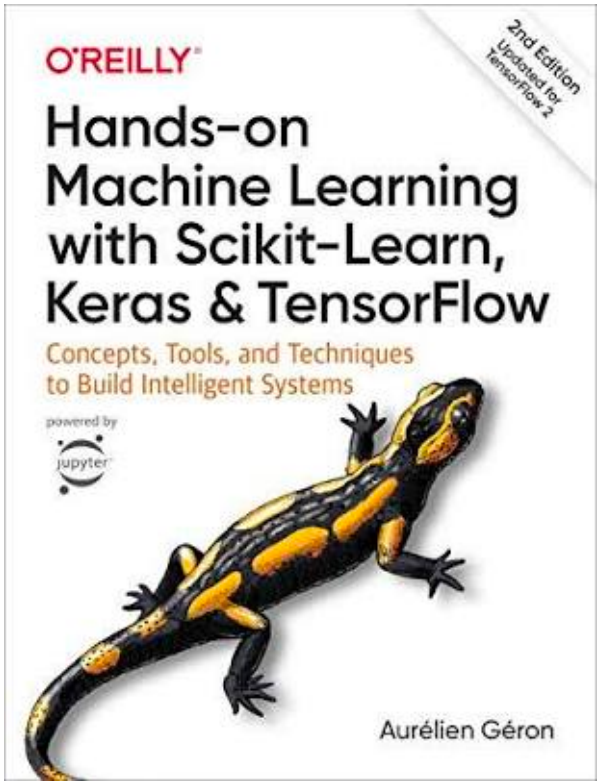
Classification of Astronomical Data Sets with scikit-learn

Eve Chase

Paper: Villar et al. 2019 (arXiv: 1905.07422)

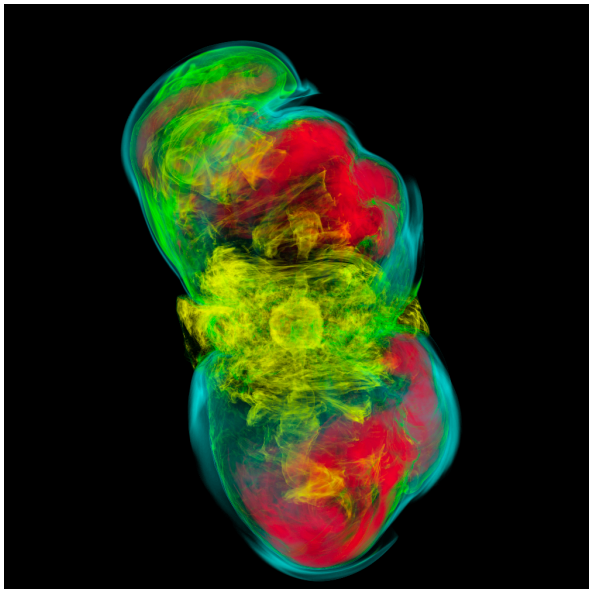
scikit-learn

- Very common Python package for machine learning
- Pro: clear documentation
- Pro: easy to switch between several ML methods
- Con: often feels like a black box
- Con: only includes a basic implementation of neural networks



Supernovae

- Exploding stars!
- Split into many different classes:
  - Type Ia: binary star merger where at least one star is a white dwarf
  - Core-collapse SNe: explosion of a single star



Supernova Classes

- Type Ia: binary star merger where at least one star is a white dwarf
- Core-collapse SNe: explosion of a single star
  - Type IIn: hydrogen with narrow spectral lines
  - Type IIP/IIL: hydrogen with no observable narrow spectral lines
  - Type Ib/Ic: no hydrogen features and weak/no silicon spectral lines
  - Type I superluminous supernova (SLSN): supernovae > 10 times brighter than typical supernova

Goal: Determine Supernova Class with few observations

- Upcoming observatories will lead to roughly **1 million supernovae detected each year**
- Impossible to devote large number of observational resources needed to carefully classify each supernova
- Use machine learning to quickly determine a supernova's class from a few photometric observations, without wasting valuable spectroscopic resources

Supernova Photometric Classification Pipelines Trained on Spectroscopically Classified Supernovae from the Pan-STARRS1 Medium-Deep Survey

V. A. Villar,[1] E. Berger,[1] G. Miller,[1] R. Chornock,[2] A. Rest,[3] D. O. Jones,[4] M. R. Drout,[5] R. J. Foley,[6] R. Kirshner,[1,7] R. Lunnan,[8] E. Magnier,[9] D. Milisavljevic,[10] N. Sanders,[11] and D. Scolnic[12]

- Presents 24 different classification pipelines to determine a supernova's class based on a few observations

- All 24 pipelines involve out-of-the-box and easy to implement scikit-learn classifiers

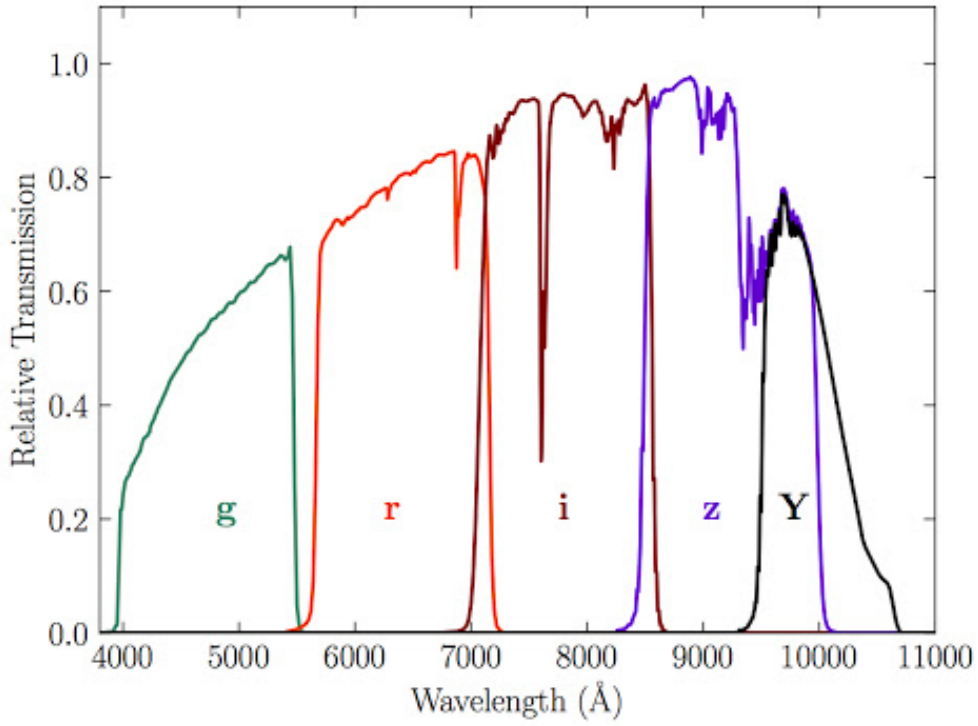
The Data

- Set of 513 well-studied supernovae that have been confidently classified
  - Type Ia: 357
  - Type IIn: 25
  - Type IIP/IIL: 93
  - Type Ib/Ic: 21
  - SLSNe: 17

Data imbalance will bias machine learning classification results.

