Astronomical Transient Object Recognition With Machine Learning

Diego Gómez Mosquera

Department of Systems and Computer Engineering
Universidad de Los Andes
Colombia

December 2017

Abstract

With the upcoming arrival of new generation of multi-epoch and multi-band (synoptic) astronomical surveys, the study and detection of astronomical variable sources is expected to occur in unprecedented scales. Automating the recognition and classification of transient events, a type of such variable sources, would reduce costs and speed up this process as well as provide scientists information of the universe in various spacial scales [1].

In this context, the current project proposes a new method to recognize and classify such astronomical events using machine learning algorithms. The method proposed consists in filtering out light curves with insufficient data, then oversampling the unbalanced transient classes using the photometric error as a Gaussian noise, and extracting several measurements from the resulting data. These measurements are statistical descriptors of the light curves, as well as coefficients obtained when fitting polynomial curves to the data. Extracted features are then used as input to multiple machine learning models, which automatically learn to detect and classify between different types of transient objects.

State of the art results were obtained by applying the proposed methodology to five different binary and multi-class experiments. In general, Random Forests were the best performing models, scoring a recall of 89.39% on binary (transient & non-transient) classification. Six-class transient classification scored a 77.86% recall, and a 66.91% recall when including an ambiguous sources class with new samples. Variations on the multi-class transient classifications mentioned above, after including non-transient sources, resulted on a 77.31% recall and a 67.23% recall respectively.

The usage of these techniques, which have shown successful results, may be further improved by using additional curve statistical descriptors and curve-fitting techniques. Nevertheless, filtering and oversampling data proved to be helpful techniques, as well as using the features proposed. The proposed method thus, presents new methods for transient object recognition that will benefit the upcoming epoch of astronomical object recognition.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter the reader is familiarized with the main ideas that are required to comprehend the rest of the project. First, astronomy-related concepts (Section 1.1) are introduced to the reader, followed by an understandable brief of the machine learning context (Section 1.2). Then, an exploration on the state of the art of automatized astronomical transient object recognition using machine learning (Section 1.3) is done. Finally, the goals of the current project are defined (Section 1.4) and the structure of the rest of the document is resented (Section 1.5).

1.1 Astronomical Context

Astronomy is a natural science that studies the origin and behaviour of all celestial objects. The term comes from the greek *astron* and the suffix *nomia*, meaning “the law of the stars”. This science also examines all other phenomena that occurs in space: planets, stars, galaxies are among many of the bodies that are researched, while gamma ray bursts, dark energy and the big bang exemplify astronomical events that are studied.

Studying this science is fundamental for the understanding of our position in the universe. It seeks to give answers to fundamental questions regarding the origin, evolution and fate of the universe, through the use of mathematics, chemistry and physics. Theories such as the big-bang and the heat death of the universe give plausible explanations to some of those questions. Such explanations are devised by theoretical astronomers, who seek to model the universe with mathematics. Experimental observation is also required for this science to be studied, thus research in this area is split into two fields: observational astronomy and experimental astronomy.

1.1.1 Observational Astronomy

Observational astronomy is the sub-field of astronomy that focuses on the collection and analysis of data about the observable universe. It does so by utilizing a numerous amount of instruments and visualization techniques. The main purpose of this field is to understand the universe through the information recorded from stellar objects.

A variety of data emitted by celestial objects is acquired and stored for usage in this field. Observations are gathered by using electromagnetic radiation, neutrinos, cosmic rays or gravitational waves. Known astronomical objects produce or reflect at least one of the previous types of data, and this information is utilized to recognize their characteristics and understand their behaviour.

The main devices used to capture stellar information are the telescopes. They have existed since the 17th century, and they are capable of capturing electromagnetic radiation in different bands of the spectrum. Telescopes may be classified by the wavelengths they detect, from radio-telescopes to gamma-ray telescopes. Due to the effects that the atmosphere has on electromagnetic waves, the best location to set a telescope is in Earth's orbit. It is still very expensive to locate a telescope in outer-space, so many telescopes are positioned instead in the crest of high mountains. There are telescopes that are set on flying devices too. Other tools capable of capturing different types of energy also exist: gravitational-wave detectors capture that gravitational waves, and neutrino detectors which capture neutrinos [2].

Electromagnetic radiation is one of the main sources of information of the universe. It is a form of energy that travels space as waves or packets (quanta). These waves travel at speed of light (approx. 300.000 km/sec) in vacuum, and at smaller speeds through other environments. The visible light that interacts with our eyes and allows us to see exemplifies a kind of electromagnetic wave.

According to their oscillation frequency and related wavelength, electromagnetic waves are classified in the electromagnetic spectrum [3]. The higher the frequency (thus smaller wavelength) of a wave, the more energy it carries. This spectrum classifies waves in seven groups, ordered with the most energetic waves first: gamma rays, X-rays, ultraviolet, visible light, infrared, microwaves and radio waves. As the wave's behaviour depends on their frequencies, this system allows a better distinction among different radiation signals.

