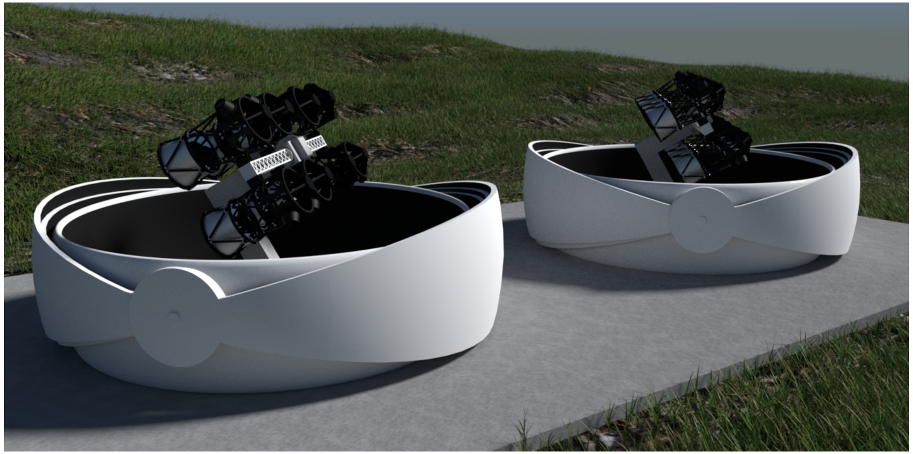
Machine Learning Object Classification for GOTO



R. Yoyponsan, U. Sawangwit, S. Komonjinda

Attānam Damayanti Paṇḍitā

CHIANG MAI UNIVERSITY SINCE 1964

NARIT

MINISTRY OF SCIENCE AND TECHNOLOGY

GOTO

GRAVITATIONAL-WAVE OPTICAL TRANSIENT OBSERVER

Outline

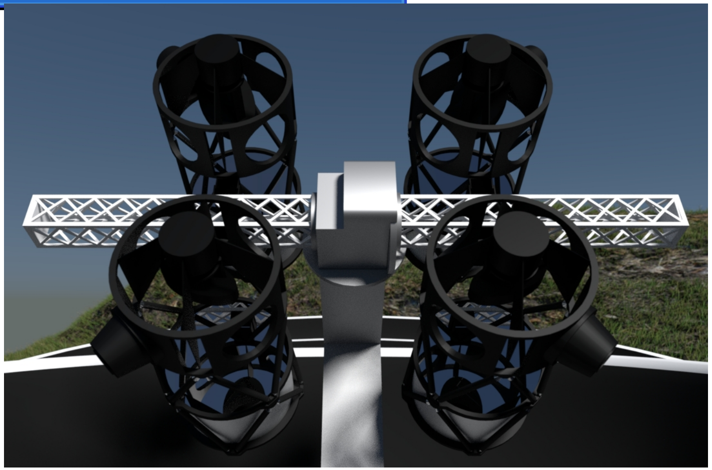
- Introduction
- GOTO specifications
- Estimated Gravitaional Wave source position
- Challenges
- Existing solutions
- Our task
- Procedure
  - Simulated data
  - Stratified k-folded cross validation
  - Machine Learning : Random Forest, Artificial Neural Network, k-Nearest Neighbors
  - Evaluation : Receiver Operation Characteristic (ROC)

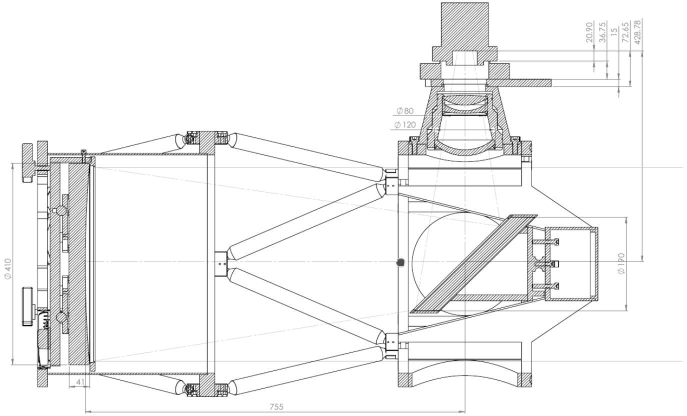
Inroduction : GOTO

- Gravitational-wave Optical Transient observers (GOTO)
- Robotic, wide field-of-view, high cadence telescopes
- Response to signal triggered by GW detectors (LIGO, VIRGO, ...)
- Complete the whole sky every 6 nights
- High data production rate
- Located at La Palma Observatory

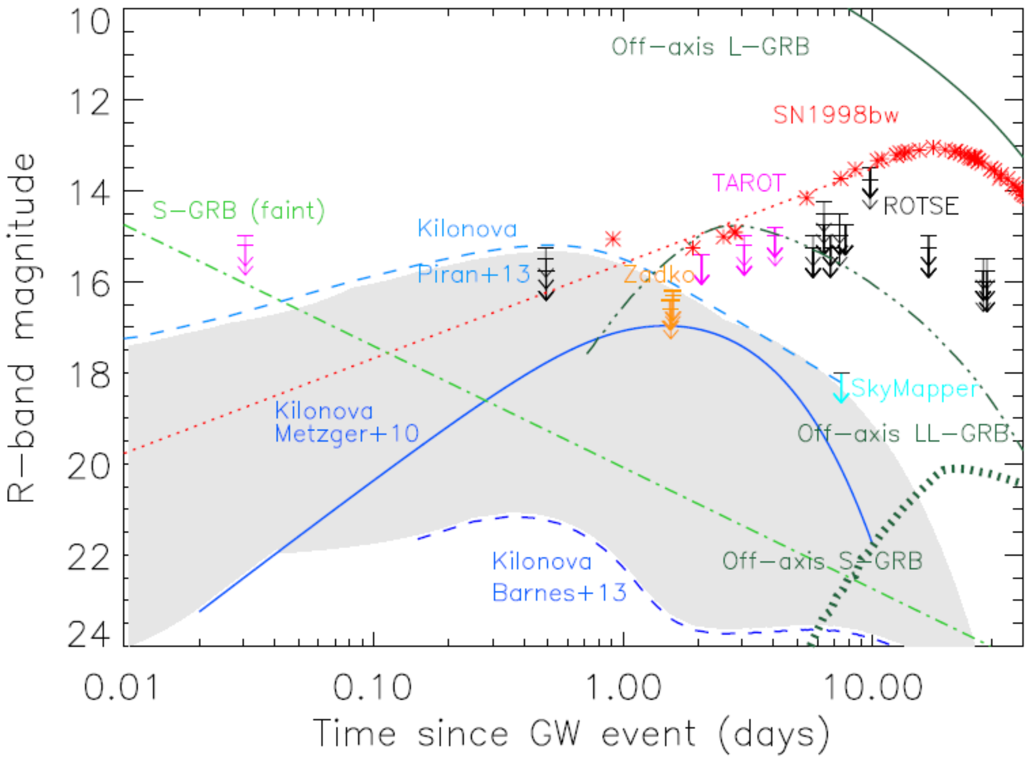
GOTO specification

- 4x40cm f/2.5 Wynne-Ricardo OTAs from APM professional telescopes Gmbh
- Shared fast-slewing GE mount
- Field of view ~18 sqr.deg (4.5 per OTA)
- 4x50 Mpixel detectors at 1.2”/pixel
- 20-21 mag in wide optical filter in 5-10 mins
- 5s readout

