Astronomy Transients

ASTRAL EIGHT Chirag Rathi , Elise P, Martina Cadiz , Sara Federle , Sherelyn Alejandro TA- Paula Llanos

Project Overview

Transients from ZTF ~ 40,000 objects and ~44 features

Exploratory data analysis

Multiclass supervised classification

Compare the performance of multiple machine learning models



Data from Zwicky Transient Facility

- Observes entire Northern Sky since 2018
- Scans every 2 days in the g, r, and i filters
- 1.2 m Samuel Oschin Schmidt telescope
- Magnitude limit: m_=20.5

Obtained Data for Billions of Sources



Roestel 2021



The Hurdles of Big Data

Supervised Machine Learning can help us analyze our large sample of data with known classifications more efficiently, allowing us to understand our sample even better

Exploratory Data Analysis: Correlation matrix



The matrix shows the correlation among the features

SF

Orange denote strong correlation, whereas darker colors denote no correlation

Exploratory Data Analysis: Pulsators and Binaries

SF



Count Plots for the different classes of pulsators (left panel) and binaries (right panel)

Dealing with Data Imbalances

- Stratified sampling
- Data augmentation techniques (SMOTE, random sampling)

Dimensionality reduction methods:

- Recursive Feature Elimination
- Principal component analysis (PCA)

Data Imbalance:

Method 1: Random oversampling or undersampling

Oversampling: produce exact (and random) copies of minority data

until minority = majority.

Undersampling: randomly remove majority data until majority = minority.

Method 2: SMOTE

Synthetic Minority Oversampling Technique.

Uses k-nearest neighbors to mimic data points in the minority class.

Module(s) used: imblearn.over_sampling.smote, imblearn.oversampling.RandomOverSampler



CR

Dimensionality Reduction Methods:

- METHOD 1: Recursive Feature Elimination
- Too much data is overwhelming.
- Hard to focus on anything.
- RFE to the rescue! This process eliminates unnecessary features by taking smaller subsets of features and applying it to the training data.

CR

Module(s) used: sklearn.feature_selection.RFE, sklearn.ensemble.RandomForestRegressor

Non-pulsato	0	3	65	0	6612	0	2	145	0	1059	337	4310	1018	35	- 6000	von-pulsators	5020	0	2	47	0	4939	0	1	129	0
Delta Scu	6759	0	0	0	0	6759	0	0	0	0	6759	0	0	0	- 5000	Delta Scu	0	5069	0	0	0	0	5069	0	0	0
Cepheids	0	43	3	0	30	1	6	20	0	0	2	53	2	0	- 4000 - 3000	Cepheids	10	0	32	1	0	22	1	5	15	0
Lyrae	D O	1		0	450	6	8	355	0	12	54	568	184	1	- 2000	RR Lyrae	89	0	1	525	0	330	3	8	274	0
۶ <u>م</u>	0	0	0	0	5	0	0	1	0	2	0	1	2	1	- 1000	ΓΡΛ	4	0	0	0	0	3	0	0	1	0
		LR					SVM					GBC			-0				IR					SVM		
on-pulsators	17 1737	0	13	2	6680	0	1	78	0	6675	0	4	79	1	- 6000	on-pulsators	3091	1858	0	120	0	5010	0	0	59	0
Delta Scu	2 5137	0	0	0	0	6759	0	0	0	0	6759	0	0	0	- 5000	Delta Scu N	918	4151	0	0	0	0	5069	0	0	0
Cepheids	22	0	0	0	9	1	13	34	0	33	0	10	13	1	- 4000 - 3000	Cepheids	30	8	0	5	0	7	1	12	23	0
SC Parage	3 276	0	20	0	165	5	1	648	0	118	0	1		0	- 2000	RR Lyrae	359	187	0	69	0	131	4	2	478	0
۶ <u>م</u>	0	0	0	0	6	0	0	0	0	6	0	0	0	0	- 1000	ΓЪΛ	3	1	0	0	0	4	0	0	0	0
pulsators	Delta Scu	Cepheids	RR Lyrae	ΛΠ	-pulsators	Delta Scu	Cepheids	RR Lyrae	ΓPΛ	-pulsators	Delta Scu	Cepheids	RR Lyrae	ΛJ	- 0		-pulsators	Delta Scu	Cepheids	RR Lyrae	Λď	-pulsators	Delta Scu	Cepheids	RR Lyrae	Λď

Test size = 20%

RFE + SMOTE

Results:

Test size = 15%

787 258 3257

421 140

GBC

0 0

506

0

Cepheids RR Lyrae LPV

44

0 5069 0 0 0

5009

0

109

742

4000

3000

2000

- 1000

4000

3000

2000

- 1000

			RF		
Non-pulsators	3338	0	1 1	41	0
Delta Scu		3379	0		0
Cepheids		0	3379	0	0
KK Lyrae				3380	0
N-1					3379

_			SVM		
	3215		33	123	
	0	3379	0		
	42		3226	111	
	275		212	2890	0
9	0				3379
	on-pulsators	Delta Scu	Cepheids	RR Lyrae	LPV