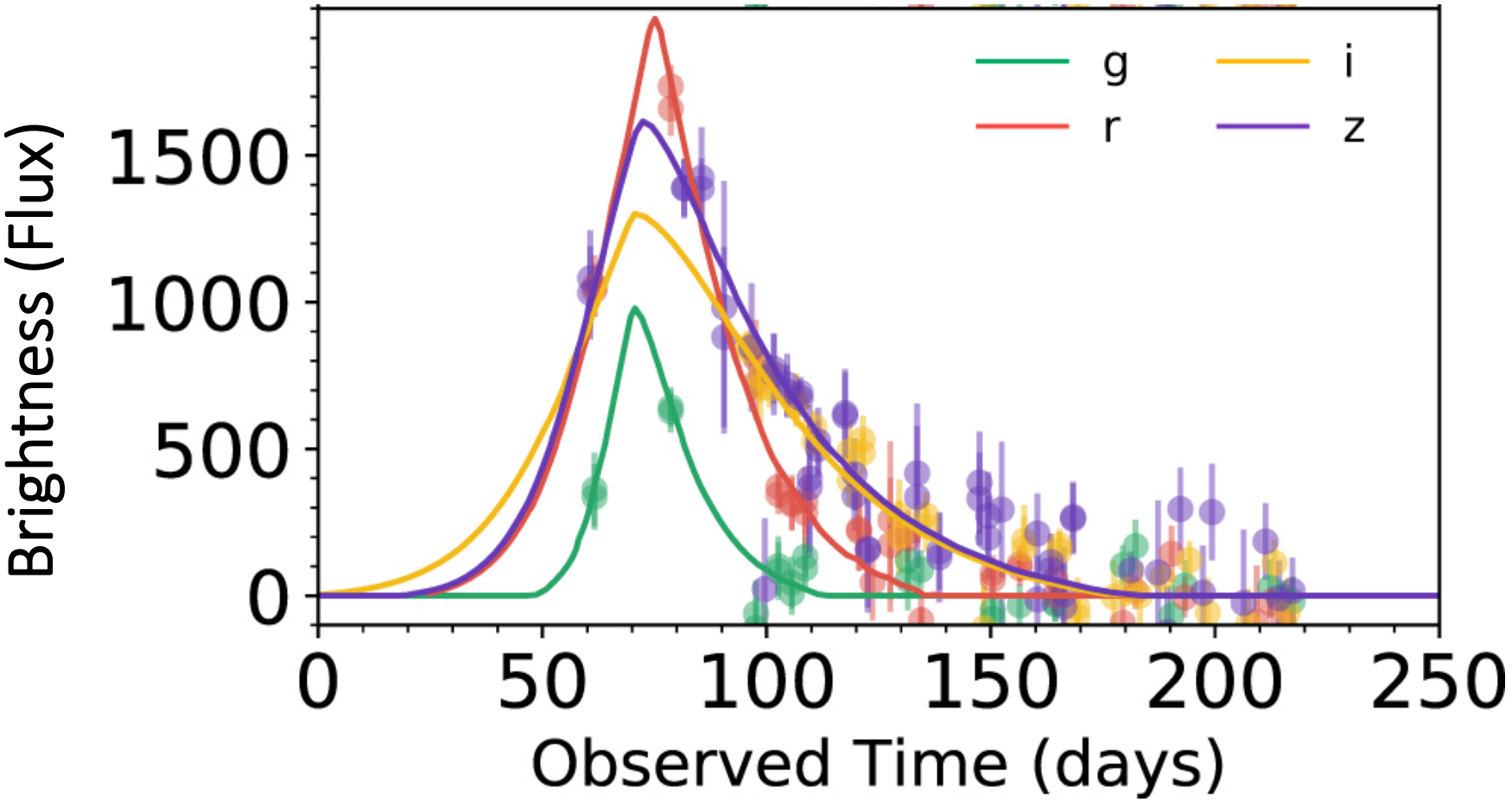
The Data

- Set of observations associated with each supernova
- Time of observation
- Brightness (magnitude) of observation
- Band of observation



The Data: Light Curves

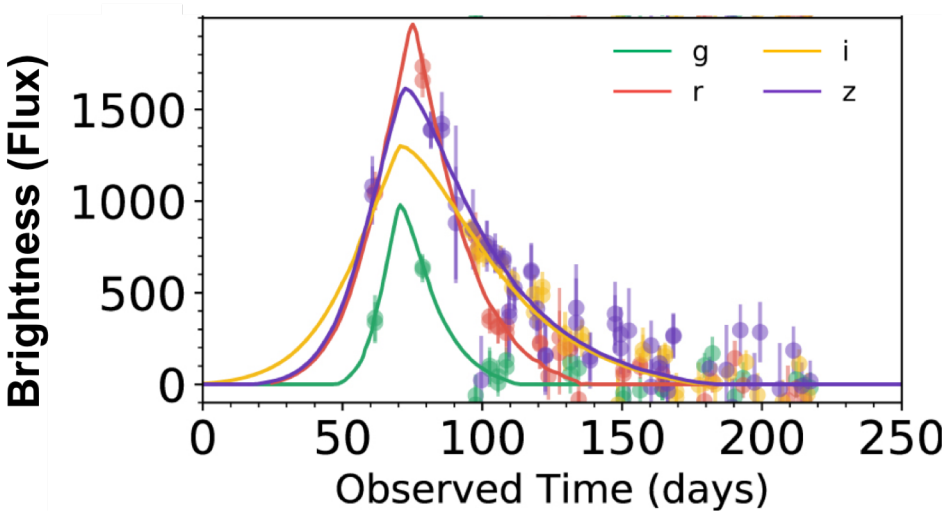
Adapted from
Villar+ 2019
arXiv: 1905.07422



The Data

Villar+ 2019
arXiv: 1905.07422

- Set of light curves associated with all 513 supernovae
- Light curves are fit to an analytic model governed by seven parameters
- Supernovae scaled to assume that they all come from the same distance



Parameter	Description
τ_{rise} (days)	Rise Time
τ_{fall} (days)	Decline Time
t_0 (MJD)	"Start" Time
A	Amplitude
β (flux/day)	Plateau slope
c (flux)	Baseline Flux
γ (days)	Plateau duration

$$F = \begin{cases} \frac{A+\beta(t-t_0)}{1+e^{-(t-t_0)/\tau_{\text{rise}}}} & t < t_1 \\ \frac{(A+\beta(t_1-t_0))e^{-(t-t_1)/\tau_{\text{fall}}}}{1+e^{-(t-t_0)/\tau_{\text{rise}}}} & t \geq t_1 \end{cases}$$