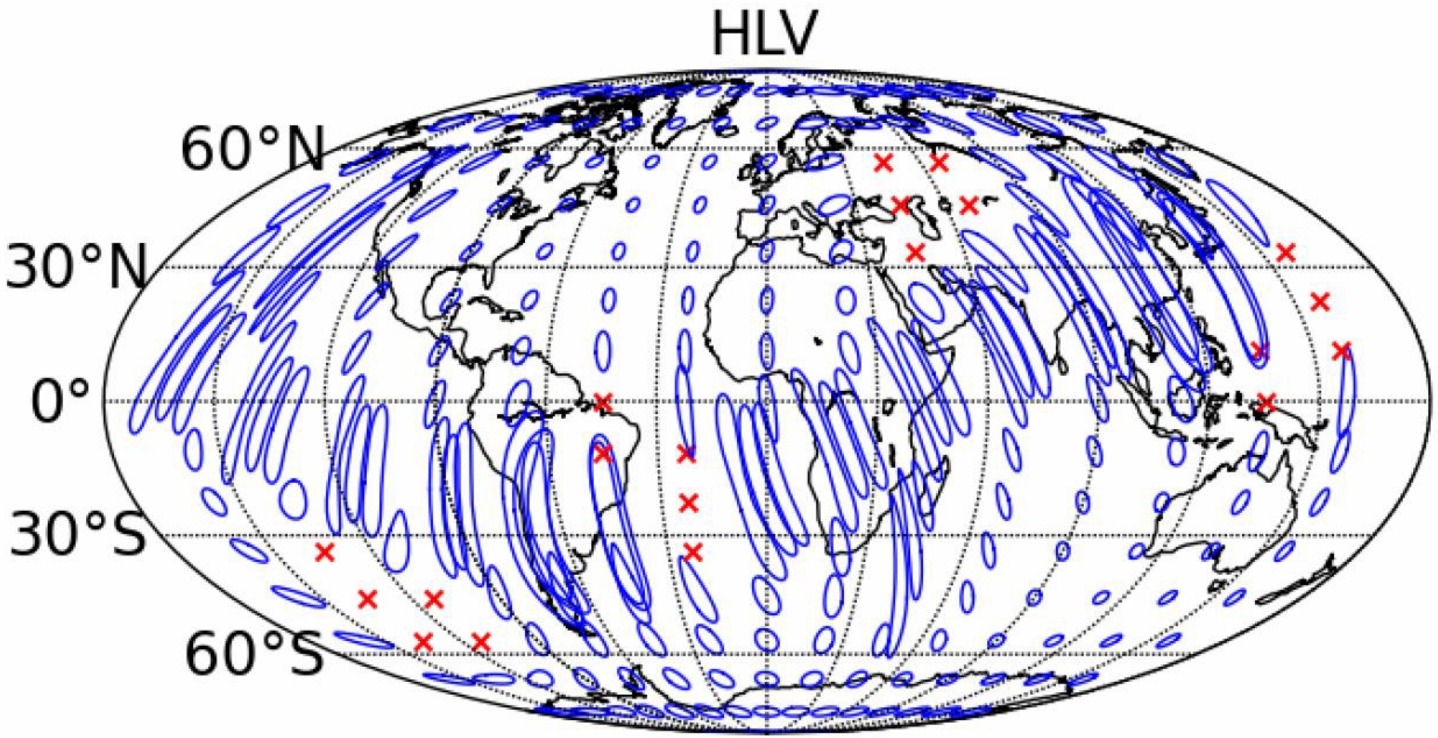




GOTO Specification



Estimated Gravitational Wave (GW) Source Location (present)

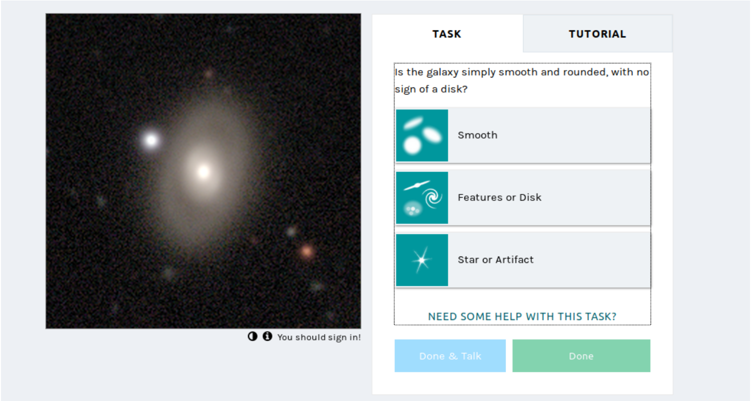


Challenges

- Require fast onsite data processing pipeline and transient detection pipeline
- Given the volume of data, it is nearly impossible to process by hand

Existing Solutions to Large Data Problem

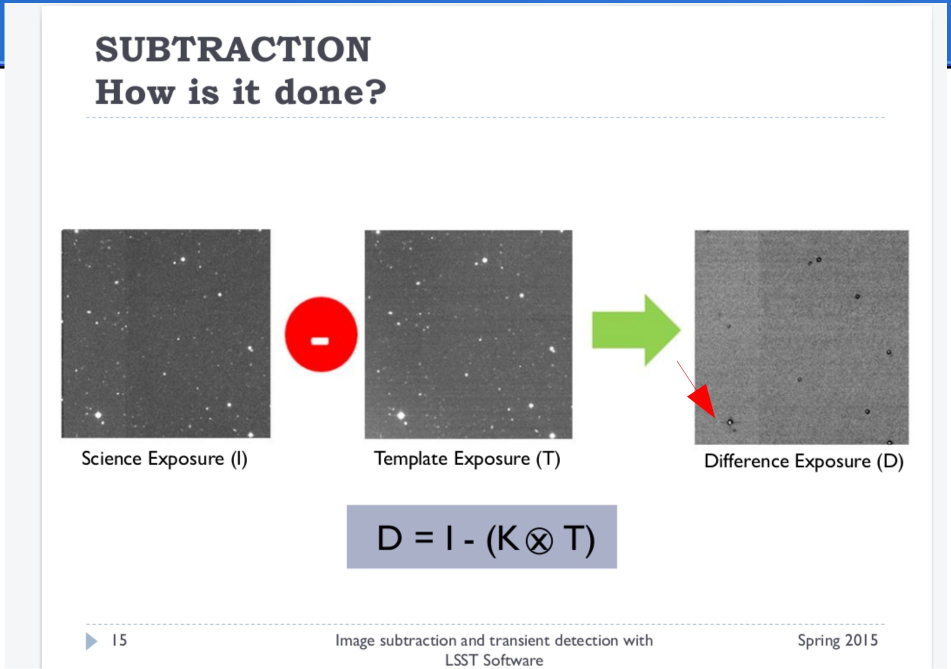
- Human scan : galaxyzoo – more than 10,000 volunteers helped to classify galaxy type from multiple catalogs; such as, SDSS, Chandra, etc. The current data comes from Dark Energy Camera Legacy Survey (DECaLS)