1.1.2 Light curves

Light curves are visualization tools used in observational astronomy. These are graphs that show the luminosity of a space region or stellar object as it varies over time. In this context, luminosity is the light intensity of the astronomical objects in the sky. Light curves might be periodic or not, depending on the patterns of the objects observed. Additionally, they tend to be bounded to certain frequency or band. Light curves are well suited for the study of phenomena which varies with time, making them very useful for time-domain astronomy [4] [5]. These instruments are very useful to characterize astronomical object's behaviour for classification.

Luminosity is usually measured in light curves with a measurement called *apparent magnitude*. This measurement is used quantify the brightness of light, understood as the electromagnetic radiation at ultraviolet, visible, or infrared wavelengths. It uses a logarithmic scale, where the most negative magnitudes imply a higher brightness source.

1.1.3 Time-domain Astronomy

Time-domain is the field of astronomy that studies how deep-sky (beyond our solar system) objects change in time, allowing scientists to understand how the universe is evolving. It's study may allow the discovering of new astronomical phenomenon, making it one of the most interesting research topics in astronomy [1]. Due to the development of digital sky surveys that capture high resolution information of the sky in multiple time-stamps, an exponential growth of data volumes has occurred. Examples of these are the Sloan Digital Sky Survey [6] and 2MASS [7].

A specific set of phenomenon studied by time-domain astronomy are the Astronomical Transients (from now on *transients*): object events which luminosity varies in short duration (in the timescale of the universe), from minutes to several years. Though their time periods are large for humans. Transients include phenomena such as supernovae, novae, neutron stars, blazars, pulsars, cataclysmic variable stars (CV), gamma ray bursts (GRB) and active galaxy nucleus (AGN). Some of these are explained below.

- **Supernovae**: Plural for supernova, they are astronomical events that occur in the last stages of a massive star's lifetime. When a star runs out of fuel and it's fusion reaction pressure cannot withstand its own gravitational force, the core collapses, which results in the giant explosion of a supernova. Their duration occurs in a time span of hundreds of seconds [8].

- **Cataclysmic Variable (CV) Stars**: CVs are binary star systems that consist of a primary white dwarf and a companion normal star. The stars are so close together that the strong gravitational field of the white dwarf accelerates matter from the companion star onto the dwarf, via a process called accretion [9].

- **Active Galactic Nuclei (AGN)**: Many galaxies have a very bright central core, with a luminosity that may outshine the entire host galaxy. Such cores are named Active Galactic Nuclei, and galaxies hosting an AGN are named active galaxies. They are known to be the most luminous steady objects in the universe and their power output is variable on time scales of minutes up to years [10].

- **Quasars and Blazars**: Blazars and quasars are both types of AGNs with very high luminosity. They emit a pair of perpendicular relativistic jets perpendicular to their accretion disks. These beams are rays of ionized matter expelled at bulk velocities of 95% - 99.9% the speed of light, presumably generated by the active interaction of matter in the accretion ring that surrounds supermassive black holes. They're one of the most energetic objects known in the universe [11].

1.2 Machine Learning Context

Machine learning is the sub-field of Artificial Intelligence that uses data to teach algorithms how to perform certain tasks. Such algorithms can automatically detect patterns in data, and use this information to perform decision making under uncertainty [12]. Predicting future data and clustering data into unspecified groups are examples of decision making tasks that machine learning algorithms can perform. As these techniques are implemented in digital systems, they can be completely automatized.

These algorithms require data for training (learn how to perform tasks) and testing their resulting performance. A dataset D must be used, which contains an array of observations $x_i, i\epsilon[1, N]$, where N is the number of datapoints. Each observation is a vector $x_{i,j}, j\epsilon[1, D]$, where D is the number of features that observations have.

1.2.1 Types of Machine Learning

Machine learning algorithms can be classified mainly in three types, according to the task that they perform [13]:

- **Supervised learning** techniques learn to predict a mapping from inputs to outputs, which could be used to predict values for new input data. There's two kind of supervised learning techniques: classification and regression. When the output is belongs to an infinite number set, the problem is called **regression**. If the output $\hat{y}$ is one in a finite set of values (either labels or numbers), the learning problem is named classification. Classification could be either binary or multi-class.

- **Unsupervised learning** are used to learn interesting characteristics of the data. In contrast to supervised learning, patterns of interest or outputs are not explicitly defined. Examples of this task are clustering, and dimensionality reduction (e.g. PCA, LDA).

- **Reinforcement learning** algorithms learn behaviours in a specific environments (such as playing a video-game), by maximizing reward signals [14]. Instead of being told what to learn explicitly, as in supervised learning, algorithms must discover which actions optimize reward by exploring a variety of choices.