KNC

3379

0

0

3379

3357

2895

3379

			LR		
Non-pulsators	2335	17	195	699	134
Delta Scu	1709		823	847	0
Cepheids	453	104	919		298
RR Lyrae	873	62	783	1584	78
ΓPΛ			460		2919
	n-pulsators	Delta Scu	Cepheids	RR Lyrae	ΛďΊ

-			GNB		_	
	475	140		400	49	- 3000
		3379	0			- 2500
	16	52	3241	54	16	- 2000 - 1500
	45	230	2340	750	15	- 1000
	130	0	233	463	2553	- 500

Test size = 10%

CR

Comparing estimators' performances



Scoring technique = accuracy Folds = division of data into n folds Cross validation score = predictor of performance. CR

Winner = GBC and RF

NOTE: Stratified folding!

Unsupervised learning: TSNE



Perplexity = 30



Perplexity = 50



Perplexity = 100



CR

Random Forest vs XGBoost

Random Forest

Fits a certain number of decision tree classifiers on various subsamples of the data

Improve predictive accuracy and control overfitting

Confusion matrix (C)

Evaluates the accuracy of the classification

C_{ii}= number of

XGBoost

SF

Implementation of Gradient Boosting

Fit n classes of regression trees on the negative gradient of the loss function

Allows for optimization of differentiable loss function

RESULTS Machine Learning performance: unbalanced case



SF

Confusion matrix for the XGBoost (*left panel*) and Random Forest Classifier (*right panel*) in the unbalanced case for the binaries

Machine Learning performance: balanced case

SF



Confusion matrix for the XGBoost (*left panel*) and the Random Forest Classifier (*right panel*) in the balanced case for the binaries classification. In both cases the test size was 30%. The classes are W Uma (0), RS CVn (1)

Machine learning: classification reports

AUDUUSL	report	-		
MAE (Mean-Abs	olute-Error)	: 0.04938	27160493827	71
MSE (Mean-Squ	ared-Error);	0.070242	656449553	
RMSE (Root-MS	E): 0.265033	311962011	7	
R2 score: 0.1	882258876112	2128		
	precision	recall	f1-score	support
	procession	recure		Support
Θ	0.97	0.99	0.98	4425
1	0.70	0.65	0.67	221
2	0.67	0.04	0.07	52
accuracy			0.96	4698
macro avg	0.78	0.56	0.58	4698
weighted avg	0.96	0.96	0.95	4698
Random	Forest report	t		
Random MAE (Mean-Abs	Forest report olute-Error)	t : 0.04938	27160493827	71
MAE (Mean-Abs MSE (Mean-Squ	Forest report olute-Error) ared-Error):	t : 0.04938 0.070242	27160493827 656449553	/1
Random MAE (Mean-Abs MSE (Mean-Squ RMSE (Root-MS	Forest report olute-Error) ared-Error): E): 0.2650333	t : 0.04938 0.070242 311962011	27160493827 656449553 7	71
Random MAE (Mean-Abs MSE (Mean-Squ RMSE (Root-MS R2 score: 0.1	Forest repor olute-Error) ared-Error): E): 0.2650333 8822588761123	t : 0.04938 0.070242 311962011 2128	27160493827 656449553 7	71
Random MAE (Mean-Abs MSE (Mean-Squ RMSE (Root-MS R2 score: 0.1	Forest report olute-Error) ared-Error): E): 0.265033 882258876112 precision	t : 0.04938 0.070242 311962011 2128 recall	27160493823 656449553 7 fl-score	71 support
Random MAE (Mean-Abs MSE (Mean-Squ RMSE (Root-MS R2 score: 0.1	Forest report olute-Error) ared-Error): E): 0.2650333 8822588761122 precision 0.97	t : 0.04938 0.070242 311962011 2128 recall 0.99	27160493827 656449553 7 f1-score 0.98	71 support 4425
Random MAE (Mean-Abs MSE (Mean-Squ RMSE (Root-MS R2 score: 0.1 0 1	Forest report olute-Error): ared-Error): E): 0.2650333 8822588761122 precision 0.97 0.72	t : 0.04938 0.070242 311962011 2128 recall 0.99 0.65	27160493827 656449553 7 f1-score 0.98 0.68	71 support 4425 221
Random MAE (Mean-Abs MSE (Mean-Squ RMSE (Root-MS R2 score: 0.1 0 1 2	Forest report olute-Error): ared-Error): E): 0.2650333 8822588761123 precision 0.97 0.72 1.00	t : 0.04938 0.070242 311962011 2128 recall 0.99 0.65 0.06	27160493827 656449553 7 f1-score 0.98 0.68 0.11	71 support 4425 221 52
AE (Mean-Abs MSE (Mean-Squ RMSE (Root-MS R2 score: 0.1 0 1 2 accuracy	Forest report olute-Error): ared-Error): E): 0.2650333 8822588761123 precision 0.97 0.72 1.00	t : 0.04938 0.070242 311962011 2128 recall 0.99 0.65 0.06	27160493823 656449553 7 f1-score 0.98 0.68 0.11 0.96	71 support 4425 221 52 4698
AE (Mean-Abs MSE (Mean-Squ RMSE (Root-MS R2 score: 0.1 0 1 2 accuracy macro avg	Forest report olute-Error): ared-Error): E): 0.2650333 8822588761123 precision 0.97 0.72 1.00 0.90	t : 0.04938 0.070242 311962011 2128 recall 0.99 0.65 0.06 0.57	27160493823 656449553 7 f1-score 0.98 0.68 0.11 0.96 0.59	71 support 4425 221 52 4698 4698