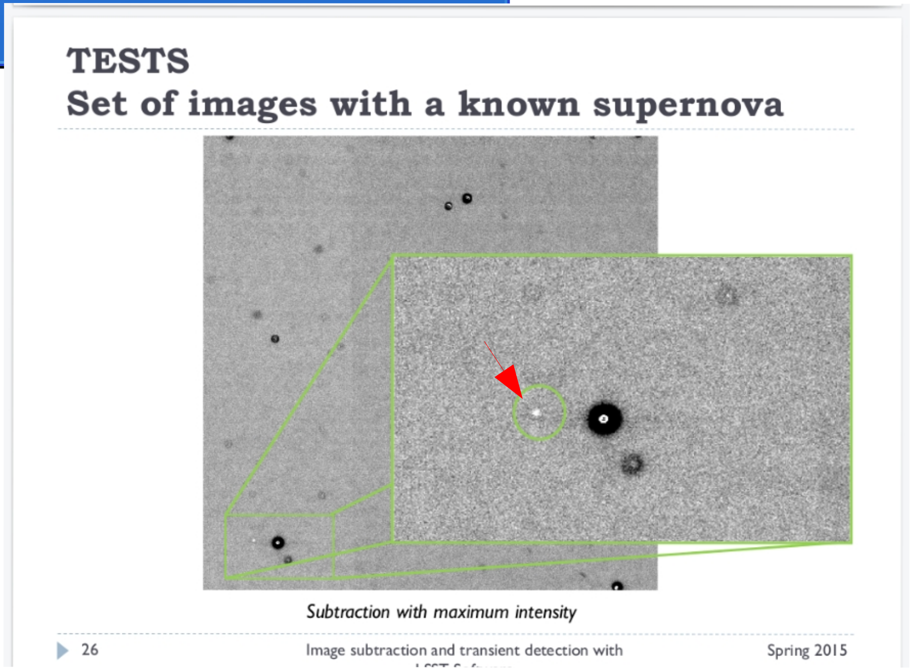


Existing Solutions to Large Data Problem

- Automated process using pipeline and ML – applied ML to detect transient events
  - Palomar Transient Factory (PTF; Smith et al. 2011),
  - Panoramic Survey Telescope and Rapid Response System Medium Deep Survey(Rest at al. 2014),
  - Dark Energy Survey (DES; Flaugher 2005)
- Transient detection use image subtraction technique

Existing solutions





- https://indico.in2p3.fr/event/11235/contributions/4789/attachments/4114/5176/PresentationReunionMontpellier.pdf

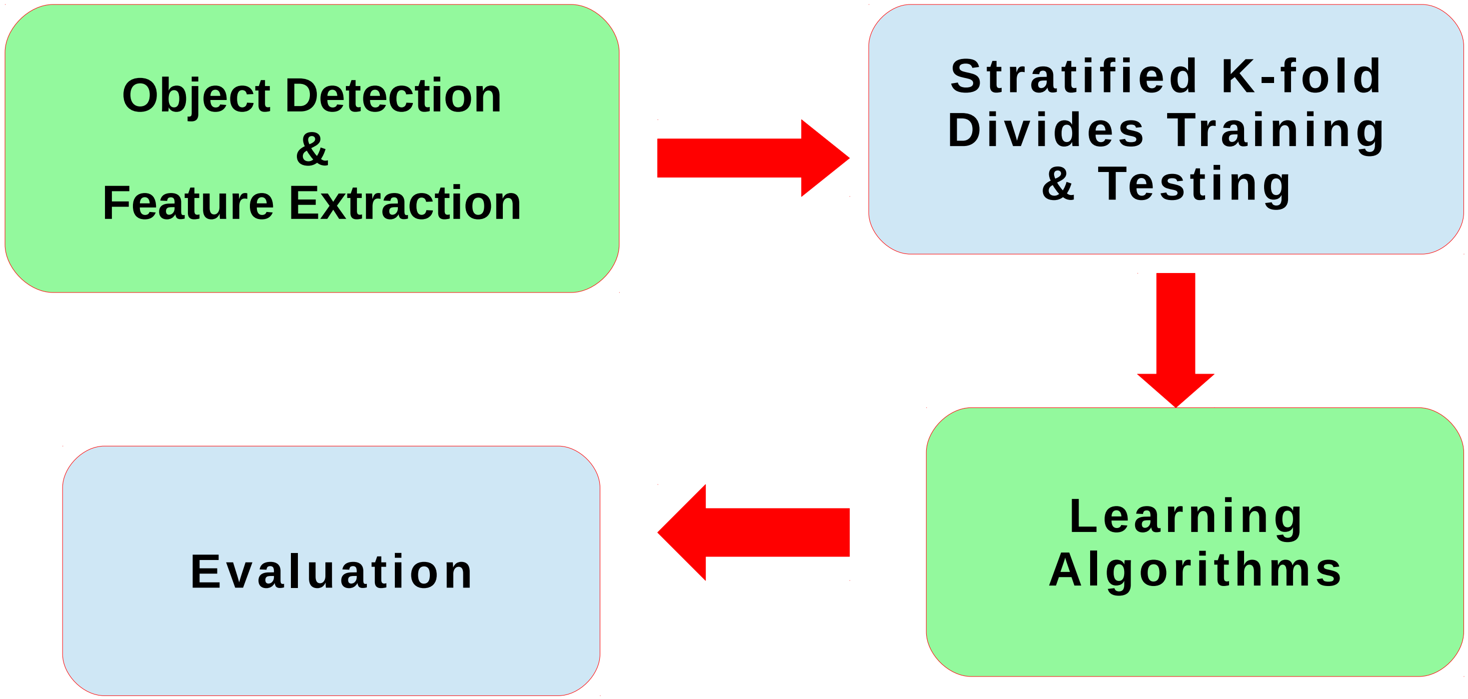
Our task

- Handle reduced images WITHOUT subtraction
- At present, we work on object detection and classification to seperate point sources from extended sources
- Experiment on Machine Learning (ML) classification with 3 algorithms; namely
  - Random Forests (RF)
  - Artificial Neural Network (ANN)
  - K-Nearest Neighbors (KNN)
- Use simulated image
- The pipeline is based on Pyhton programming language.

