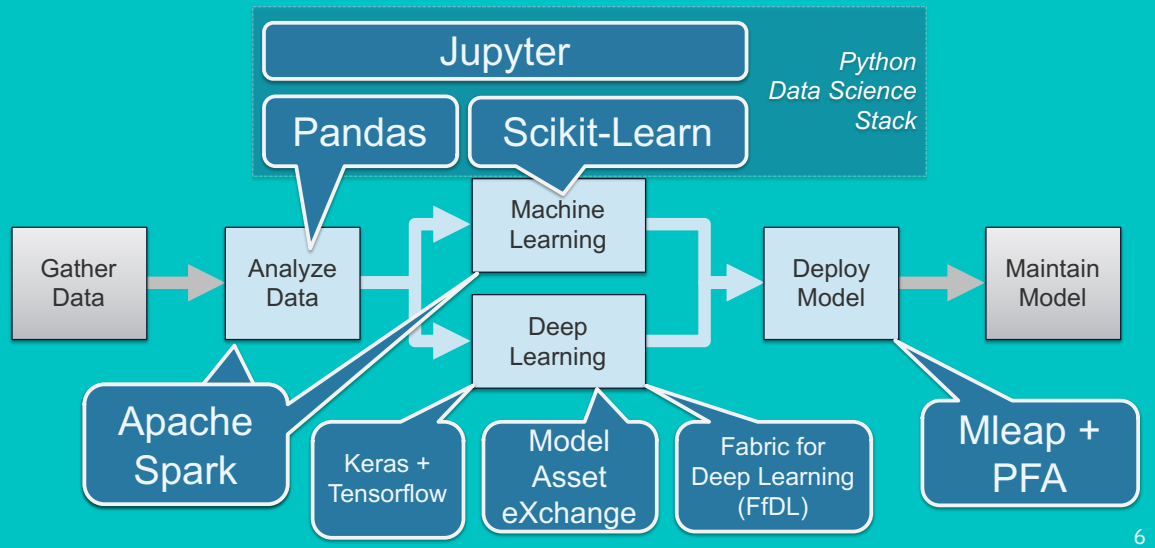
Content – Showcase solutions to complex and real world AI problems

Community – Bring developers and data scientists to engage with IBM

CODAIT

codait.org

Improving Enterprise AI lifecycle in Open Source



IBM Watson Studio Data Science Experience

ALL YOUR TOOLS IN ONE PLACE

IBM Watson Studio Data Science Experience is an environment that brings together everything that a Data Scientist needs. It includes the most popular Open Source tools and IBM unique value-add functionalities with community and social features, integrated as a first class citizen to make Data Scientists more successful.

IBM Data Science Experience

Click here to watch video

Why This Workshop for building Predictive Models for Imbalance Datasets?

Challenges in building ML model

- Non Representative data
- Insufficient data
- Poor quality data
- Imbalance data set
- Irrelevant features
- Over fitting the model on data
- Under fitting of the model on data
- Whether model will generalize or not

Imbalance Dataset: A Story …

We get real life labeled data sets from our clients and we are asked to build a classifier to predict if someone will buy our products.

We clean our data and build latest state of art ML model for it.

We run cross validation and feature engineering to select best model.

We evaluate model and it doesn’t perform good on test dataset but in theory it should perform good.

We explore more and finds that the number of labeled samples where someone buys the product is very less say 1 % and our classifier didn’t learn since it thought that those product that was bought was noise.

What we will do in this workshop …

- Data Set Description.
- Exploratory Analysis to understand the data.
- Use various preprocessing to clean and prepare the data.
- Use naive XGBoost to run the classification.
- Use cross validation to get the model.
- Plot, precision recall curve and ROC curve.
- We will then tune it and use weighted positive samples to improve classification performance.
- We will also talk about the following advanced techniques.
- Oversampling of majority class and Undersampling of minority class.
- SMOTE algorithms.

Tools and Technologies

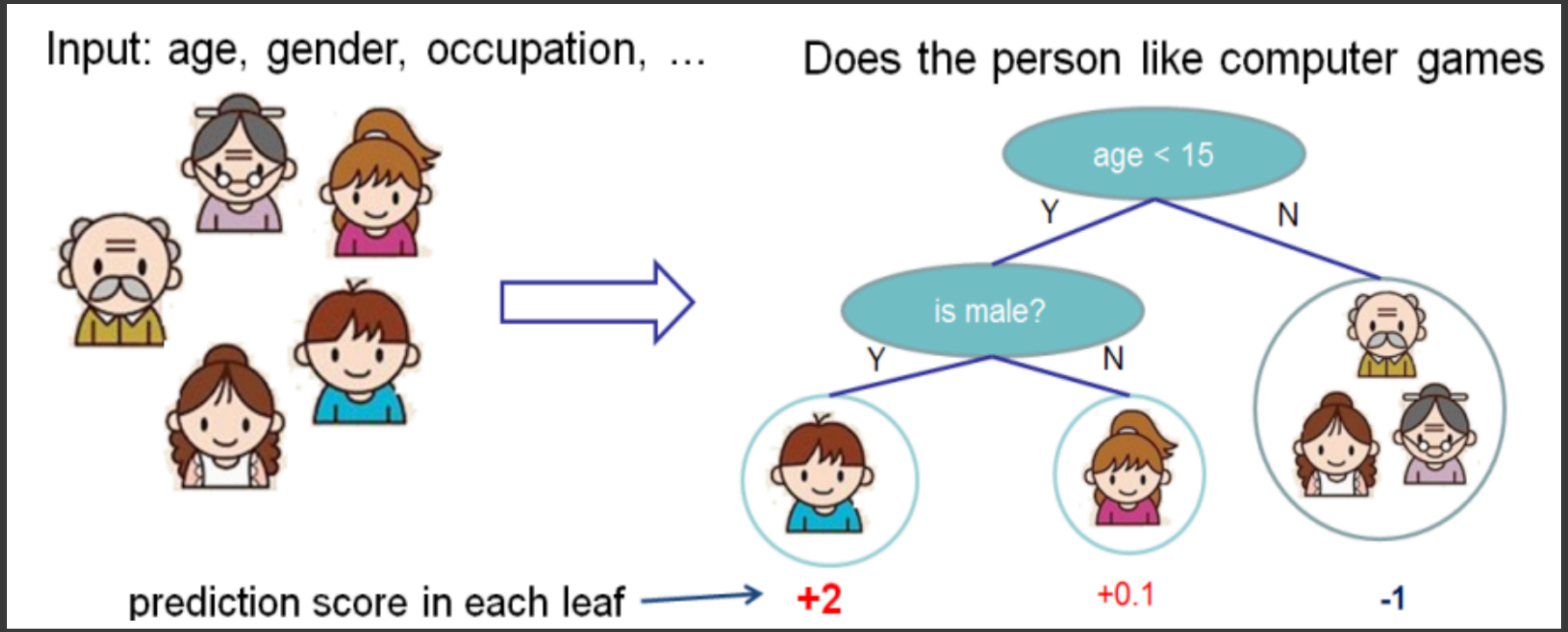
Scikit-Learn

- Library of ML models and utilities
- Has many out of box ML models and utilities that works out of box
- Python is now very popular among data-scientists.
- XGBoost allows one to tune various parameters.
- Has pipeline, estimators and transformers concept that allows one to build big ML apps
- Well Documented and a lot of examples and support group.