$$\gamma \equiv t_1 - t_0$$

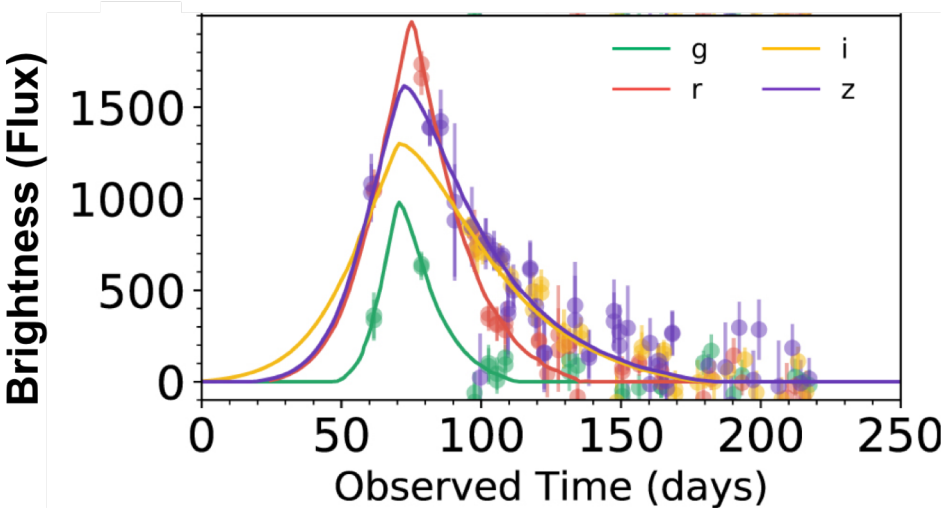
Feature Selection

Selecting subset of variables and data points to use in machine learning model construction

Option 1: Model Parameters (M)

- Use seven parameters from the analytic light curve fits for input
- Total: 28 input parameters

Parameter	Description
$\tau_{\rm rise}$ (days)	Rise Time
$\tau_{\rm fall}$ (days)	Decline Time
t_0 (MJD)	"Start" Time
A	Amplitude
β (flux/day)	Plateau slope
c (flux)	Baseline Flux
γ (days)	Plateau duration

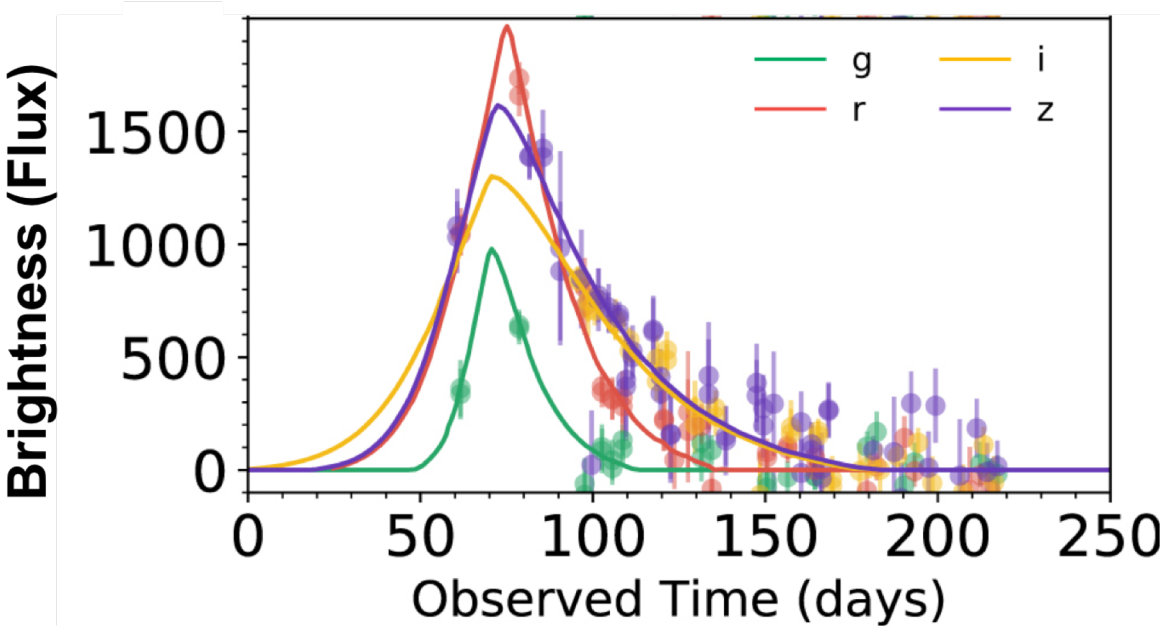


Villar+ 2019
arXiv: 1905.07422

Option 2: Light Curve (LC)

Villar+ 2019
arXiv: 1905.07422

- Directly input brightness data
- Consider 10 observations in each filter
- Total: 40 input parameters



Option 3: Hand-Select (HS)

- For each filter, take four parameters
  - Peak magnitude
  - Time it takes the magnitude to dim by 1 mag
  - Time it takes the magnitude to dim by 2 mag
  - Time it takes the magnitude to dim by 3 mag
- Total: 16 input parameters

Option 4: Principal Component Analysis (PCA)

- Identify a linear combination of parameters that best capture the variance in the data

$$m_{z,t_0}, m_{g,t_1}, m_{z,t_2}, m_{g,t_3},$$

$$a_0 m_{z,t_0} + a_1 m_{g,t_1} + a_2 m_{z,t_2} + a_3 m_{g,t_3}$$

Option 4: Principal Component Analysis (PCA)

- Identify a linear combination of parameters that best capture the variance in the data

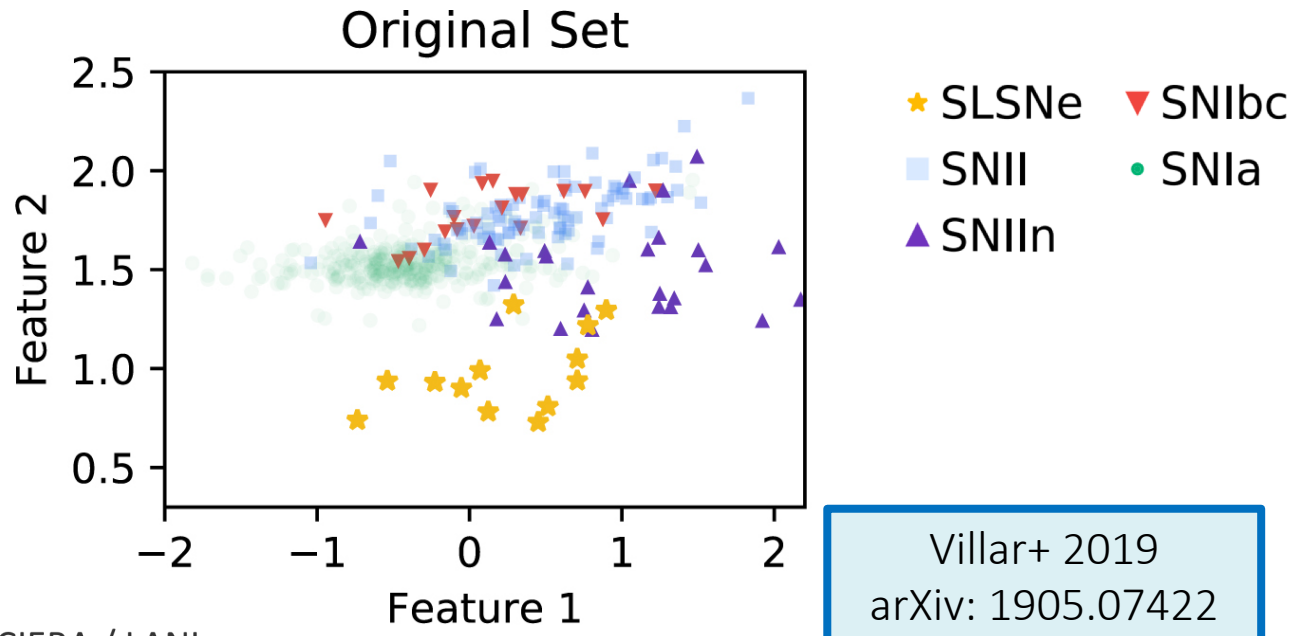
$$m_{z,t_0}, m_{g,t_1}, m_{z,t_2}, m_{g,t_3},$$

$$a_0 m_{z,t_0} + a_1 m_{g,t_1} + a_2 m_{z,t_2} + a_3 m_{g,t_3}$$

- Repeat. Find another (orthogonal) linear combination of parameters that does the next-best job of capturing variance in the data