Machine Learning Technique	AUC	Accuracy	Recall	Precision	F1-score	Confusion Matrix		
Random Forest (RF) Section 5.6	**0.97**	**0.91**	0.91	**0.93**	**0.92**		**Object**	**Not Object**
						Object	3541	342
						Not Object	259	2732
K-Nearest Neighbours (KNN) Section 5.3	0.94	0.89	0.90	0.91	0.90		**Object**	**Not Object**
						Object	3506	377
						Not Object	363	2628
SkyNet Section 5.5	0.94	0.88	0.89	0.90	0.89		**Object**	**Not Object**
						Object	3461	422
						Not Object	399	2592
Support Vector Machine (SVM) Section 5.4	0.93	0.86	0.90	0.85	0.87		**Object**	**Not Object**
						Object	3514	369
						Not Object	605	2386
Minimum Error Classification (MEC) Section 5.1	0.90	0.84	**0.92**	0.83	0.87		**Object**	**Not Object**
						Object	3559	324
						Not Object	754	2237
Naïve Bayes (NB) Section 5.2	0.80	0.77	0.86	0.77	0.81		**Object**	**Not Object**
						Object	3333	550
						Not Object	998	1993

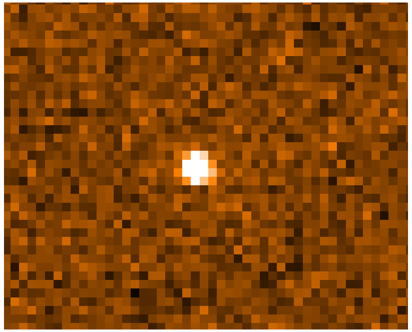
Du Buisson, et al. 2014

Procedure

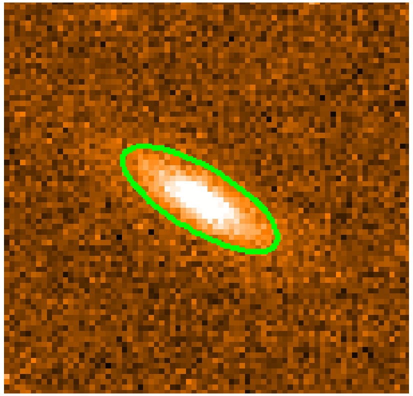

Object Detection
&
Feature Extraction
Stratified K-fold
Divides Training
& Testing
Learning
Algorithms
Evaluation

Simulated data

- Simulated Data generated by SkyMaker
- 8176 X 6132 pixels in total of 16 images
- FWHM of point spread function = 0.9
- Pixel size = 1.24 arcsec
- Clear filter
- In total, we are able to detect 249,640 objects
  - Point sources : 161,386 , assigned to class 1
  - Extended sources : 88,251 , assigned to class 0