Why and What of XGBoost.

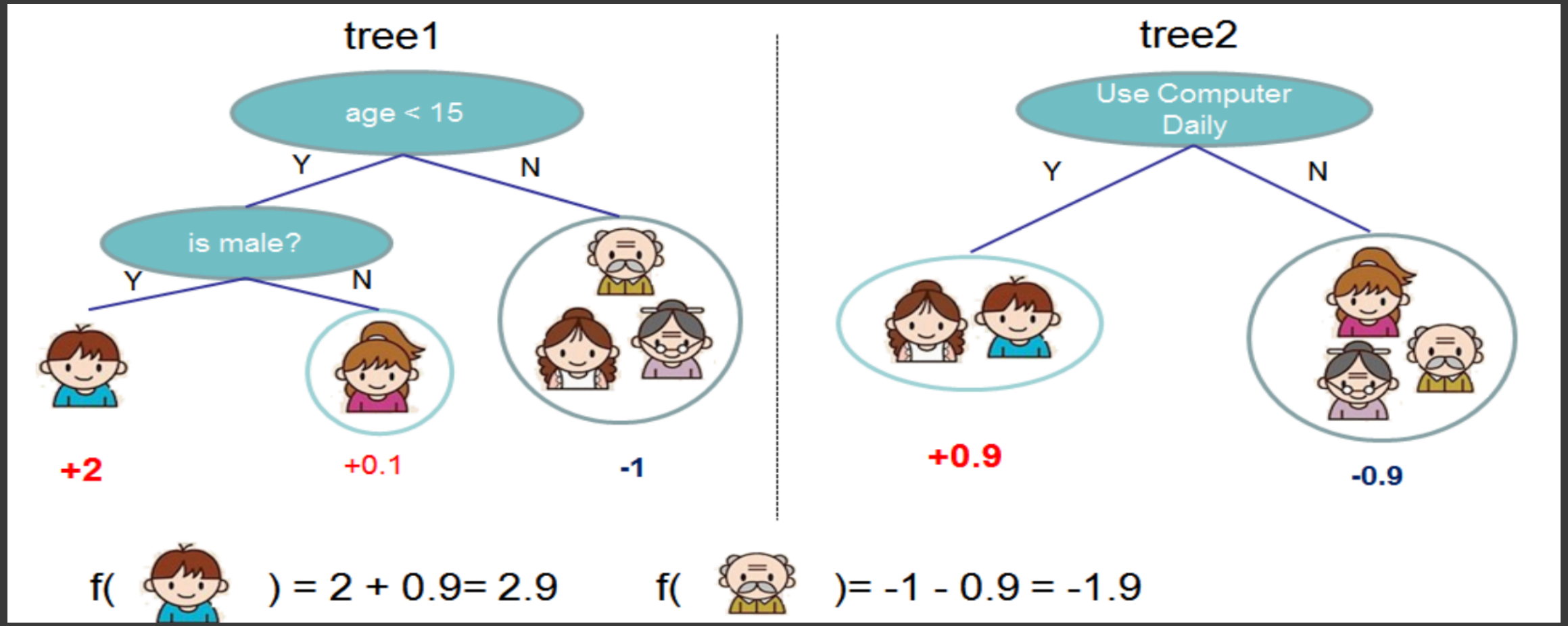
- XGBoost is extreme gradient boosting algorithm based on trees and tends to perform very good out of the box compare to other ML algorithms.
- XGBoost is popular amongst data-scientist and one of the most common ML algorithms used in Kaggle Competitions.
- XGBoost allows one to tune various parameters.
- XGBoost allows parallel processing.
- Works very well with Python and Scikit Learn.

XGBoost



Source: Official Tutorial => https://xgboost.readthedocs.io/en/latest/tutorials/model.html

XGBoost (continue ...)



Source: Official Tutorial => https://xgboost.readthedocs.io/en/latest/tutorials/model.html

XGBoost (continue ...)

- Model: assuming we have K trees

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F}$$

- Objective

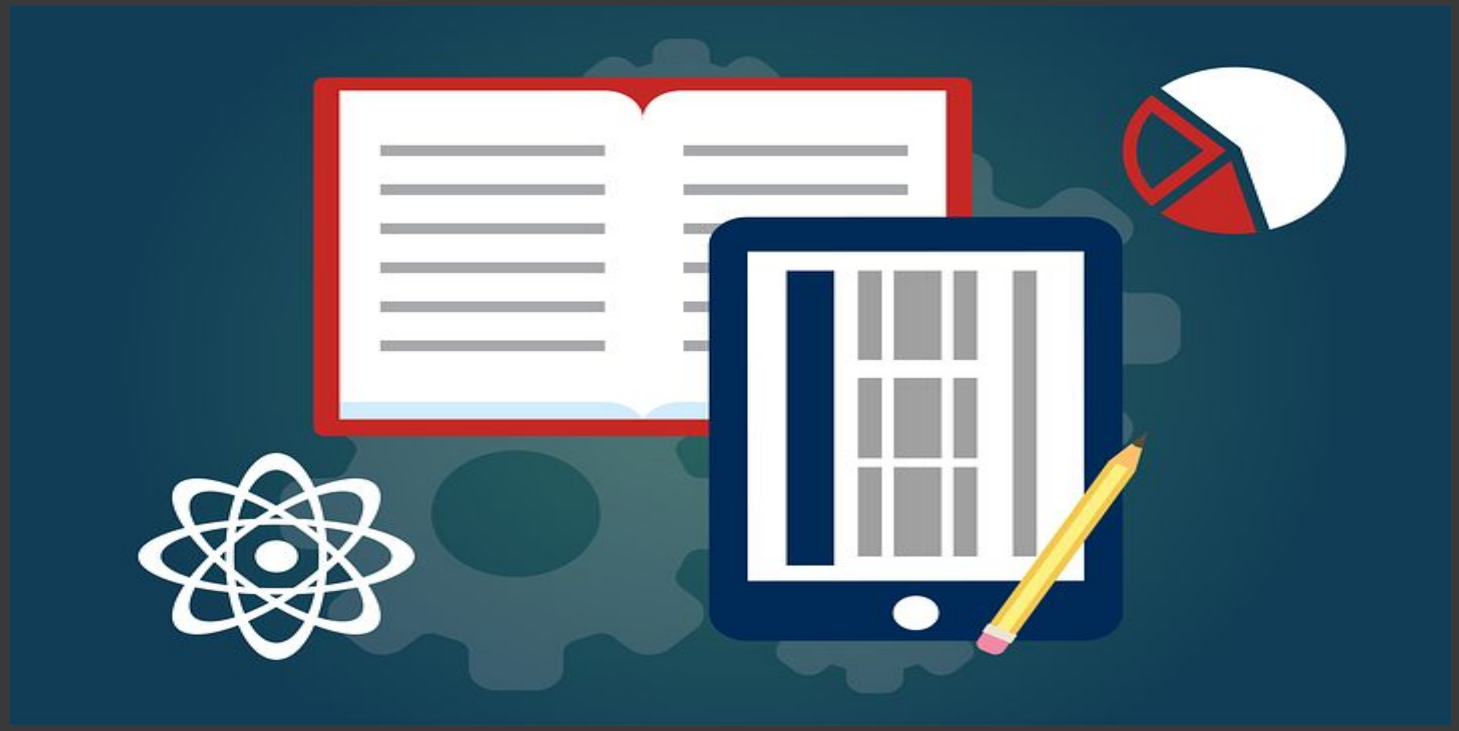
$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Training loss