1.2.2 Classification Models

Several supervised learning algorithms exist, each with characteristics suitable for different contexts. Some of the most relevant algorithms are briefly explored below.

- **Artificial Neural Networks (ANN)**: These are machine learning models inspired by the operation and functioning of the brain. They consist of artificial neurons and connections between them, which simulate their biological counterpart. The first model was invented in 1958 and has been improved since. Currently they are the principal components of *deep-learning*, a field of machine learning that enables a computer to build complex concepts from simpler ones [15].

- **Classification and Regression Trees(CART)**: These are models that partition the space into cuboid regions, with edges aligned to the axes. Each cuboid is then assigned an outcome (classification value) [16]. They are called trees because the decision process reassembles a tree data structure, where the leaves represent the classification values for each cuboid region.
- **Support Vector Machines (SVMs)**: SVMs are models that find the decision boundary of the input space with the largest distance to all input points. They classify objects by finding lowest-error line that separates classes. Making use of an insight called kernel trick they are able to separate non-linearly separable data too, by embedding data into a higher-dimensional space [13].

1.2.3 Model Evaluation

Trained models are assessed to evaluate their performance and compare them. Various metrics and measurements are used to understand quantitatively their results. Moreover, validation techniques help to have certainty that the results are correct. These model evaluation basics are explained next.

1.2.3.1 Performance metrics

Traditionally used measurements in machine learning are described in this section for the case of binary classification. These measurements can be easily extended for multi-class classification, though those cases are not covered in this introduction.

Confusion Matrix A confusion matrix is a table of errors that specifies the possible outcomes when making predictions in classification. They give visual evidence of the mistakes in which the algorithm incurs. A binary classification confusion matrix will present the two types of error that may occur when doing predictions: *false positives*, which arise when predicting the objective class $\hat{y} = 1$ and the truth value is the negative class $y = 0$; *false negatives*, which arise when predicting the negative class, and the truth value is the objective class [12]. Table 1.1 describes a binary confusion matrix.

		Truth	
		1	0
Estimate	1	True Positives (TP)	False Positives (FP)
	0	False Negatives (FN)	True Negatives (TN)

Table 1.1: Generic confusion matrix for binary classification.

Accuracy Accuracy is defined as the percentage of correctly classified samples. It's value will be in the range $[0., 1.]$, where values close to 1 are desired. Using the confusion matrix, this measurement is calculated with the Formula 1.1.

$$\frac{TP + TN}{TP + TN + FP + FN} \tag{1.1}$$

Accuracy may not necessarily be the best metric when handling unbalanced classes. A classification algorithm that predicts accurately the class with more samples might score a high accuracy, even if it is miss classifying most of the alternate class' samples. Using Precision and Recall can solve this problem.

Precision Precision is defined as the fraction of detections detected by the model that were correct. It's value will be in the range $[0., 1.]$, where values close to 1 are desired. Using the confusion matrix, this measurement is calculated with the Formula 1.2.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1.2}$$

Recall Recall is defined as the fraction of true events detected. It's value will be in the range $[0., 1.]$, where values close to 1 are desired. Using the confusion matrix, this measurement is calculated with the Formula 1.3.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{1.3}$$

F1-Score It is defined as the harmonic average of Precision and Recall. The F1-Score combines the Precision and Recall of a model in a single value, which maximizes as both metrics also maximize. It's value will be in the range $[0., 1.]$, where values close to 1 are desired. Using the confusion matrix, this measurement is calculated with the Formula 1.4.

$$\text{F1-Score} = \frac{2PR}{R + P} \tag{1.4}$$

where:

P = Precision
N = Recall

1.2.3.2 Validation

Models are prone to over-fitting. This means they are likely to have decision boundaries so well adjusted to the training data, that they're unable to generalize correctly to new inputs. Over-fitting occurs partly because input data is high-dimensional, and a vast amount of data would required to model perfectly the observation's density distributions. As a result, if the model parameters are tuned according to the performance of the training data, the resulting decision boundaries become adjusted to this data and the model will generate errors when making predictions on new input data. It's important to note that methods based on Bayesian inference may not incur in over-fitting at all, but they're out of the scope of this project.

To solve the over-fitting problem a validation set V can be used. By adjusting the model's parameters to the performance metrics obtained from using the validation data instead of the training data, the model learns to generalize to unseen inputs. This data-set must not be used for training and it could be created by extracting a sub-set of data from D before training (a maximum of 25%). The process of using the validation data set to assess the model's performance on unseen inputs, and verify if it learns to generalize, is named *validation.*

Acquiring unbiased performance metrics that assess the model's final performance is required to obtain valid results. To do so, a data set that was not used during the training-validation process should be used for their computation. An additional test set should be used to generate these results. Such test set could also be extracted from the original set of data.