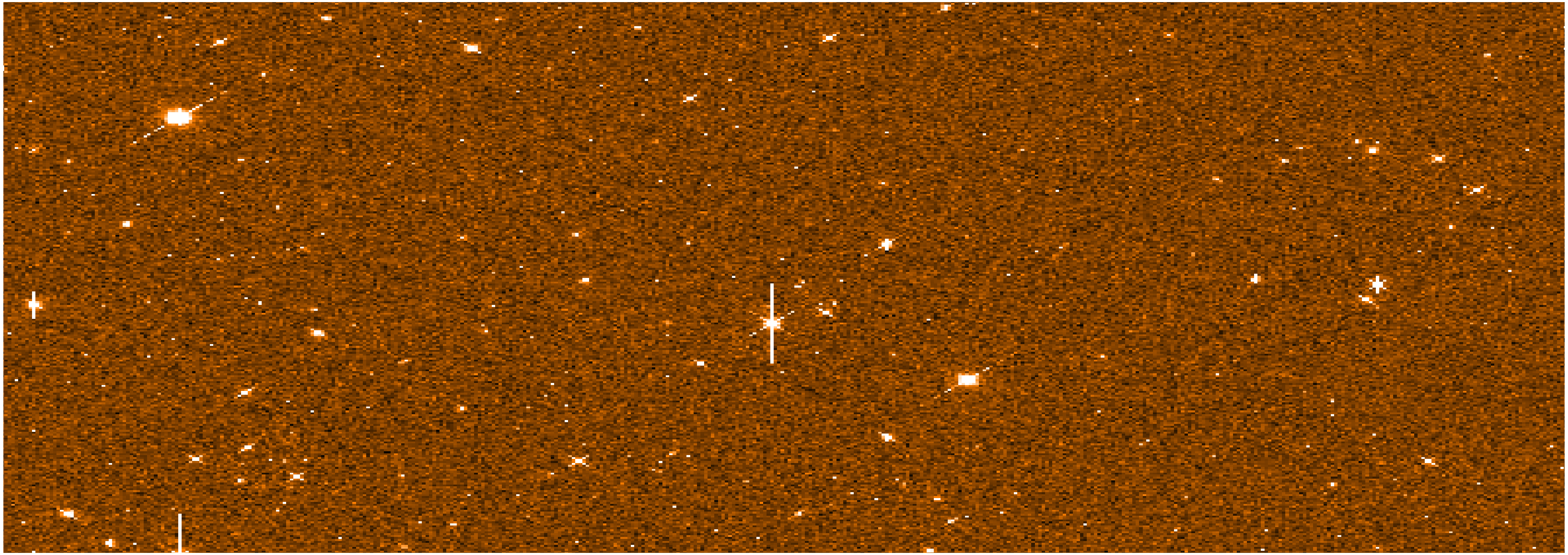


Class 1 :
Point source

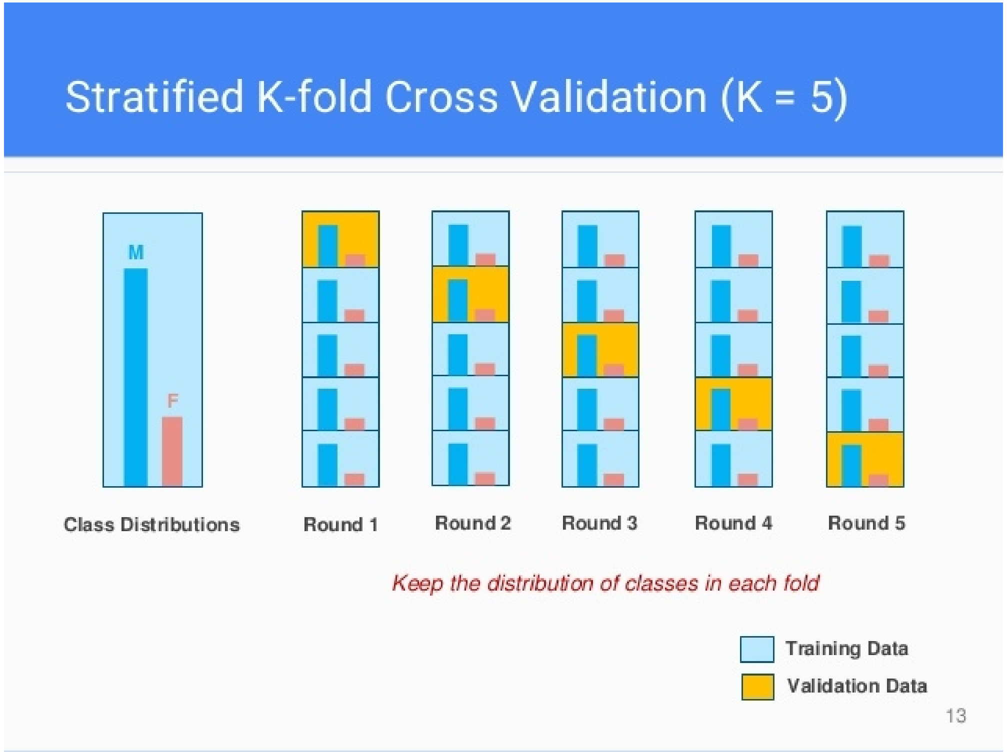


Class 0 :
Extended
Source

Image example



Train/Test data – Stratified k-Fold Cross validation



https://www.slideshare.net/markpeng/general-tips-for-participating-kaggle-competitions

By Mark Peng

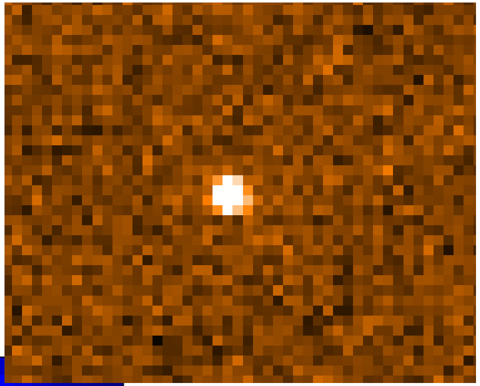
Object Detection and Extraction

- Use Source Extractor (Bertin & Arnouts, 1997)
- Detection threshold of 2 sigma
- Example of features from 27 Source Extractor features:
  - **FLUX_APER** – Flux vector within fixed circular aperture (default diameter 5 pixels)
  - **KRON_RADIUS** – Kron apertures
  - **A_IMAGE** – Profile RMS along major axis
  - **B_IMAGE** – Profile RMS along minor axis
  - **ELLIPTICITY** – 1 - B_IMAGE/A_IMAGE
  - **CLASS_STAR** – S/G classifier output f rom Source Extractor’s neural network

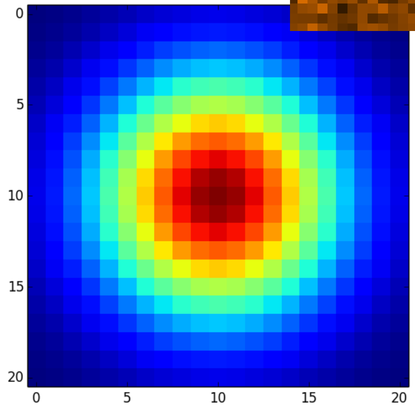
Extract goodness-of-fit test for 2d gaussian test

- We assume circular gaussian profile for point source, with addition back ground offset :

$$f(x,y)=a_0+a_1\exp-\left(\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}\right)$$



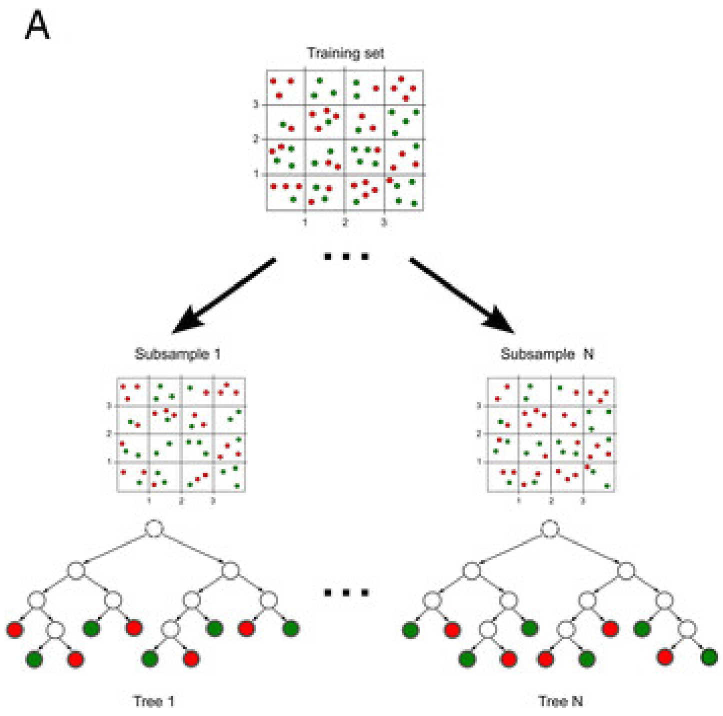
- 2D Gaussian fitting in 20x20 pixels surrounding a centroid
- Calculate chi2 as a goodness-of-fit test
- **require parallel computing to speed up the process



In Total we have 28 features for machine learning process.

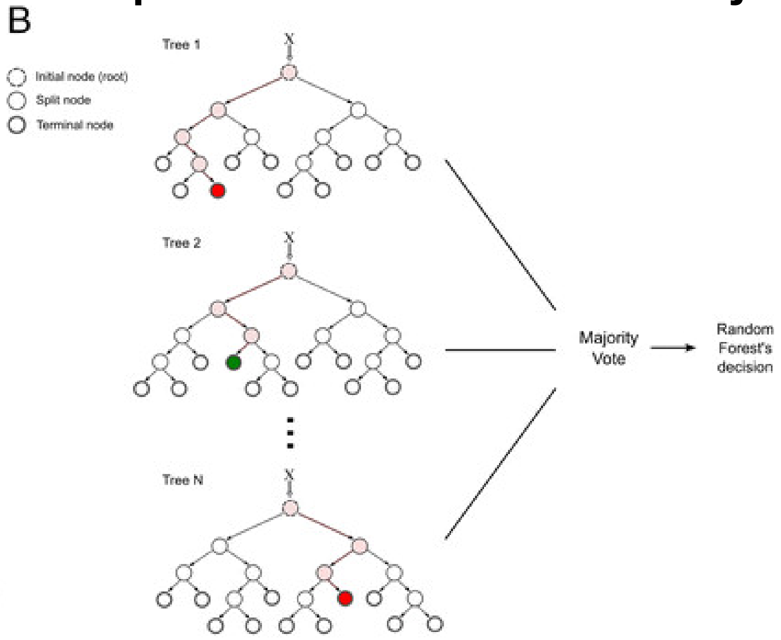
Random Forests (RF)