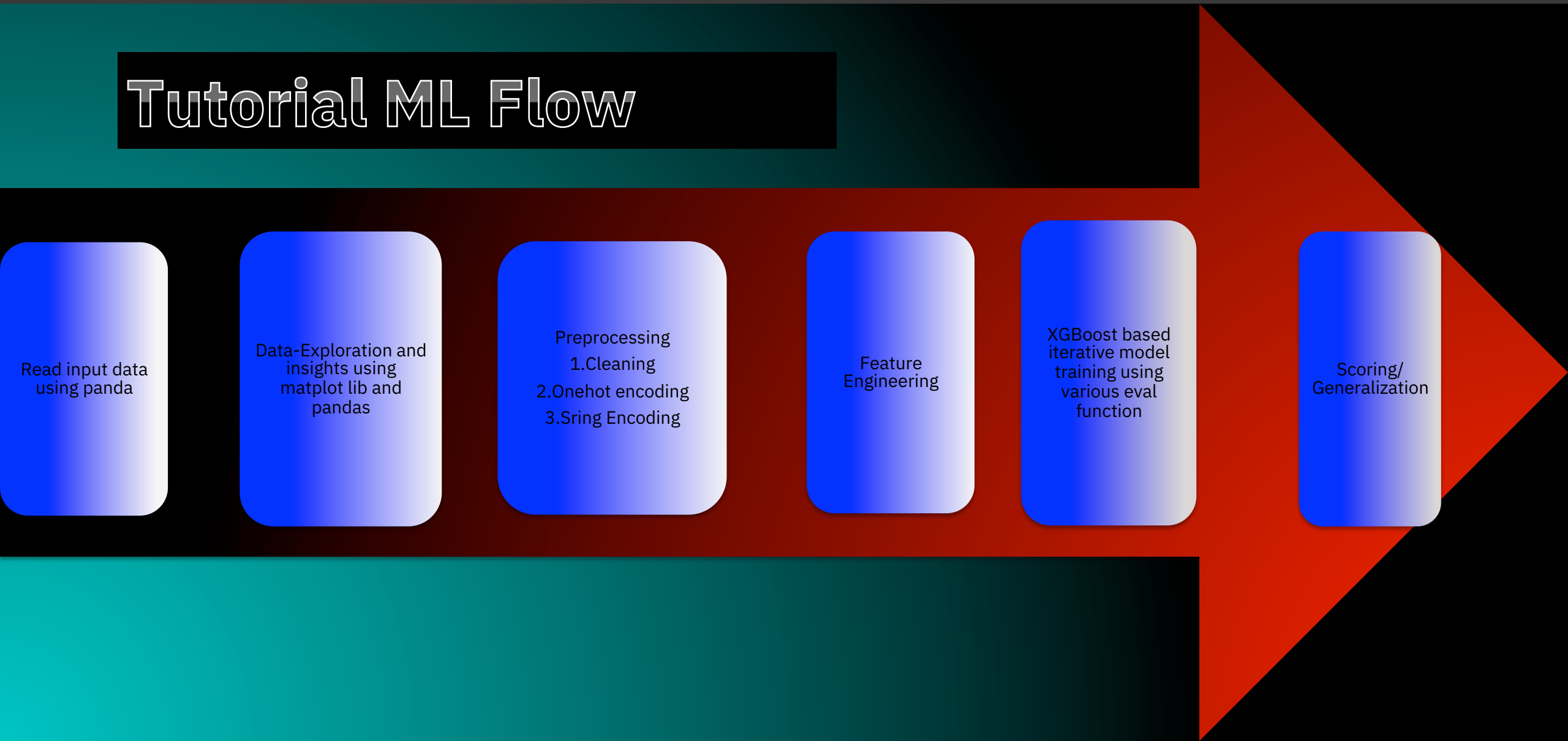
Complexity of the Trees

Source: From the XGBoost Author=> https://homes.cs.washington.edu/~tqchen/pdf/BoostedTree.pdf

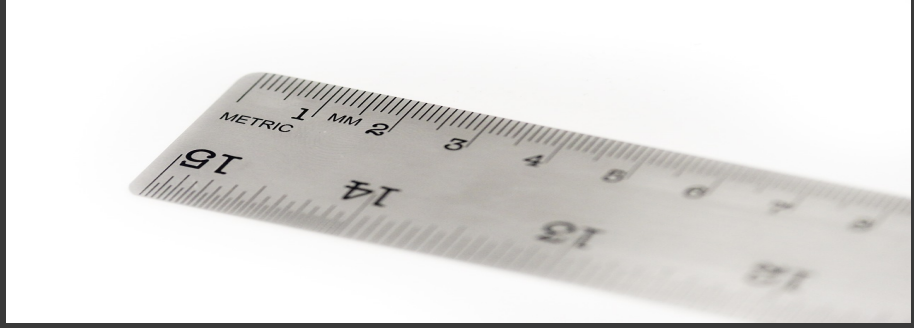
Overview of the workshop



Scikit-based ML flow



Metrics for Model Performance.



To come up with the best model, we should evaluate model performance for comparison among various models. There are many ways to evaluate the model performance for classification. Before we go into model training we will understand various model evaluation criteria