Cross-validation When D is too small, the error metrics obtained using V aren't useful. As using a very small validation set increases the statistical uncertainty around the estimated mean validation error, the obtained metrics will be less precise. To mitigate this problem and be able use performance metrics, an approach called *cross-validation* exists. It consists in using the whole data-set D for both training and validation. The most common cross-validation technique is named k-fold cross validation, which partitions the data-set D into k equal size non-overlapping sub-sets. It then uses $k-1$ sub-sets for training, and 1 sub-set for validation, and repeats the this process k times, using each time a different sub-set for validation. The validation error is then estimated by taking the average validation error across all k trials.

1.3 State of the art

Transient and variable star detection is currently implemented in most surveys with a process named difference imaging [17] [18] [19]. It consists in aligning the image of interest on a reference image of the same region of sky, processing the former to match it's point-spread function in all regions with the latter's, and subtracting both [20]. Variable stars and transient objects which were not visible in the reference image will remain in the resulting image. Nevertheless, difference images will also contain additional artificial (bogus) artifacts generated in the process, which occur due to imperfections in the image processing phase, in the telescope used or due to natural phenomena. Distinguishing between bogus and real objects is still a hand-made process which is expensive and slows down knowledge discovery [1].

Researchers have studied how to recognize such bogus objects automatically by using machine learning algorithms on raw images. This is shown in [21], where Convolutional Neural Networks (CNN) were trained on the Skymapper Supernova and Transient Survey data to recognize if sources were real. In this paper, an accuracy of 97.3% was obtained testing on real instances, and a 99.7% accuracy was obtained testing on bogus sources. A similar study is [22], where a rotation invariant CNN was trained on data from the High cadence Transient Survey(HiTS). An accuracy of 99.45% was obtained when testing the algorithm on new data.

Scientists have also studied the transformation of candidate transient images into features, to train automatic recognition algorithms that identify real and bogus sources. In [23] and [24], 2-dimensional source images are angle-normalized to reduce the number of artifacts and then converted into a 1-dimensional arrays, by joining contiguous columns and inserting values for the heights of the bins of a normalized histogram of the image. A hierarchical classifier of three Self-Organizing Maps models is trained with data from data from the OGLE-IV data-reduction pipeline, and an accuracy of 97% of real transient and 97.5% of bogus candidate detections is obtained. Similarly, in [25] a 1-dimensional vector of each candidate is constructed by shifting off each column of the images (centered on the candidate) and concatenating those columns together to produce a vector. The image data used in this study consists of real astrophysical transients and bogus detections from the Pan-STARRS1 Medium Deep Surveys, and is used to train ANNs, SVMs and Random Forests (RF). By accepting a false positive rate of 1%, a completeness of 93.8% is obtained. Finally, in [26] eigen-image analysis (PCA, LDA) of single-epoch multi-band images from the SDSS supernova survey are used as input on RF, K-Nearest Neighbors (KNN) and ANNs. They obtain a completeness (recall) of 96%, while only incorrectly classifying at most 18% of artifacts as real objects, corresponding to a precision (purity) of 84%.

Astronomers have also extracted features from transient's light-curves to classify them in different types. In [27], light curves of objects belonging to 7 transient types and non-transient sources are used for classification. The dataset includes of photometric observations from the Catalina Real Time Transient Survey and the Downes set [28]. Light curves are modeled with a Gaussian process regression and 10 curve measurements are extracted from it, as well as 16 non-periodic features defined in [29]. The features are used as input to train 5 classification algorithms: Linear Discriminant Analysis, Decision Trees, SVMs and ANNs. A top accuracy of 90.5% was obtained on 8-class classification; binary classification of transients and non-transients achieved an accuracy of 92.0%. 7-class transient classification scored an accuracy of 74.3%. Finally, 5-class transient classification obtained a 79.0% accuracy.

Other studies also use feature extraction for transient classification. In [30], data from the CRTS containing 6 types of transient objects was used to derive both periodic and non-periodic features. These features were used as input to train the models: ANNs with Quasi-Newton algorithm, RF and KNN. A 79.4% efficiency was obtained on 6-class transient classification. Likewise, in [31] classification of photometric supernovae was researched using the algorithms: Naive Bayes, KNN, SVM, ANN, and Boosted Decision

Trees (BDTs). The dataset used belongs to the Supernova Photometric Classification Challenge, and consists of simulated light curves of non-Ia and Ia supernovae. Model-dependent techniques, independent techniques that fit parametric models to curves and model-independent wavelet approaches were used for feature extraction. An area under the curve (AUC) of 0.98 was obtained.

Lastly, the direct use of light curves as inputs is currently under research. In [32], Recurrent Neural Networks are trained to classify supernovae using data from the Supernovae Photometric Classification Challenge (SPCC). The best binary classifier type-Ia and non-type-Ia supernovae scored an accuracy of 94.7% and an area under the curve (AUC) of 0.986. Conversely, 3-class classification between supernovae types using bidirectional neural networks achieved an accuracy of 90.4% with an AUC of 0.974. Other studies that have employed RNNs for classification using astronomical light-curves include [33].