- Based off of decision tree
- Each tree in the forest is built from a random bootstrap sample of the original data (Machado et al. 2015, *Veterinary Research*)



Random Forests (RF)

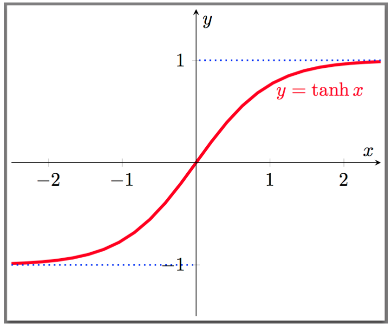
- The prediction is the majority result from all trees

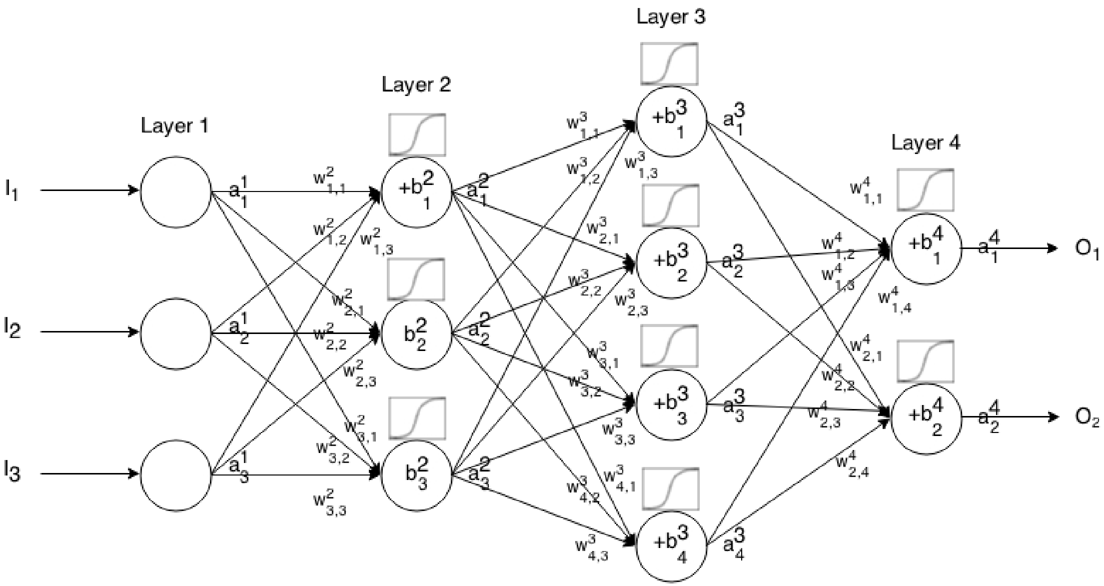


Machado et al. 2015

Artificial Neural Network (ANN)

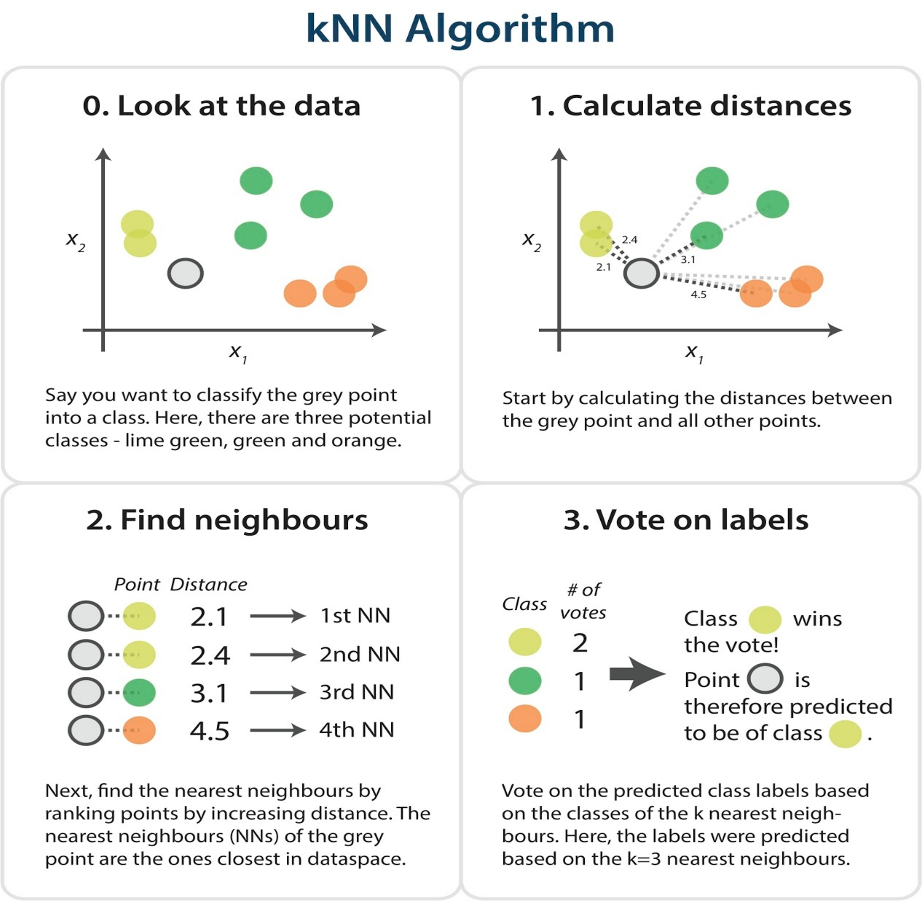
- Imitate biological nerual network by creating neurons (nodes) and link between neurons (vertices)
- Feed inputs forward
- Learn by adjust weight on each vertices by eror minimization (Back propagation)
- Our model uses tanh as an activation function





k-Nearest Neighbors (KNN)

http://garylv.github.io/machine-learning/2016/05/01/k-nearest-neighbours/



Receiver Operating Characteristic (ROC)

- A graph showing relation between true positive rate (tpr) and false positive rate (fpr)of a classifier

$$\mathrm{TPR} = \frac{\mathrm{TP}}{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{FNR}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{N} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = 1 - \mathrm{TNR}$$