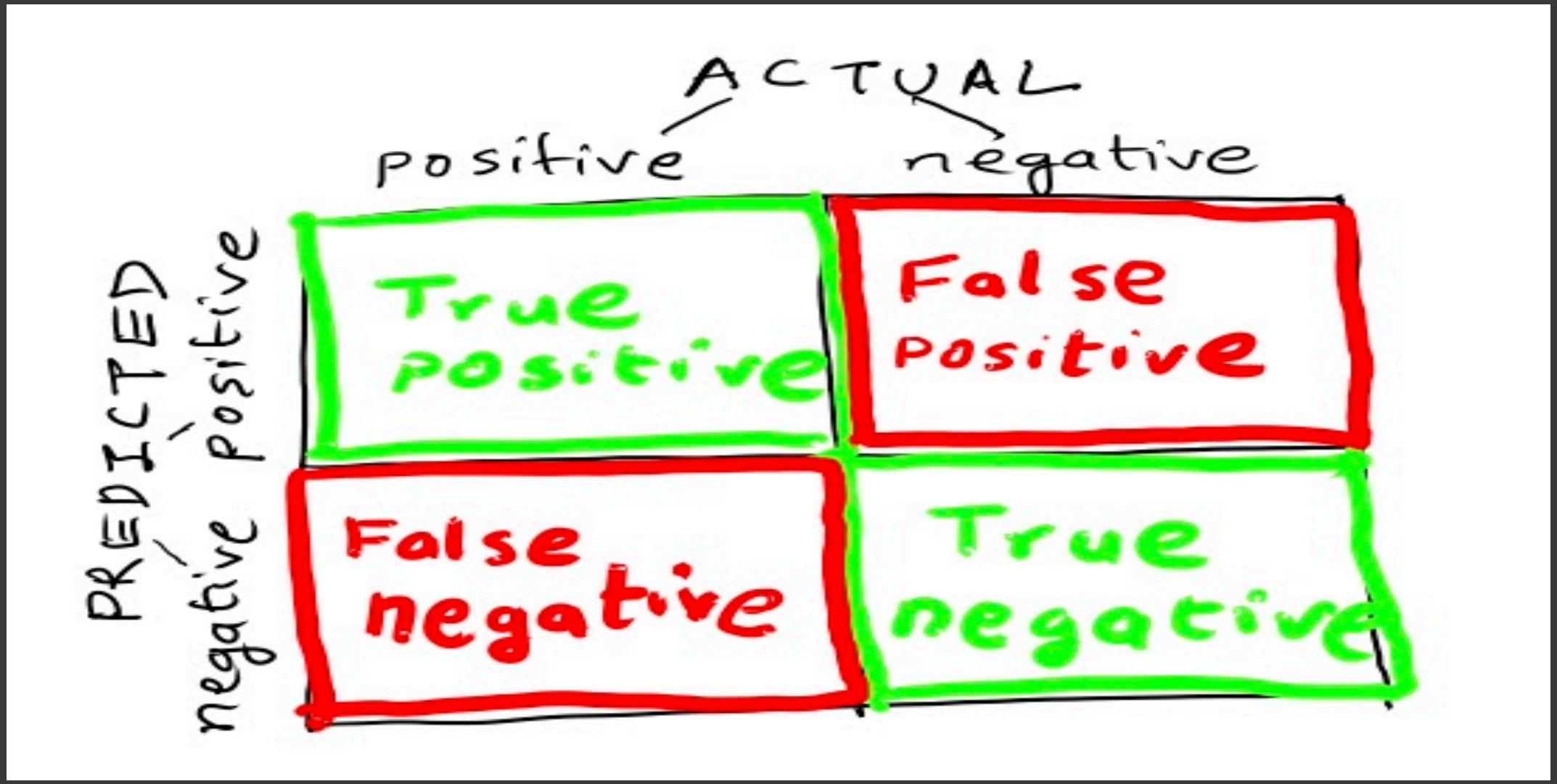
1) Accuracy Score

2) Confusion Matrix

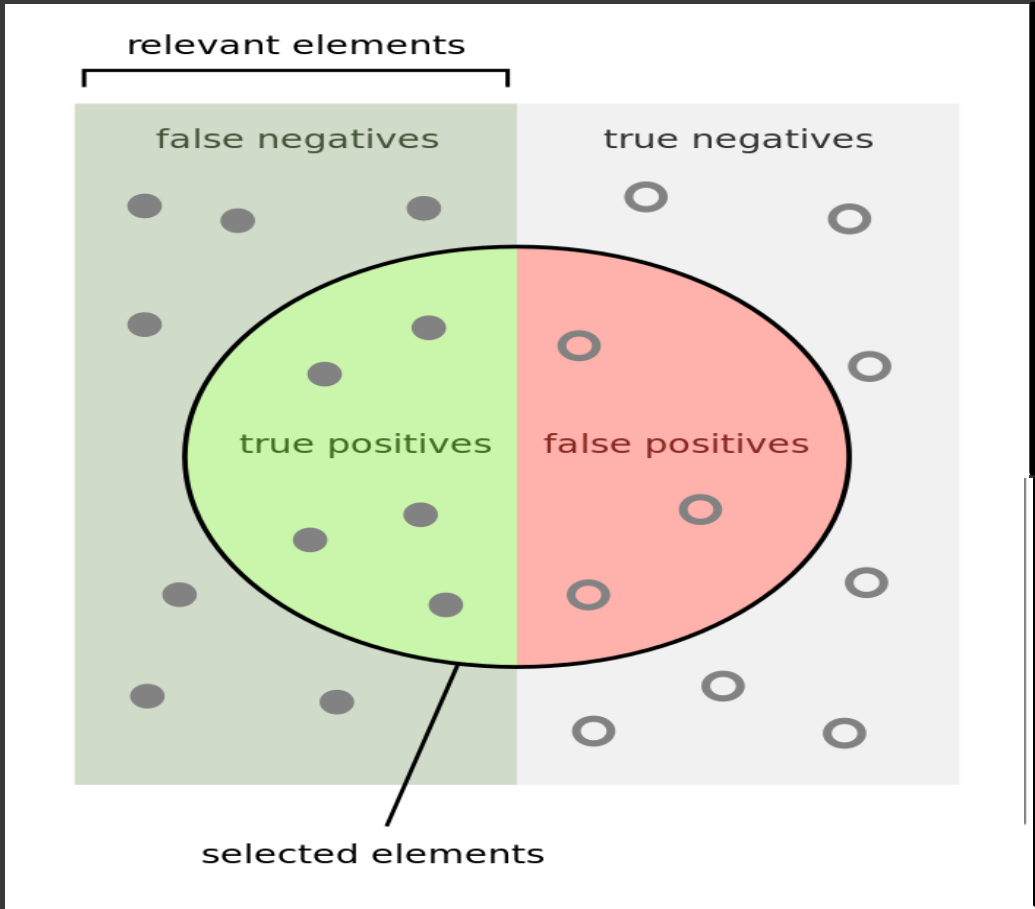
3) ROC curve

4) Precision Recall Curve

Confusion Matrix



Receiver Operating Characteristics (ROC)