1.4 Project objectives

As mentioned before, astronomical transient recognition is a process in which humans still intervene, making it a slow and expensive procedure. Through computational techniques, transient detection can be performed much faster than human astronomers, with real-time triggering of follow-up observations that optimize the economical and temporal resources. Moreover, concerns exist in that hand-made classification tends to be biased and hard to standardize among astronomers [34], while classification using supervised machine learning techniques are deterministic and give degrees of certainty on the results.

In the given context, the objectives of this project are:

1. Propose a machine learning system that classifies astronomical sources as transients and not transients, using their respective light curves.

2. Propose a machine learning system that classifies astronomical sources as one of seven types of transients, using their respective light curves.

3. Implement a working pipeline that takes astronomical objects light curves as input and returns the best suited classifier for the binary

and multi-class classification tasks mentioned in the previous two objectives.

4. Lay the foundations on the development Transient Object Classifiers in the scope of the LSST project [35], that will enable building more complex and accurate systems in the near future by research groups at Universidad de los Andes and CCPM.

1.5 Document Structure

Remaining document's chapters are divided in the following way:

- Chapter 2: Describes the data-set used for this project.
- Chapter 3: Presents the methodology used and the experiments performed to classify transient objects using machine learning.
- Chapter 4: Shows and comments on the results obtained for each experiment proposed in Chapter 3.
- Chapter 5: Concludes the project by analyzing the results and suggesting future work that could improve the results obtained in this project.

Chapter 2

Data

The data-set used in this project was explored before it was used for classification, and the results of this process are shown in the current chapter. Presenting the information obtained on the data is relevant, as it gives the reader an understanding of the behaviour of real astronomical light curves. Exploring the data was also a necessary step to design the methodology found in Chapter 3, since the algorithms can perform incorrectly if data is incomplete or inconsistent.

This episode presents the mentioned information in the following way: first an overview of the data an its origin is introduced (Section 2.1). Then, multiple metrics of the light curves are displayed (Section 2.2), all of which were computed for the different sub-sets of curves used in the project. Finally, light curves of representative samples of astronomical objects are displayed (Section 2.3).

2.1 Dataset Overview

The data used in this project belongs to the Catalina Real-Time Transient Survey (CRTS) [36]. As it's name implies, the CRTS is an astronomical survey in the search of transient and highly variable objects. It covers 33,000 squared degrees of the sky in search of this kind of objects and has surveyed astronomical sources since 2007. All discoveries are captured in real-time and publicized immediately after, so that others may too observe the events. Three telescopes are used to capture data from the sky: Mt. Lemmon Survey (MLS), Catalina Sky Survey (CSS), and Siding Spring Survey (SSS). So far, CRTS has discovered more than 15.000 transient events.

The data-set used in this project contains information of 4985 transient objects detected with the CSS telescope of the CRTS. This f/1.8 Schmidt telescope is located in the Santa Catalina Mountains, north of Tucson, Arizona. It is equipped with a 111-megapixel (10,560 x 10,560 pixel) detector, and covers 4000 square degrees per night, with a limiting magnitude of 19.5 in the visual filter band [37].

Transient event data is divided into two catalogues: a transient object catalogue, with information on the id, classification type and position of each source; a transient light curve catalogue, containing all photometric observations (magnitude and error) of the transient objects, associated to their respective id and observation date (as modified Julian date). These two catalogues compose the transient data-set used.

All transient objects are classified in the CRTS data-set according to their type. The most relevant classes found are: supernovae, cataclysmic variable stars (CV), blazars, flares, asteroids, active galactic nuclei (AGN), and high-proper-motion stars (HPM). Though most objects in the transient object catalogue belong to certain class, there's uncertainty with some (they might display this with an interrogation sign e.g. SN? or showing as having more than one possible class e.g. SN/CV).

Furthermore, the data-set used in this project contains information of 16940 non-transient sources. The sources in this data-set were selected by sampling light curves of objects within a 0.006 degree radius from CRTS detected transients, and by removing known transient light curves from this set. Though this process should return only non-transient sources, it is possible that non-detected transients are being captured here and catalogues as non-transients. Non-transient data is then grouped in a light curve catalogue containing all photometric observations of non-transients, in the same way as the transient-catalogue. There is not a non-transient object catalogue in this data-set.

2.2 Light curve Metrics

Data-set's light curves were grouped into different sub-sets for classification. Two sub-sets of data were created by filtering the total number of light curves by their respective number of observations: those with a minimum of 5 and those with at least 10. Metrics on the raw data without filtering and the given sub-sets are presented next.

2.2.1 Raw data-set's metrics

The first metric calculated on the raw data-set is displayed in Table 2.1. This table displays the transient and non-transient object count, showing that there are approximately 3.5 times more sources of the latter class.