Option 4: Principal Component Analysis (PCA)

- Find the six linear combinations of parameters that best separate the data set (six “principal components”) for each filter
- Total: 24 inputs



Villar+ 2019
arXiv: 1905.07422

Option 4: Principal Component Analysis (PCA)

- Common feature selection method used in machine learning

- Improves computational efficiency by cutting down on feature size

scikit-learn:
sklearn.decomposition.PCA

Data Augmentation

Leveraging existing data to artificially increase the size of the training set

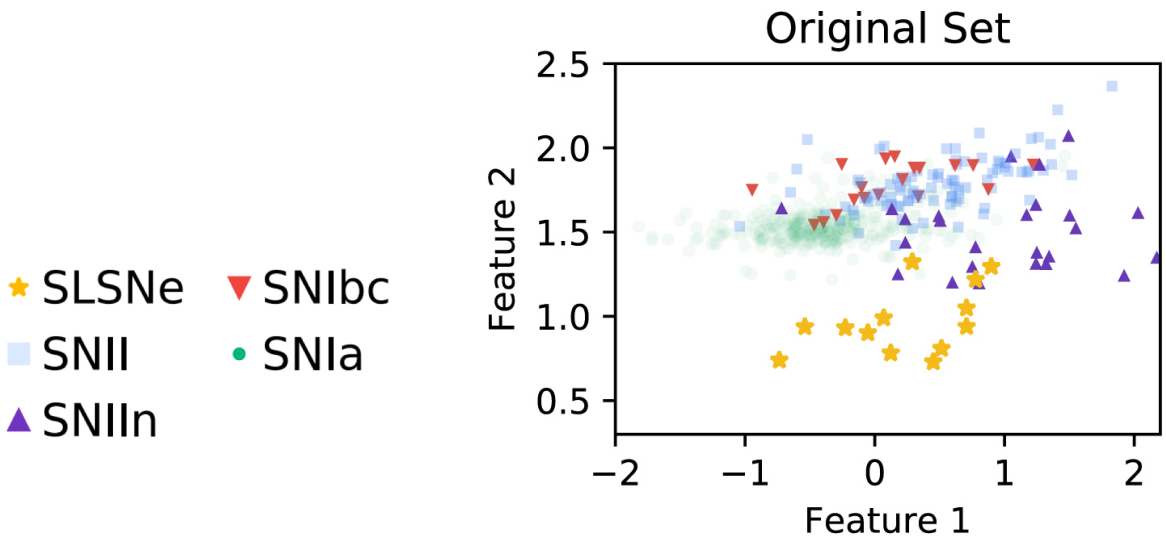
Data Augmentation

- Input dataset
  - Type Ia: 357
  - Type IIn: 25
  - Type IIP/IIL: 93
  - Type Ib/Ic: 21
  - SLSNe: 17

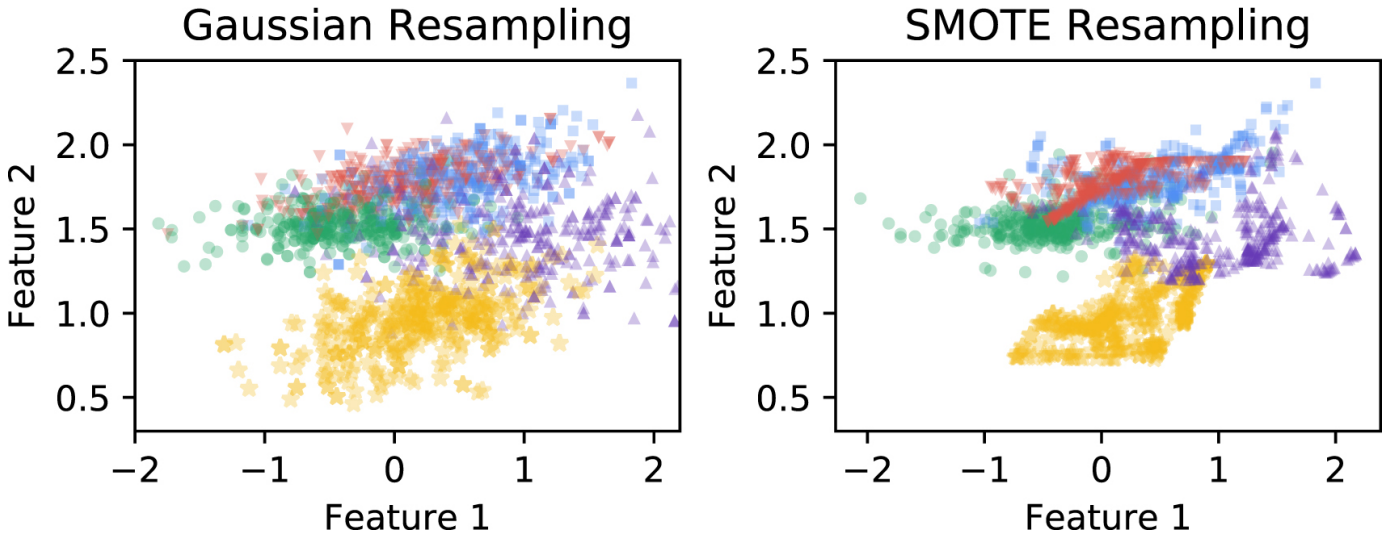
Data imbalance will bias machine learning classification results.