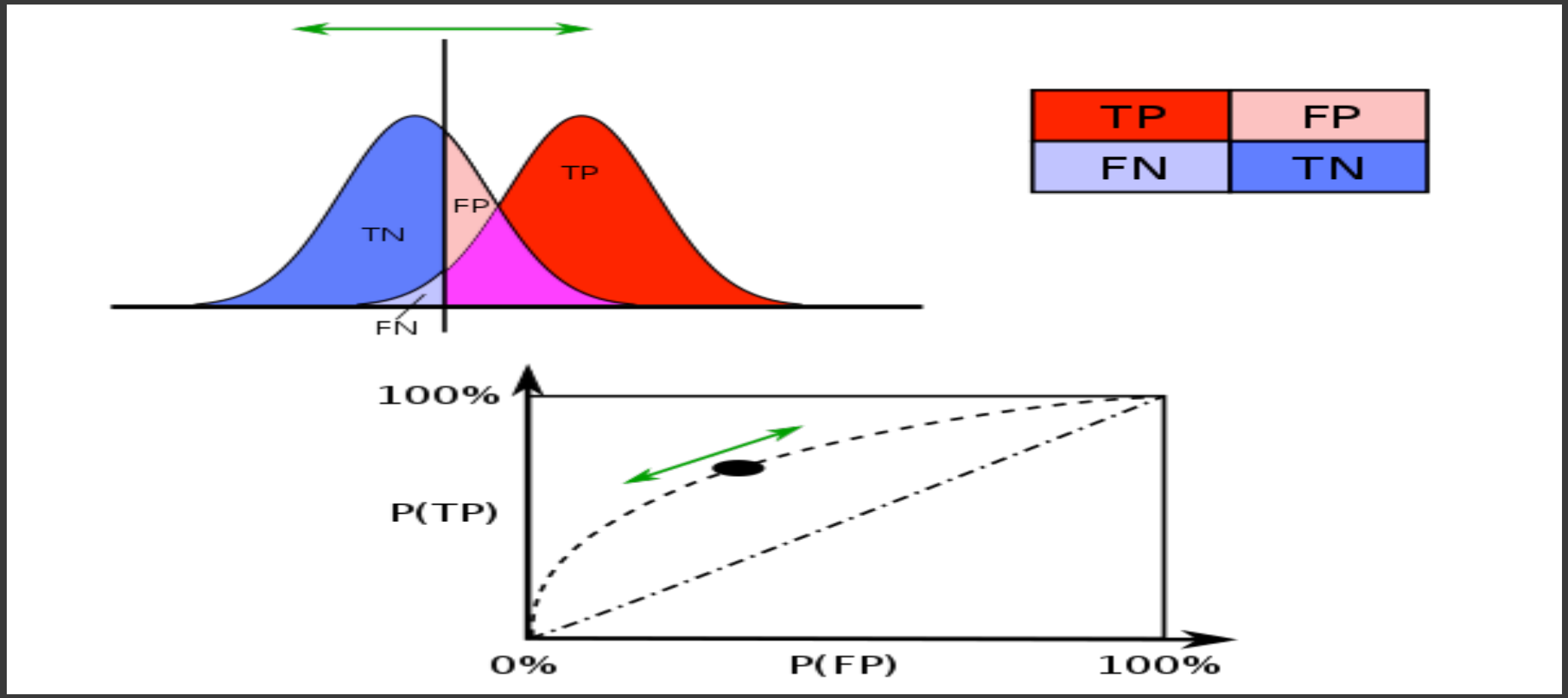


Sensitivity= $\frac{\text{true positives}}{\text{relevant elements}}$ = TP/(TP+FN)

Specificity = $\frac{\text{true negatives}}{\text{non-relevant elements}}$ = TN/(TN+FP)

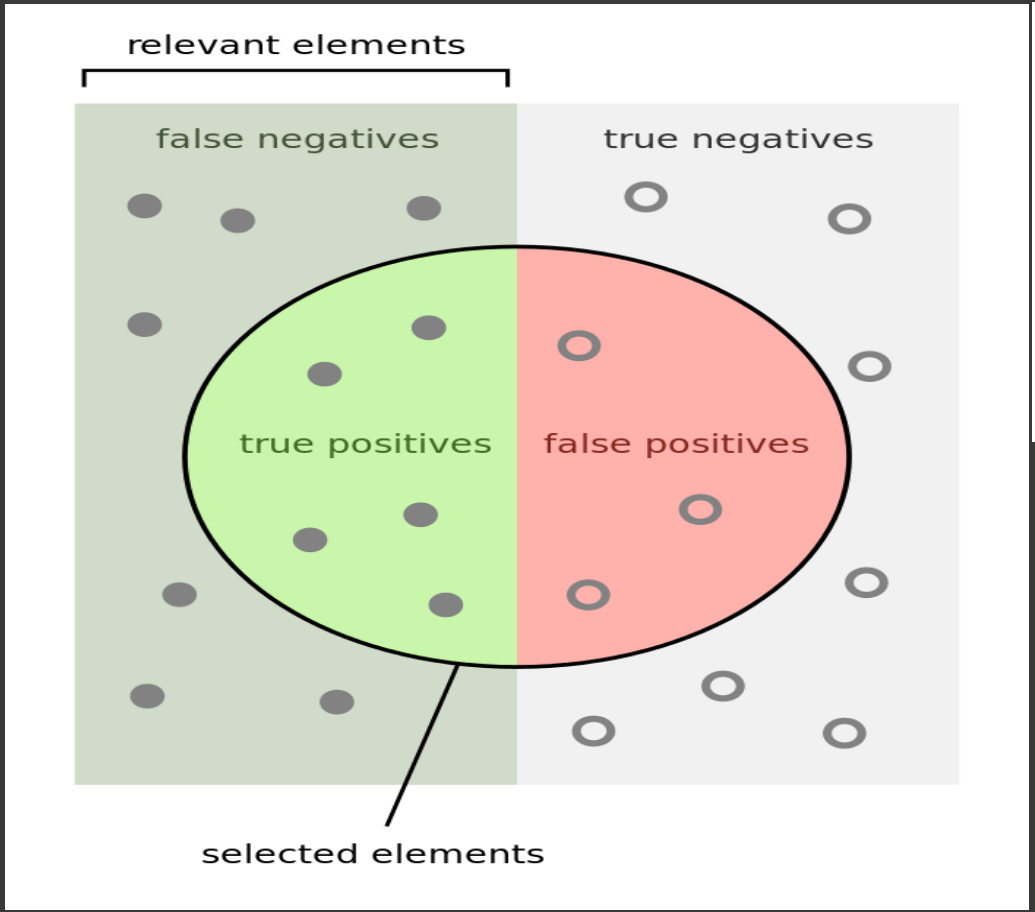
By FeanDoe - Modified version from Walber's Precision and Recall https://commons.wikimedia.org/wiki/File:Precisionrecall.svg, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=65826093

ROC (continue...)



Source:Wikipedia By Sharpr - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=44059691

Precision-Recall Curve(PR Curve)



How many selected items are relevant?