Type	Transient	Non-Transient
Obj. Count	4985	16940

Table 2.1: Raw data-set's astronomical object count.

Another metric obtained from this data-set is the number of objects that belong to the top transient sub-classes. Table 2.2 shows the top 7 types of transients in this data-set, ordered in descending order by their object count. As the table shows, the top certain sub-classes are: Supernovae (SN), Cataclysmic Variable (CV) Stars, High Proper Motion (HPM) Stars, Active Galactic Nuclei (AGN), Blazar and Flare.

Class	SN	CV	HPM	AGN	SN?	Blazar	Flare
Obj. Count	1539	943	436	429	294	239	215

Table 2.2: Raw data-set's top 7 transient classes with the most objects.

Finally, a group of statistics were calculated on the number of observations per object light curve. These statistics include: the mean value, standard deviation, minimum and maximum values, the 25th, 50th and 75th percentiles. Such metrics were computed for transients in Table 2.3) and for non-transients in Table 2.4.

Mean	Std	Min	25%	50%	75%	Max
90.58	112.33	1	9	35	140	880

Table 2.3: Raw data-set's transient observation count per object.

Mean	Std	Min	25%	50%	75%	Max
113.60	129.26	1	16	61	176	1266

Table 2.4: Raw data-set's non-transient observation count per object.

2.2.2 Filtered data-set's metrics

Data-set's filtering is a process that is done in order to obtain light curves with enough observation samples for them to be characterized. This is a step that is thoroughly described in Chapter 3, and it mainly consists in creating two sub-sets of light curves with at least 5 and 10 observations. Additionally, the non-transient set of data is sub-sampled to have the same number of transient light curves. In this section, metrics for such filtered data sub-sets are presented.

2.2.2.1 Objects filtered by 5 observations minimum

The first metric calculated on the raw data-set is displayed in Table 2.5. This table displays the transient and non-transient object count, showing that Both classes have the same number of objects, as expected and explained in Section 3.1. In comparison to Table 2.1 this data sub-set contains 601 less transients, totaling 4384.

Type	Transient	Non-Transient
Obj. Count	4384	4384

Table 2.5: Data-set's transient observation count per object, when filtered by at least 5 observations.

Another metric obtained from this data-set is the number of objects that belong to the top transient sub-classes. Table 2.6 shows the top 7 types of transients in this data-set, ordered in descending order, by their object count. As the table shows, the top certain sub-classes are: Supernovae (SN), Cataclysmic Variable (CV) Stars, Active Galactic Nuclei (AGN), High Proper Motion (HPM) Stars, Blazar and Flare.

Class	SN	CV	AGN	HPM	SN?	Blazar	Flare
Obj. Count	1295	862	427	412	239	237	207

Table 2.6: Data-set's top 7 transient classes with the most objects, when filtered by at least 5 observations.

Finally, a group of statistics were calculated on the number of observations per object light curve. These statistics, regarding the observation count, include: the mean value, standard deviation, minimum and maximum values, the 25th, 50th and 75th percentiles. Such metrics were computed for transients in Table 2.7 and for non-transients in Table 2.8. In comparison to Table 2.3, the mean number of observations and standard deviation of transient light curves increased. Moreover, in comparison to Table 2.4, non-transient light curves average number of observations increased and its standard deviation slightly decreased.

Mean	Std	Min	25%	50%	75%	Max
102.62	114.63	5	15	48	163.25	880

Table 2.7: Data-set's transient observation count per object, when filtered by at least 5 observations.

Mean	Std	Min	25%	50%	75%	Max
120.37	115.85	5	27	73	196	527

Table 2.8: Data-set's non-transients observation count per object, when filtered by at least 5 observations.

2.2.2.2 Objects filtered by 10 observations minimum

The first metric calculated on the raw data-set is displayed in Table 2.9. This table displays the transient and non-transient object count, showing that both classes have the same number of objects, as expected and explained in Section 3.1. This data sub-set contains 601 less transients than in Table 2.1, and 656 less than in Table 2.5, totaling 4384.

Type	Transient	Non-Transient
Obj. Count	3728	3728

Table 2.9: Data-set's transient observation count per object, when filtered by at least 10 observations.

Another metric obtained from this data-set is the number of objects that belong to the top transient sub-classes. Table 2.10 shows the top 7 types of transients in this data-set, ordered in descending order, by their object count. As the table shows, the top certain sub-classes are: Supernovae (SN), Cataclysmic Variable (CV) Stars, Active Galactic Nuclei (AGN), High Proper Motion (HPM) Stars, Blazar and Flare. In comparison to Table 2.6, all the top classes remain in the same order, except the ambiguous supernovae 'SN?' which's count decreased significantly.