Data Augmentation

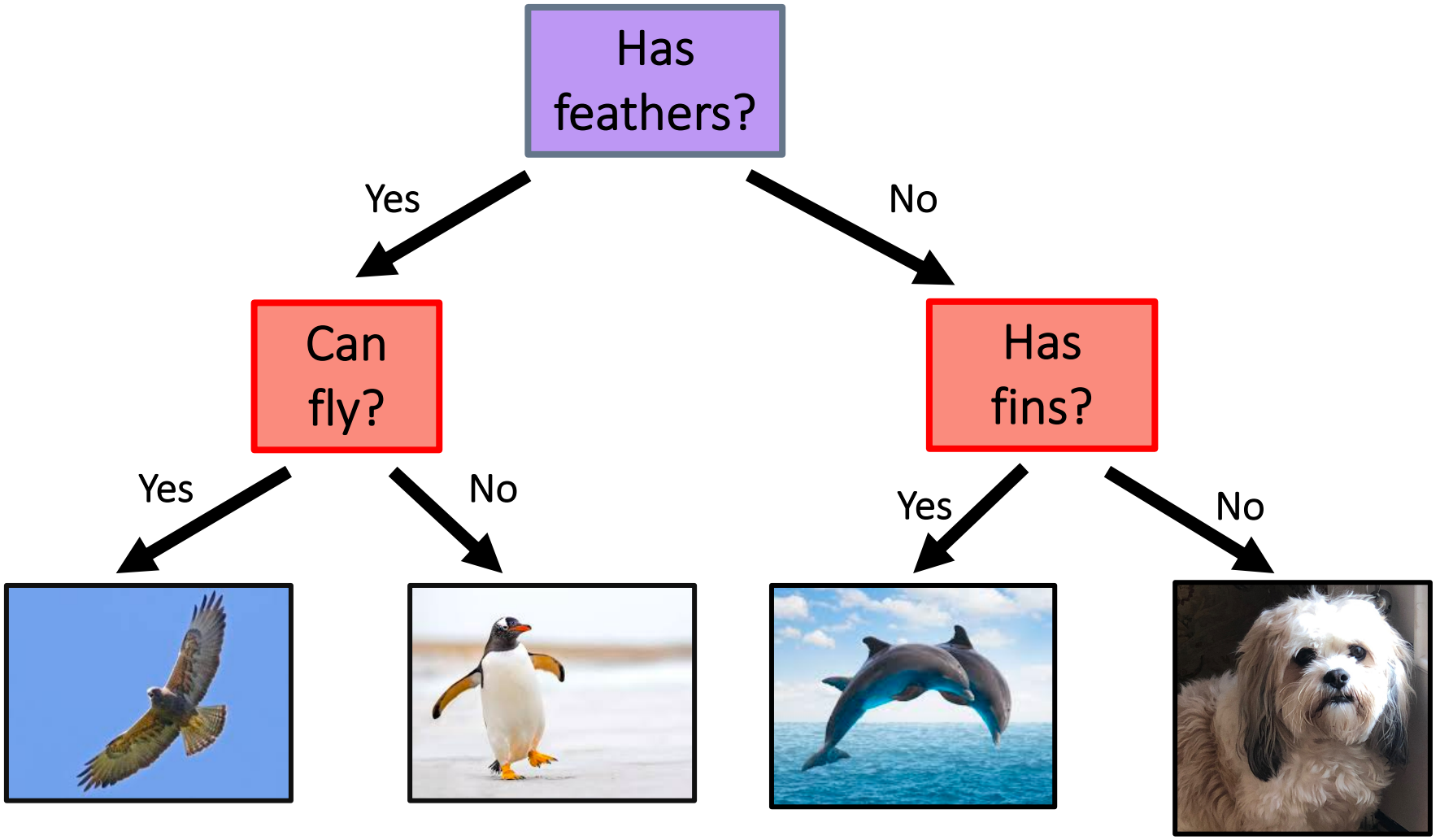
Villar+ 2019
arXiv: 1905.07422



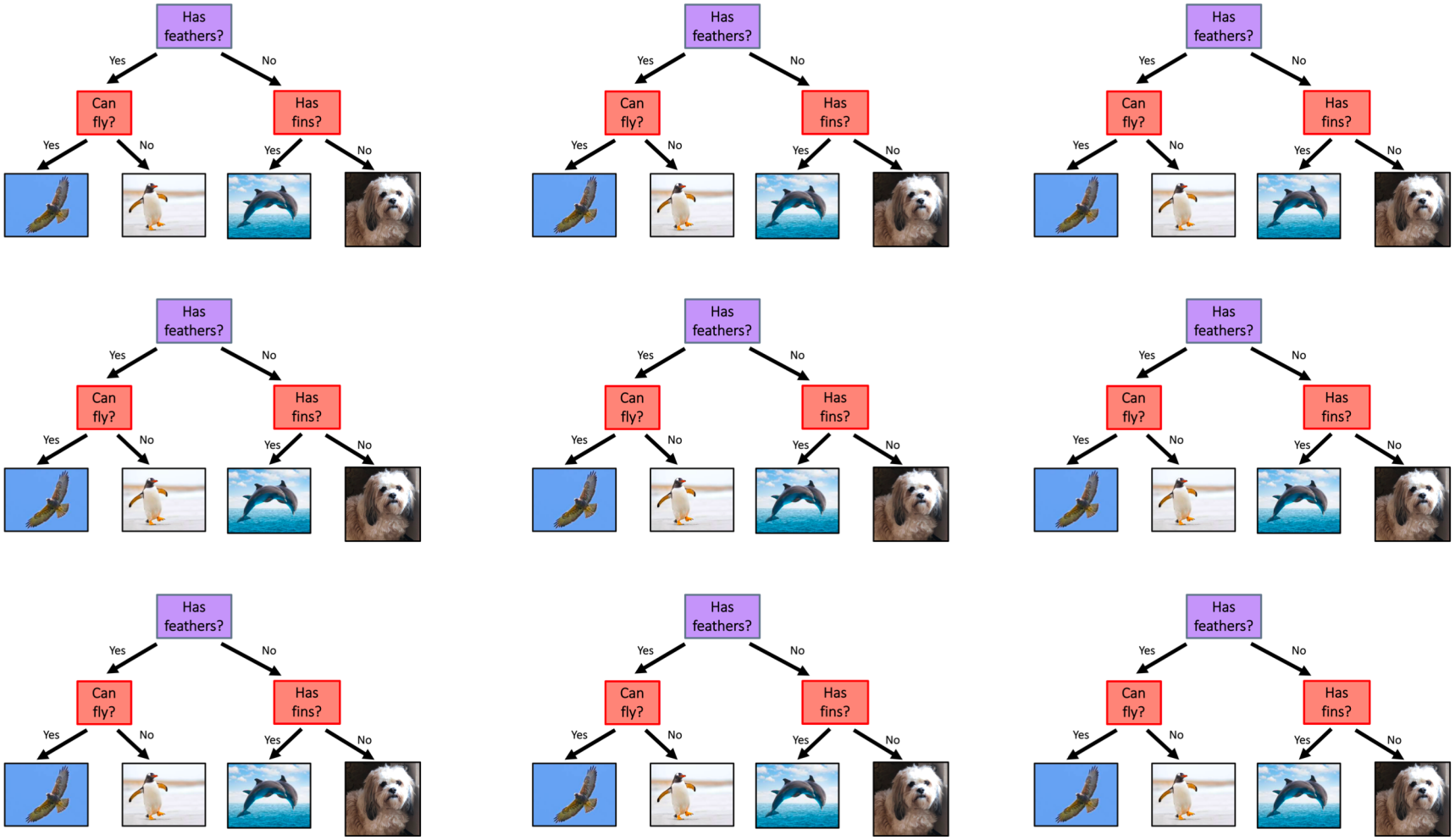
imbalanced-learn:
https://github.com/scikit-learn-contrib/imbalanced-learn



Random Forest (RF)