Precision = $\frac{\text{true positives}}{\text{true positives + false positives}}$

$$\frac{TP}{(TP+FP)}$$

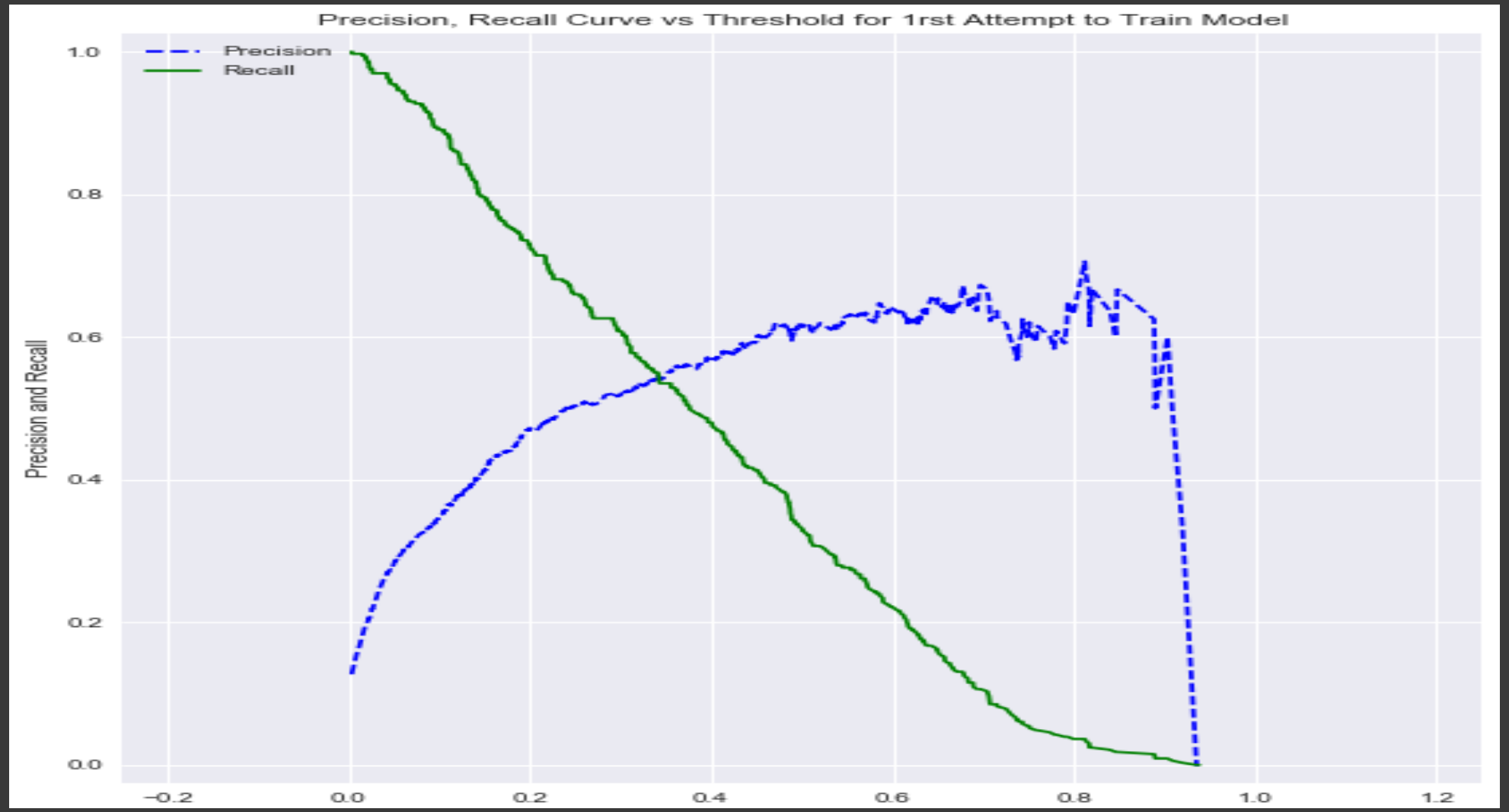
How many relevant items are selected?

Recall = $\frac{\text{true positives}}{\text{true positives + false negatives}}$

$$\frac{TP}{(TP+FN)}$$

By Walber [CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0)], from Wikimedia Commons

PR Curve (Continue...)



F1-Score

- Since there is the inverse relationship between precision and recall .
- You increase one , other will go down and vice versa.
- And hence if we want to do justice to both the metrics, we usually use the combination of both .
- We use harmonic mean to combine them.

F1Score = Precision*Recall/(Precision+Recall)

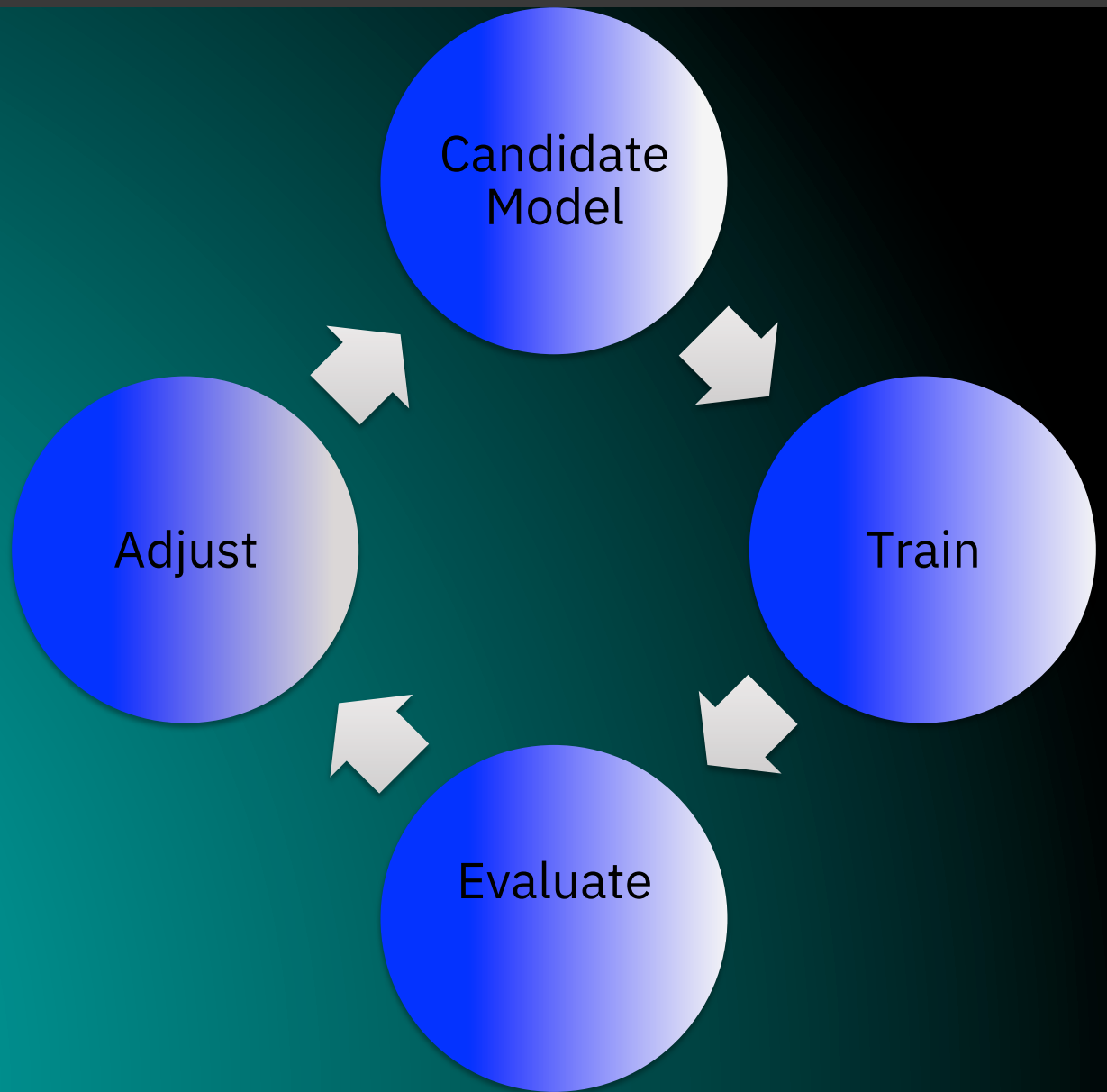
Which Metrics to use for Imbalance Datasets?