Class	SN	CV	AGN	HPM	Blazar	Flare	SN?
Obj. Count	1050	782	426	400	232	188	127

Table 2.10: Data-set's top 7 transient classes with the most objects, when filtered by at least 10 observations.

Finally, a group of statistics were calculated on the number of observations per object light curve. These statistics, regarding the observation count, include: the mean value, standard deviation, minimum and maximum values, the 25th, 50th and 75th percentiles. Such metrics were computed for transients in Table 2.11 and for non-transients in Table 2.12. In comparison to Table 2.7, the mean number of observations and standard deviation of transient light curves increased slightly. Conversely, non-transient light curves average number of observations increased and its standard deviation remained almost the same, in comparison to Table 2.4.

Mean	Std	Min	25%	50%	75%	Max
119.50	116.40	10	24	72	188	880

Table 2.11: Data-set's transient observation count per object, when filtered by at least 10 observations.

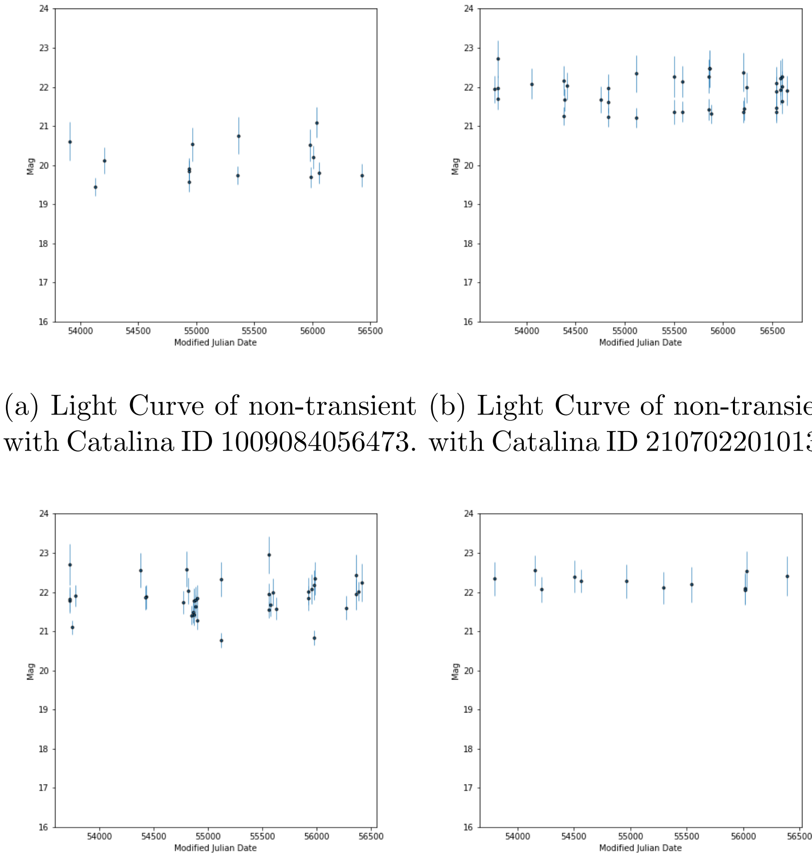
Mean	Std	Min	25%	50%	75%	Max
131.95	115.43	10	39	87	208	527

Table 2.12: Data-set's non-transients observation count per object, when filtered by at least 10 observations.

2.3 Light Curves

Transient objects vary their luminosity in short timescales, while non-transient's luminosity tend to stay constant through time. The following figures illustrate various data-set samples from both types of astronomical objects, which visually demonstrate how their behaviour differ. Light curves' apparent magnitudes for a given time-stamp are represented with dots, while vertical lines symbolize the magnitude's error for the same time value. Time-stamps (X-axis) are displayed in Modified Julian Date units, which are the number of days since midnight November 17, 1858.

Figure 2.1 displays four non-transient object light curves. Furthermore, four types of non-transient objects are also displayed in the following figures: Figure 2.2 shows samples of four AGN object light curves, Figure 2.3 visualizes samples of four different Blazar object light curves, Figure 2.4 presents the light curves of four different Cataclysmic Variable stars, and Figure 2.5 displays samples of four Supernovae light curves. It is noticeable from these figures that non-transient objects tend to remain with a constant Magnitude and errors, while transient objects vary in a much bigger scale.