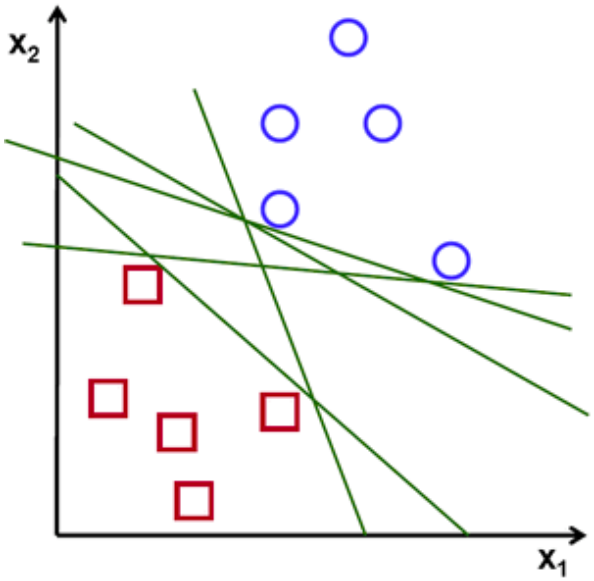
Random Forest (RF)

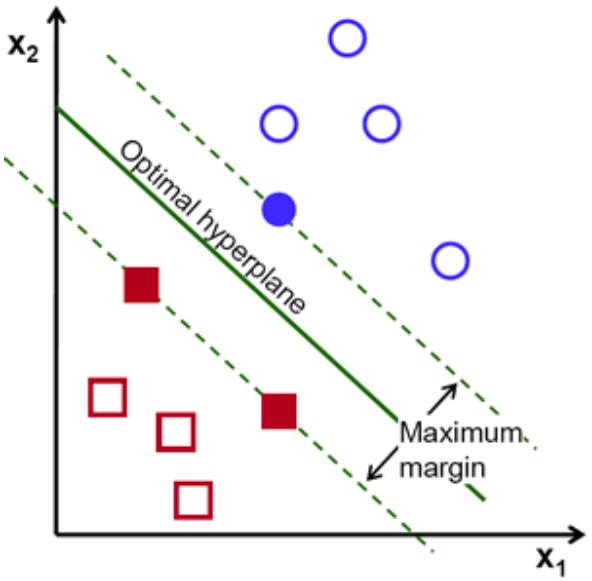


scikit-learn: sklearn.ensemble.RandomForestClassifier

Support Vector Machines (SVM)

- Find optimal hyperplane to separate data

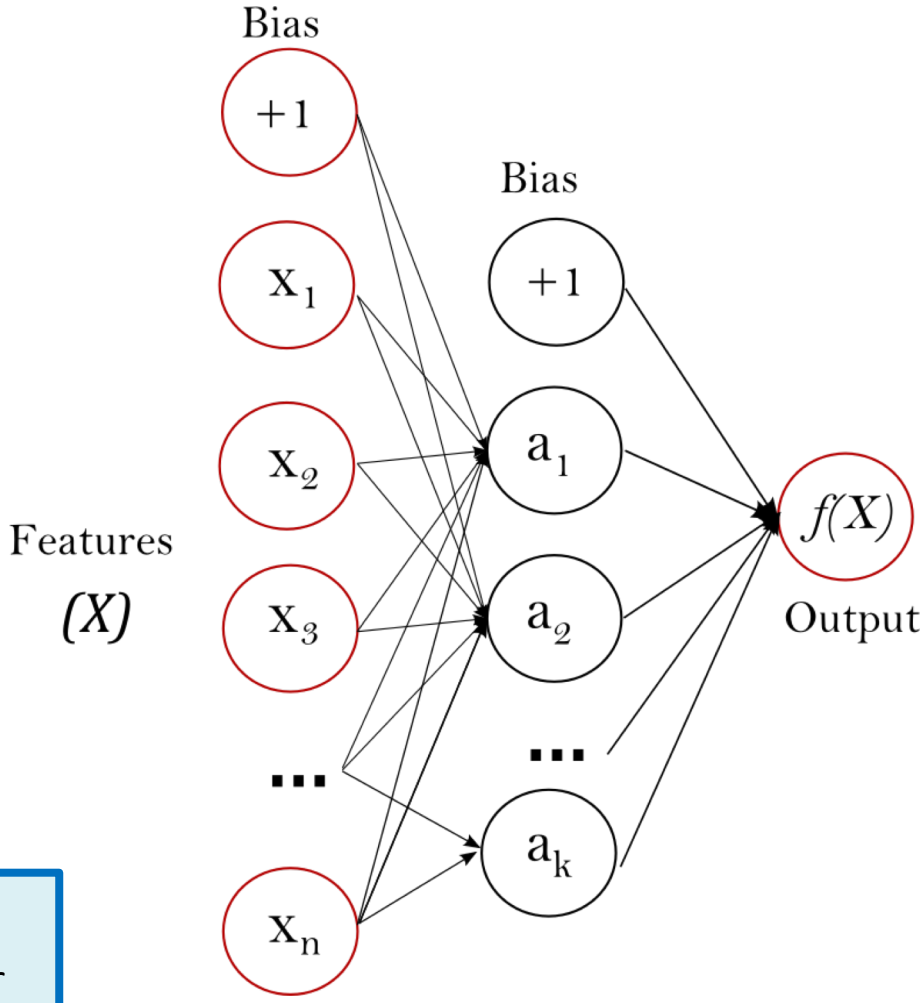




scikit-learn: sklearn.svm.SVC

Multi-Layer Perceptron (MLP)

- Very fancy name for a very normal neural network
- "Fully connected": each neuron is connected with every neuron in the previous layer



scikit-learn:
sklearn.neural_network.MLPClassifier

24 Classification Pipelines

- Three classification models
  - Random Forest (RF)
  - Support Vector Machine (S)
  - Neural Network (NN)
- Four feature selection methods
  - Model Parameters (M)
  - Light Curve Data (LC)
  - Hand-Selected Features (HS)
  - Principal Component Analysis (PCA)
- Two data augmentation schemes:
  - SMOTE Resampling (S)
  - Gaussian Resampling (G)

How do we determine how well each pipeline performs on the data?

Accuracy: Potentially Misleading

- Split into testing and training sets
- Train the classifier on the training set and evaluate performance on the testing set

$$\frac{\text{True Positive} + \text{True Negative}}{\text{Total Samples}}$$

- Accuracy: of all the events in the testing set, how many are correctly labeled as either in the class or not in the class

Purity (a.k.a. Precision)