XGBoost	report	-		
MAE (Mean-Abs	solute-Error)	: 0.02467	92452830188	368
MSE (Mean-Sou	ared-Error)	0 039320	75471698113	34
PMCE (Poot-MC	E) · 0 109204	615055605	66	
	DE/. 0.190294	010900000	00	
R2 score: 0.9	9413230119973	818	<i>c</i> -	
	precision	recall	tl-score	support
Θ	1.00	0.95	0.97	4494
1	0.97	1.00	0.98	4370
2	0.98	1.00	0.99	4386
accuracy			0.98	13250
macro avo	0.98	0.98	0 98	13250
weighted avg	0.98	0.98	0.98	13250
Random	Forest repor	+		
MAE (Mean-Aba	alute-Error)	. 0 02316	9811320754	716
MSE (Moon Sal	ared Error)	0.025004	2206226415	1
PMCE (Peat MC		0.033094	5550220415.	1
RMSE (ROOT-MS	E): 0.18/334	832913266	58	
R2 score: 0.9	476299435293	33		
	precision	recall	f1-score	support
Θ	1.00	0.95	0.97	4494
1	0.97	1.00	0.98	4370
2	0.98	1.00	0.99	4386

0.98

0.98

0.98

0.98

0.98

0.98

0.98

13250

13250

13250

SF

weighted avg

accuracy macro avg

Machine Learning performances: scores and runtimes

	Score	Runtime	Training	Runtime	Prediction
Model					
Random Forest	96.104725		8.162835		0.072484
XGBoost	95.998297		8.334034		0.044336
SVC	94.870158		4.988982		1.064658
KNN	94.061303		0.004444		2.072916
	Score	Runtime	Training	Runtime	Prediction
Model					
XGBoost	98.264151		8.334034		0.044336
Random Forest	98.249057		22.581923		0.274669
KNN	94 075472		0.011730		17 209707

Comparison of different Machine Learning methods applied for the binaries classification for the unbalanced (top panel) and balanced (bottom panel) cases

SF



CNIN architecture used

Deep Learning classification

Aim to use the light curves image set

Data preparation

Train set (8K), test set (4K), validation set (2K)

PCA for dimensionality reduction

• Image resolution 28x28

Training stage

۲

۲

- EarlyStopping w/ 10 of patience
- Dropout

Time consumed for training: 55.541 seconds

RESULTS Deep Learning performance



Confination Maturity fountles CNINI

----- CNN report -----MAE (Mean-Absolute-Error): 0.12378077101718532 MSE (Mean-Squared-Error): 0.17533673943334882 RMSE (Root-MSE): 0.41873230044188 R2 score: 0.746722162385355

	р	recision	recall	f1-score	support
	0	0.90	0.94	0.92	1500
	1	0.94	0.82	0.88	1325
	2	0.88	0.93	0.91	1481
accurac	у			0.90	4306
macro av	g	0.90	0.90	0.90	4306
weighted av	g	0.90	0.90	0.90	4306

CNN report



CONCLUSIONS & FUTURE WORK

RESULTS

- Classified the pulsators successfully using RFE process.
- Classified binaries with different classification methods → Random Forest and XGBoost gave the best results
- The machine learning models presented worked better than the CNN

NEXT STEPS

- Need more data for particular sub-classes of pulsators and binaries to improve the classification
 - Try another data augmentation technique

ACKNOWLEDGMENTS

Thank you to La Serena School For Data Science ! Niharika Sravan and Paula Llanos

REFERENCES

Jan van Roestel, Dmitry A. Duev, Ashish A. Mahabal, Michael W. Coughlin, Przemek Mróz, Kevin Burdge, Andrew Drake, Matthew J. Graham, Lynne Hillenbrand, Eric C. Bellm, Thomas Kupfer, Alexandre Delacroix, C. Fremling, V. Zach Golkhou, David Hale, Russ R. Laher, Frank J. Masci, Reed Riddle, Philippe Rosnet, Ben Rusholme, Roger Smith, Maayane T. Soumagnac, Richard Walters, Thomas A. Prince, & S. R. Kulkarni (2021). The ZTF Source Classification Project. I. Methods and Infrastructure. *The Astronomical Journal*, *161(6)*, *267*.

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

Guillaume Lemaitre, Fernando Nogueira, & Christos K. Aridas (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, *18(17)*, *1-5*.

Hunter, J. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90-95.

Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.

Wes McKinney (2010). Data Structures for Statistical Computing in Python . In *Proceedings of the 9th Python in Science Conference* (pp. 51 - 56).

Michael L. Waskom (2021). seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021.

CREDITS:

This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**