- We will use precision recall with varying threshold to find out the best threshold
- But we should note that Precision and recall have inverse relationship and hence if we want high precision then we should be fine with low recall and vice versa.

- There are many applications where high precision might be desired. For example if we train a model to detect safe website for kids. It's ok to reject many safe website (low recall) as long as we correctly reject bad website (high precision)
- However for our applications, i.e bank client's CD subscription prediction, higher recall is desirable. Since it is perfectly fine to have false client i.e model predict client will accept the subscription but will actually decline it, as long as we predict almost all the client who is likely to accept the subscription and thus improving the balance sheet.

Model Training

Model Training Iteration



First Attempt to XGBoost Model Training

- We will split dataset into training and test datasets
- We will first create a simple XGBoost model with using traditional ML techniques.
- We will evaluation models using various metrics we discussed.
- Analyze the reports and concludes the model is not good.

Strategies for better Classifier for the Imbalance Dataset.

1. Use the weighted class i.e give higher weights to minority positive class i.e 'yes’ label.
2. Over-Sampling of the minority class and under-sampling of the majority class.
3. Use the SMOTE (Synthetic Minority Oversampling Techniques).

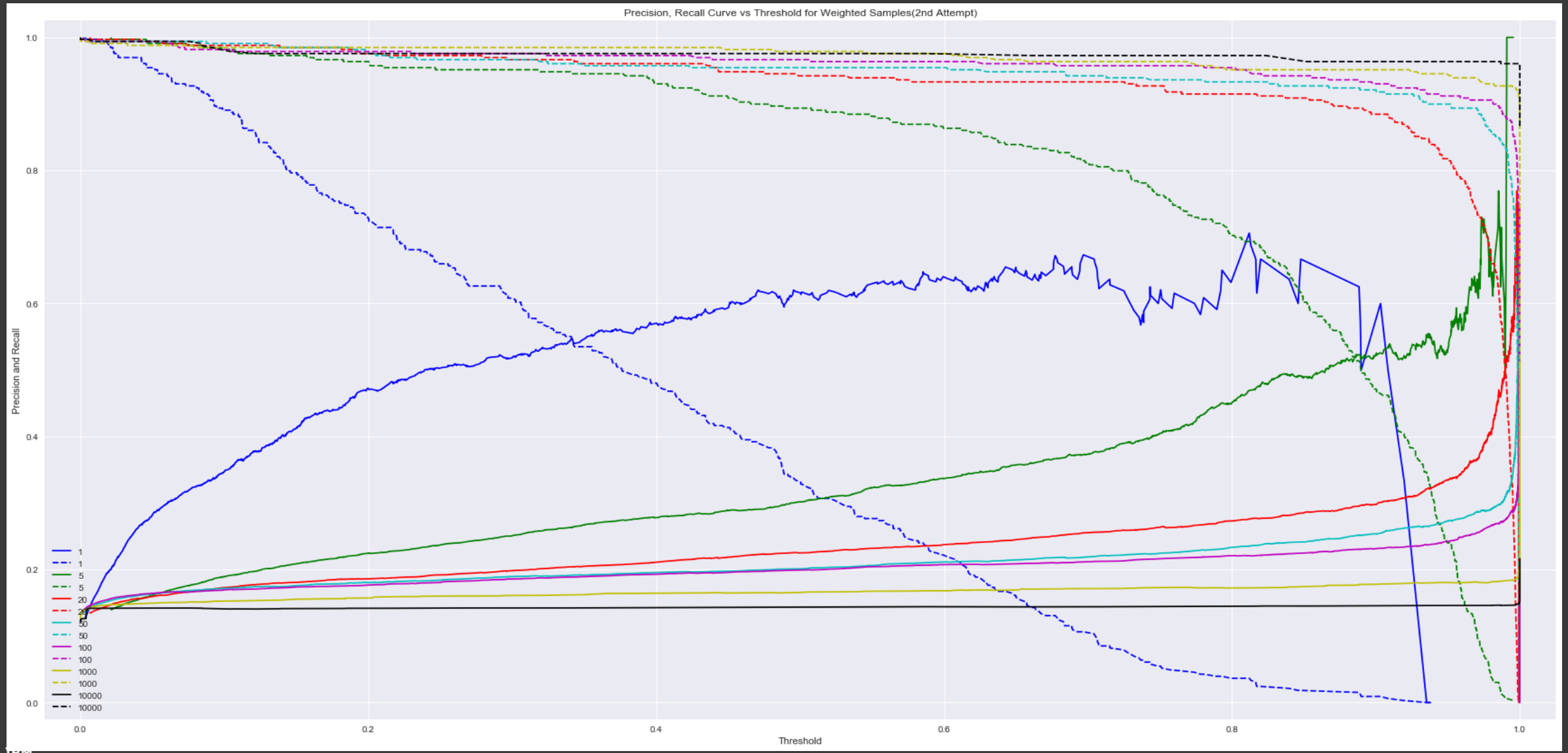
Weighted Class classifier

Since at training time, classifier doesn’t learn from minority samples

The natural approach would be to some how increase the number of minority samples

One way to do is to give more weightage to minority class at the time of evaluation of the node split.

Weighted Class classifier (continue…)