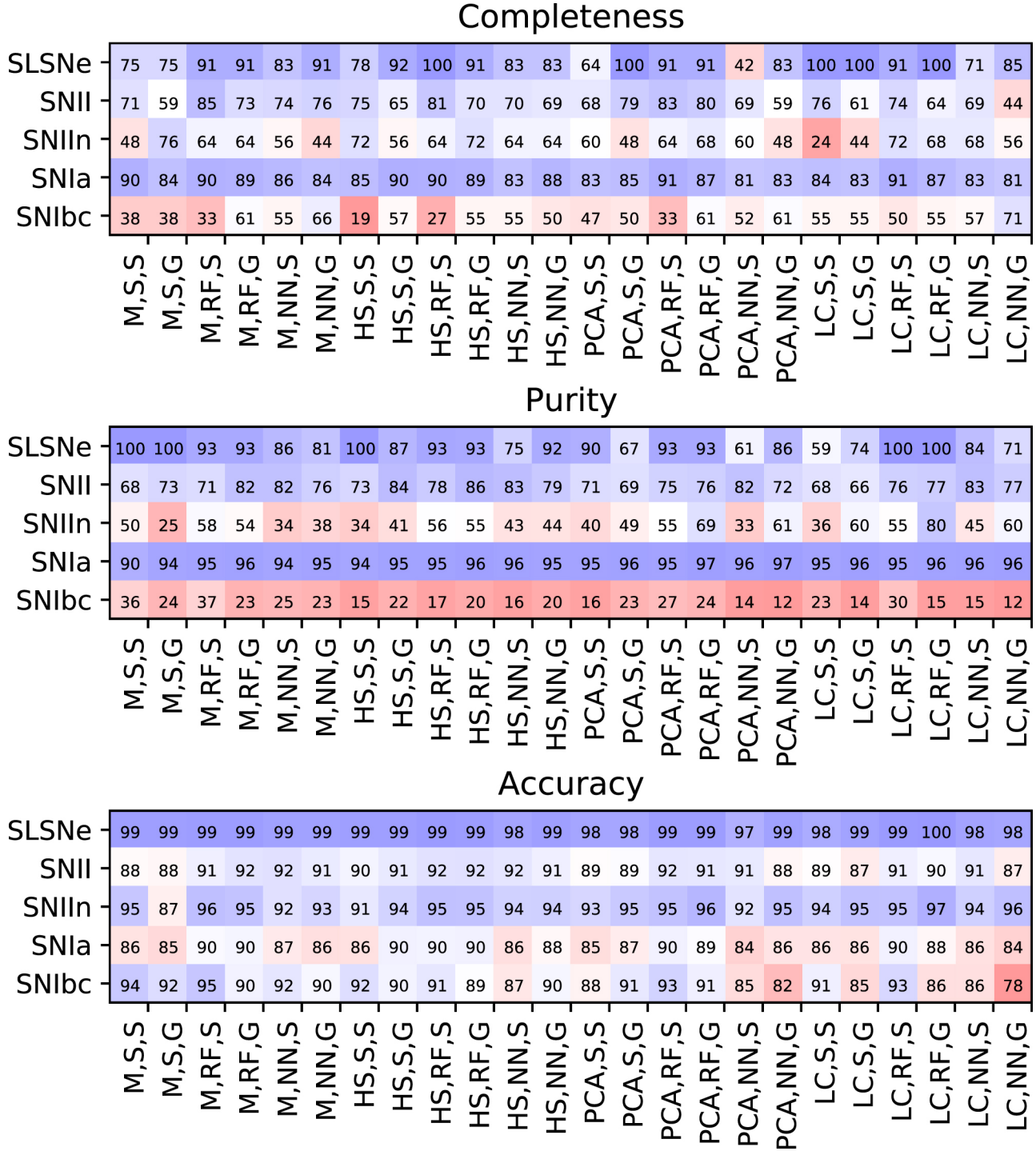
- Take all events **predicted** to belong to a class and count the percentage that are correctly identified

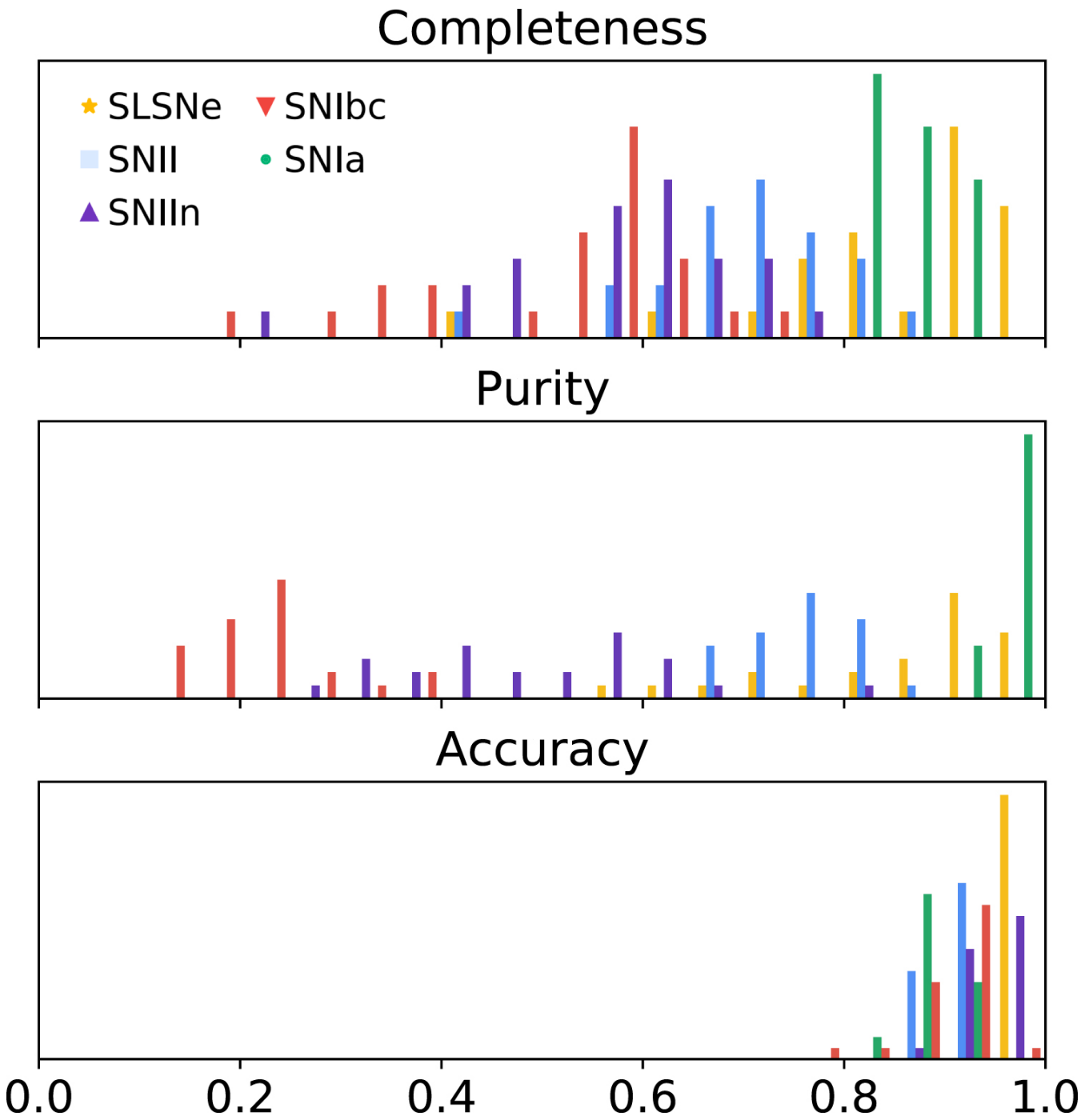
$$\frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