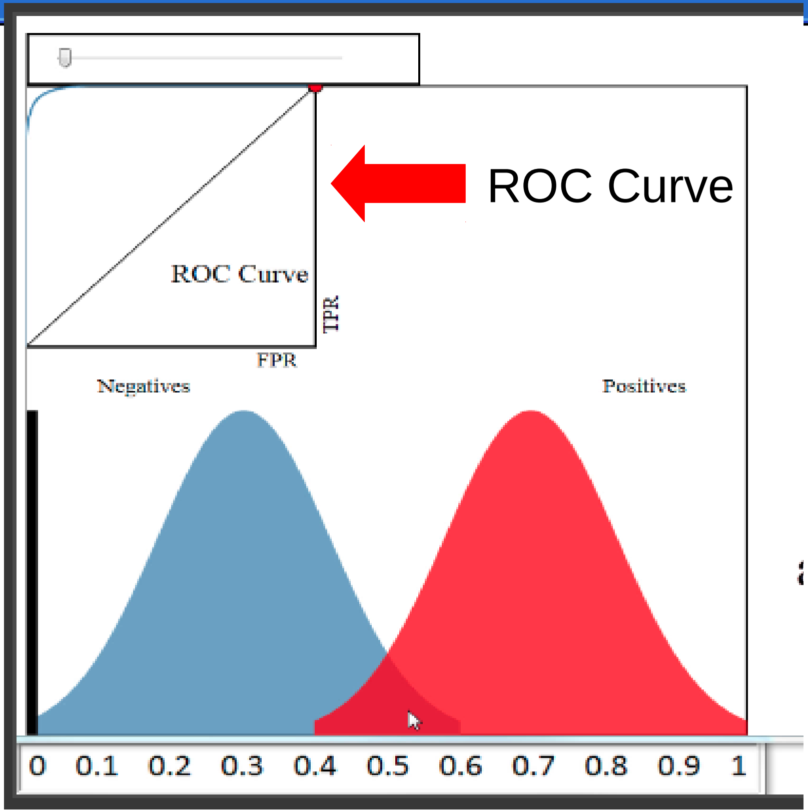
Example from RF testing:
TPR = 0.989559
FPR = 0.051898

Predicition

Actual Class

	Class 0	Class 1
Class 0	**16734** (True Neg. TN)	**916** (False Pos. FP)
class1	**337** (False Neg. FN)	**31940** (True Pos. TP)

Example of ROC curve



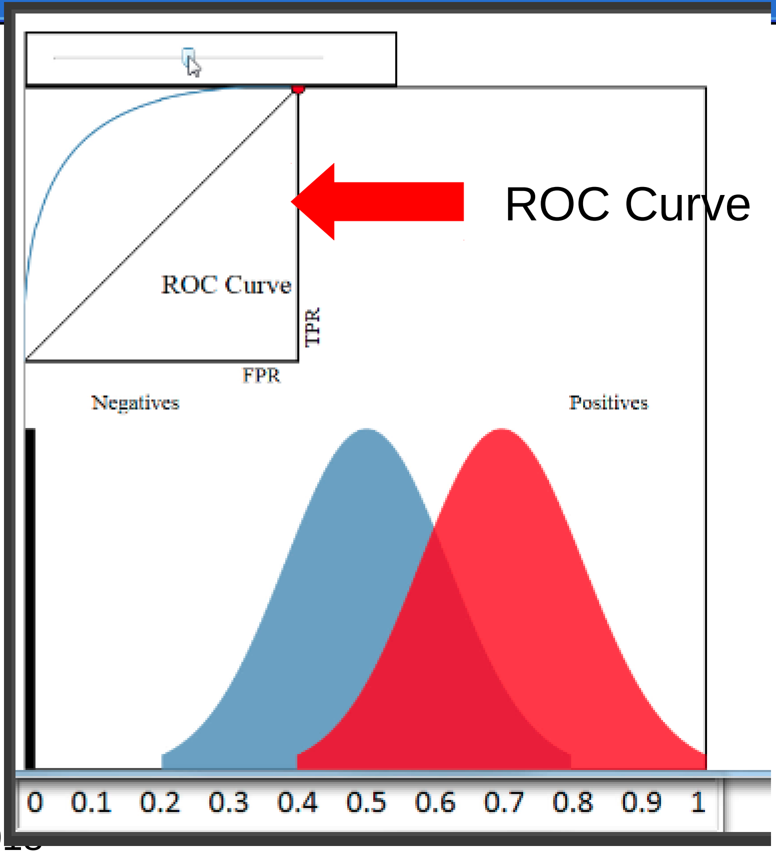
Let Blue = Class 0
Red = Class 1

Y axis for number of objects
X axis for prediction probabilities

Area Under the Curve (AUC) AUC provides an aggregate measure of performance across all possible
classification thresholds.AUC ranges in value from 0 to 1.

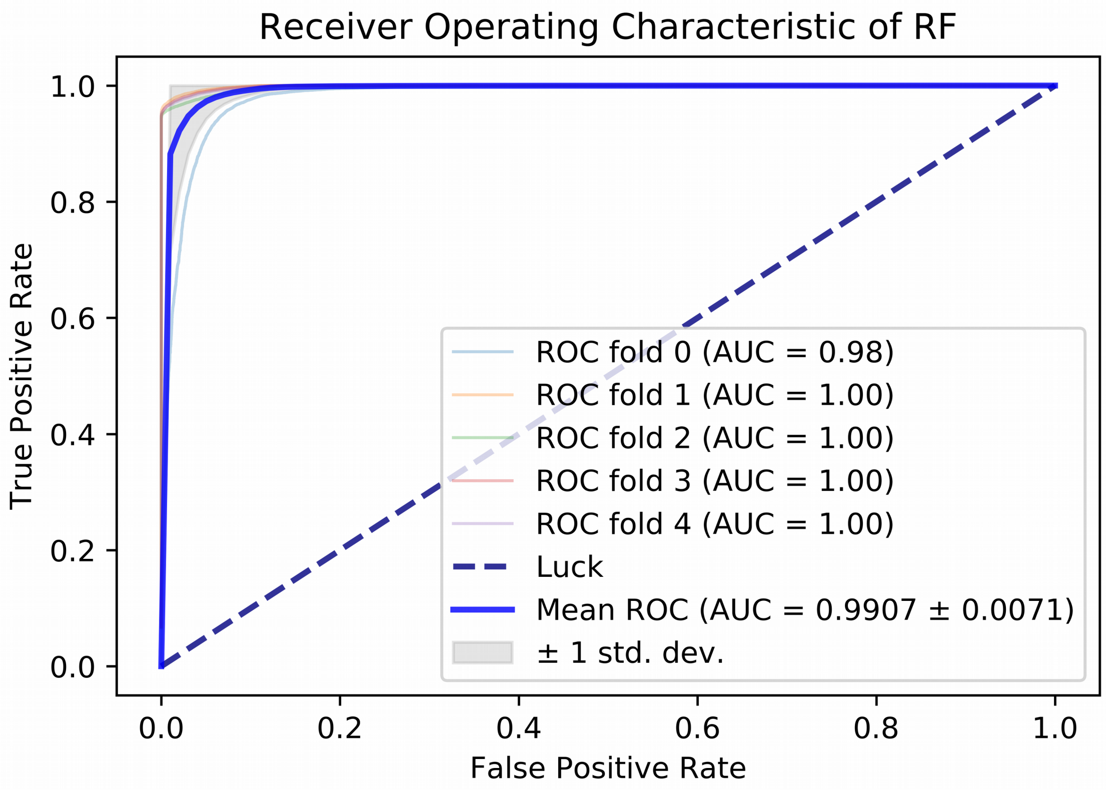
AUC = 0.0 means 100% wrong
AUC = 1.0 means 100% right