Minority Over-Sampling and Majority Under-sampling

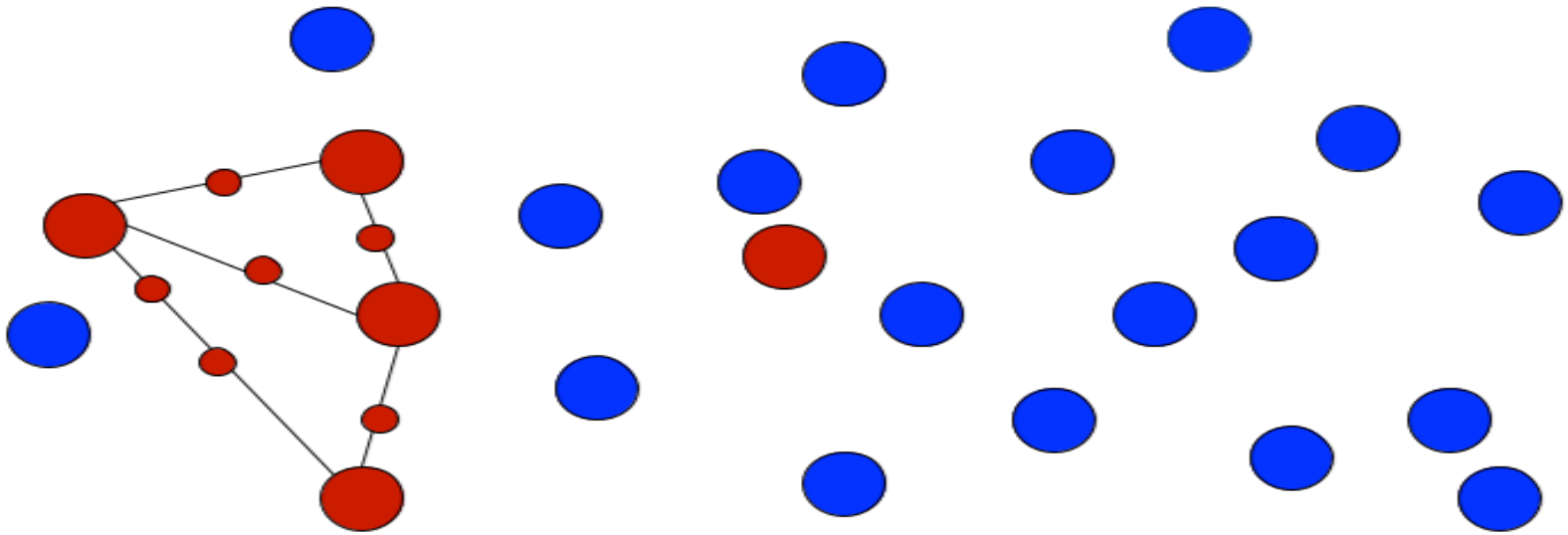
- Idea is similar to Weighted class
- Instead of giving higher weights, we just replicate a subset of random minority samples
- Also we can under samples the majority class i.e don’t take all the samples from majority class.

SMOTE algorithm (Synthetic Minority Oversampling Technique)

- Intuitively, just replicating the minority sample sounds cheating.
- So better approach is to have new synthetic samples created by interpolating and extrapolating the two or more samples.

- The candidate samples for interpolation can come from majority or minority or the combination of them.
- The suggestion by SMOTE is to use the samples on the decision boundary
- Paper : https://arxiv.org/pdf/1106.1813.pdf

SMOTE (continue...)



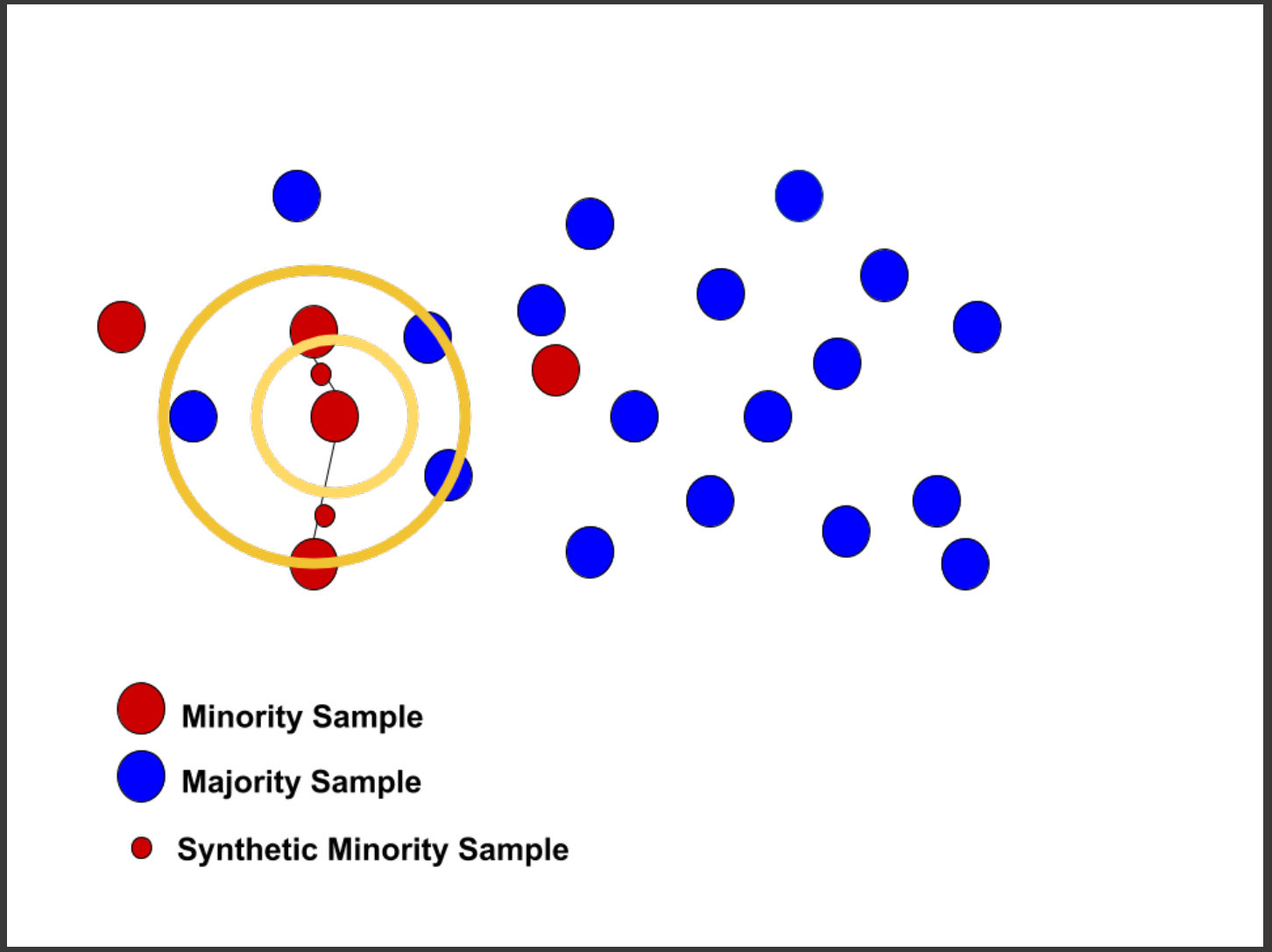
Minority Sample

Majority Sample

Synthetic Minority Sample

SMOTE(continue…)

- Nearest neighbor minority samples synthesis.
- For the center red circle for K=2, we have two neighbors and we have possibility of two synthesis samples for dup_size=1,
- We randomly picks one of the small new red samples.



Hands on tutorial for building predictive classifier for imbalance datasets.

Checklist

1) Look the package installation guide accompanying this tutorial.

2) Clone the github:

   https://github.com/aloknsingh/ds_xgboost_clf_4_imbalance_data

3) Data: data/bank.csv
4) Notebook: notebooks/predict_bank_cd_subs_by_xgboost_clf_for_imbalance_dataset.ipynb

5) You can use one of the following platforms

   1) Your laptop

   2) IBM Watson studios

4. Our goal is to finish notebook in 50 minutes.

Workshop Begins…

Q/A