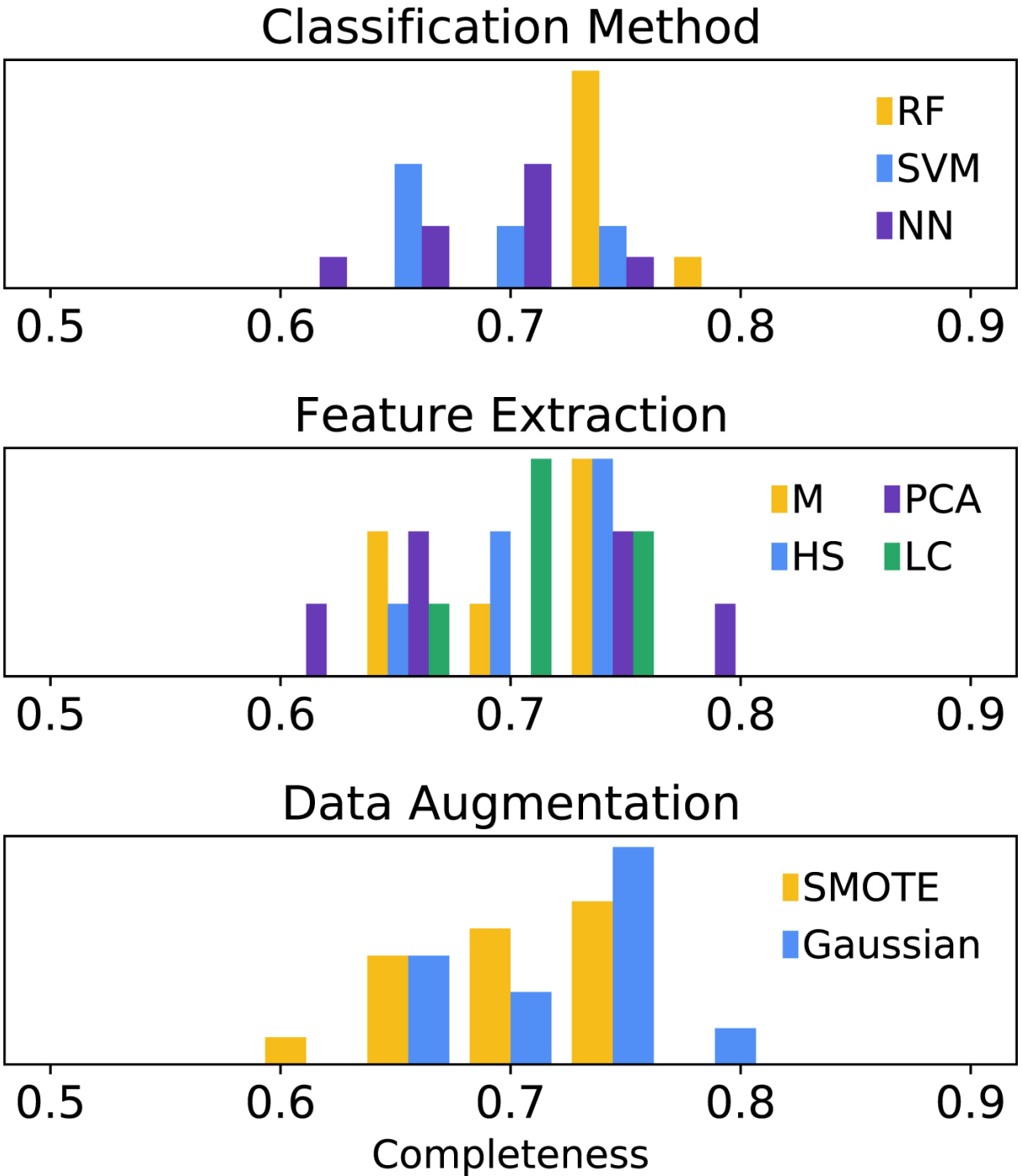
Completeness (a.k.a. Recall)

- Take all events **with known labels** corresponding to a given class and count the percentage that are correctly identified

$$\frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$


Completeness
SLSNe 75 75 91 91 83 91 78 92 100 91 83 83 64 100 91 91 42 83 100 100 91 100 71 85
SNII 71 59 85 73 74 76 75 65 81 70 70 69 68 79 83 80 69 59 76 61 74 64 69 44
SNIIn 48 76 64 64 56 44 72 56 64 72 64 64 60 48 64 68 60 48 24 44 72 68 68 56
SNIa 90 84 90 89 86 84 85 90 90 89 83 88 83 85 91 87 81 83 84 83 91 87 83 81
SNIbc 38 38 33 61 55 66 19 57 27 55 55 50 47 50 33 61 52 61 55 55 50 55 57 71
M,S,S M,S,G M,RF,S M,RF,G M,NN,S M,NN,G HS,S,S HS,S,G HS,RF,S HS,RF,G HS,NN,S HS,NN,G PCA,S,S PCA,S,G PCA,RF,S PCA,RF,G PCA,NN,S PCA,NN,G LC,S,S LC,S,G LC,RF,S LC,RF,G LC,NN,S LC,NN,G
Purity
SLSNe 100 100 93 93 86 81 100 87 93 93 75 92 90 67 93 93 61 86 59 74 100 100 84 71
SNII 68 73 71 82 82 76 73 84 78 86 83 79 71 69 75 76 82 72 68 66 76 77 83 77
SNIIn 50 25 58 54 34 38 34 41 56 55 43 44 40 49 55 69 33 61 36 60 55 80 45 60
SNIa 90 94 95 96 94 95 94 95 95 96 96 95 95 96 95 97 96 97 95 96 95 96 96 96
SNIbc 36 24 37 23 25 23 15 22 17 20 16 20 16 23 27 24 14 12 23 14 30 15 15 12
M,S,S M,S,G M,RF,S M,RF,G M,NN,S M,NN,G HS,S,S HS,S,G HS,RF,S HS,RF,G HS,NN,S HS,NN,G PCA,S,S PCA,S,G PCA,RF,S PCA,RF,G PCA,NN,S PCA,NN,G LC,S,S LC,S,G LC,RF,S LC,RF,G LC,NN,S LC,NN,G
Accuracy
SLSNe 99 99 99 99 99 99 99 99 99 99 98 99 98 98 99 99 97 99 98 99 99 100 98 98
SNII 88 88 91 92 92 91 90 91 92 92 92 91 89 89 92 91 91 88 89 87 91 90 91 87
SNIIn 95 87 96 95 92 93 91 94 95 95 94 94 93 95 95 96 92 95 94 95 95 97 94 96
SNIa 86 85 90 90 87 86 86 90 90 90 86 88 85 87 90 89 84 86 86 86 90 88 86 84
SNIbc 94 92 95 90 92 90 92 90 91 89 87 90 88 91 93 91 85 82 91 85 93 86 86 78
M,S,S M,S,G M,RF,S M,RF,G M,NN,S M,NN,G HS,S,S HS,S,G HS,RF,S HS,RF,G HS,NN,S HS,NN,G PCA,S,S PCA,S,G PCA,RF,S PCA,RF,G PCA,NN,S PCA,NN,G LC,S,S LC,S,G LC,RF,S LC,RF,G LC,NN,S LC,NN,G


Completeness
SLSNe
SNIbc
SNII
SNIa
SNIIn
Purity
Accuracy
0.0
0.2
0.4
0.6
0.8
1.0


Classification Method
RF
SVM
NN
0.5
0.6
0.7
0.8
0.9
Feature Extraction
M
PCA
HS
LC
0.5
0.6
0.7
0.8
0.9
Data Augmentation
SMOTE
Gaussian
0.5
0.6
0.7
0.8
0.9
Completeness