Example of ROC Curve



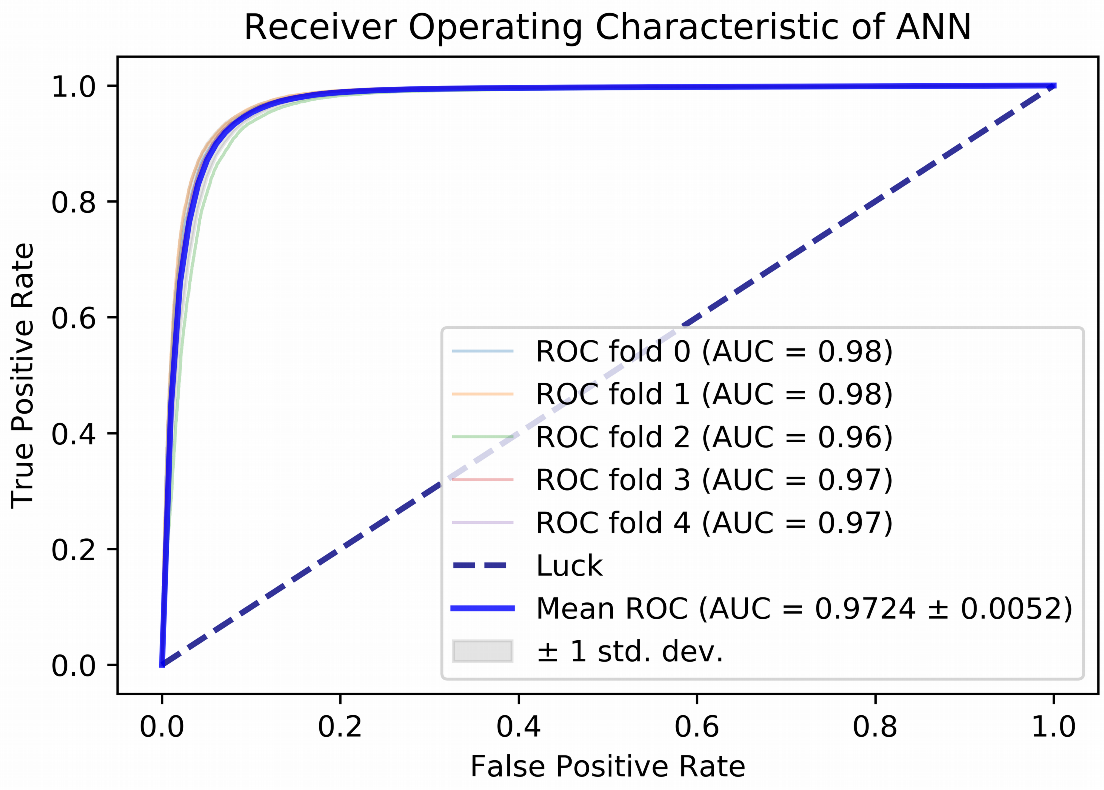
https://www.dataschool.io/roc-curves-and-auc-explained/

ROC of RF



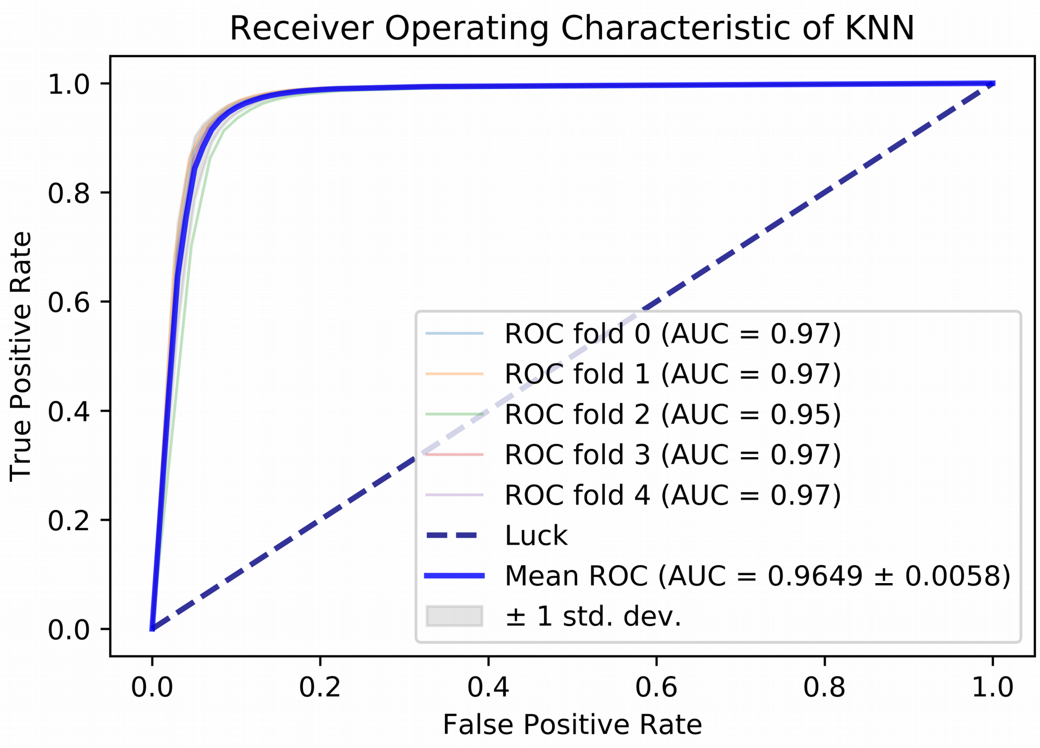
Receiver Operating Characteristic of RF
True Positive Rate
False Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
ROC fold 0 (AUC = 0.98)
ROC fold 1 (AUC = 1.00)
ROC fold 2 (AUC = 1.00)
ROC fold 3 (AUC = 1.00)
ROC fold 4 (AUC = 1.00)
Luck
Mean ROC (AUC = 0.9907 ± 0.0071)
± 1 std. dev.

ROC of ANN



Receiver Operating Characteristic of ANN
True Positive Rate
False Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
ROC fold 0 (AUC = 0.98)
ROC fold 1 (AUC = 0.98)
ROC fold 2 (AUC = 0.96)
ROC fold 3 (AUC = 0.97)
ROC fold 4 (AUC = 0.97)
Luck
Mean ROC (AUC = 0.9724 ± 0.0052)
± 1 std. dev.

ROC of KNN



Receiver Operating Characteristic of KNN
True Positive Rate
False Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
ROC fold 0 (AUC = 0.97)
ROC fold 1 (AUC = 0.97)
ROC fold 2 (AUC = 0.95)
ROC fold 3 (AUC = 0.97)
ROC fold 4 (AUC = 0.97)
Luck
Mean ROC (AUC = 0.9649 ± 0.0058)
± 1 std. dev.

Thank you for your attention