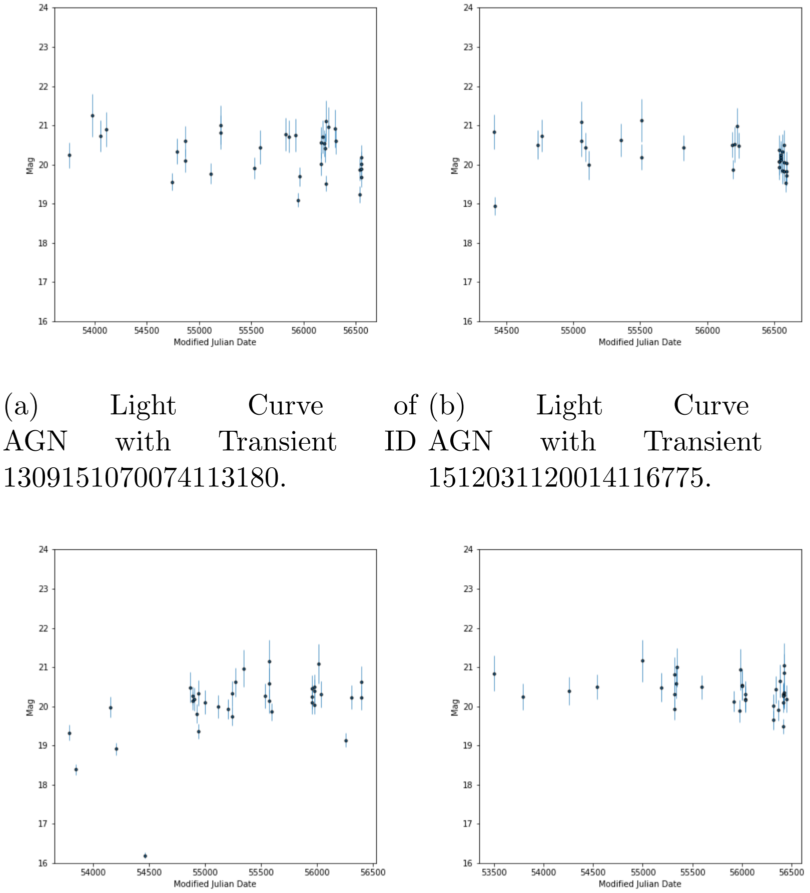
(a) Light Curve of non-transient with Catalina ID 1009084056473.

(b) Light Curve of non-transient with Catalina ID 2107022010130.

(c) Light Curve of non-transient with Catalina ID 2121100028895.

(d) Light Curve of non-transient with Catalina ID 2004185017956.

Figure 2.1: Non-transient objects Light Curves.



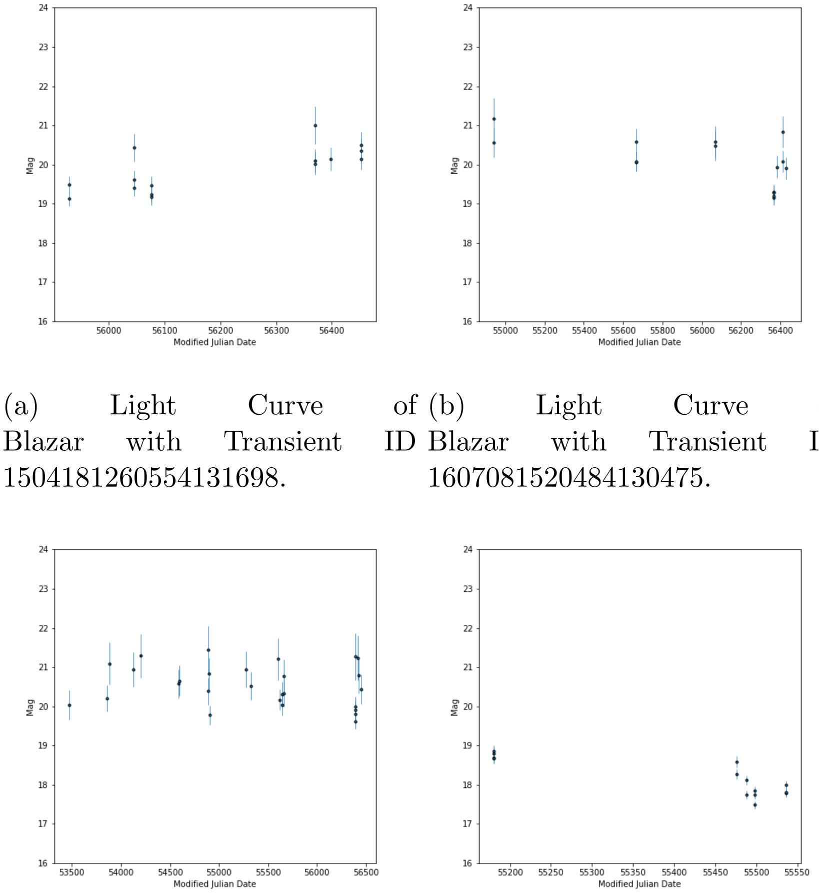
(a) Light Curve of AGN with Transient ID 1309151070074113180.

(b) Light Curve of AGN with Transient ID 1512031120014116775.

(c) Light Curve of AGN with transient ID 1511021180544114352.

(d) Light Curve of AGN with transient ID 1504270070764122223.

Figure 2.2: AGN Light Curves.



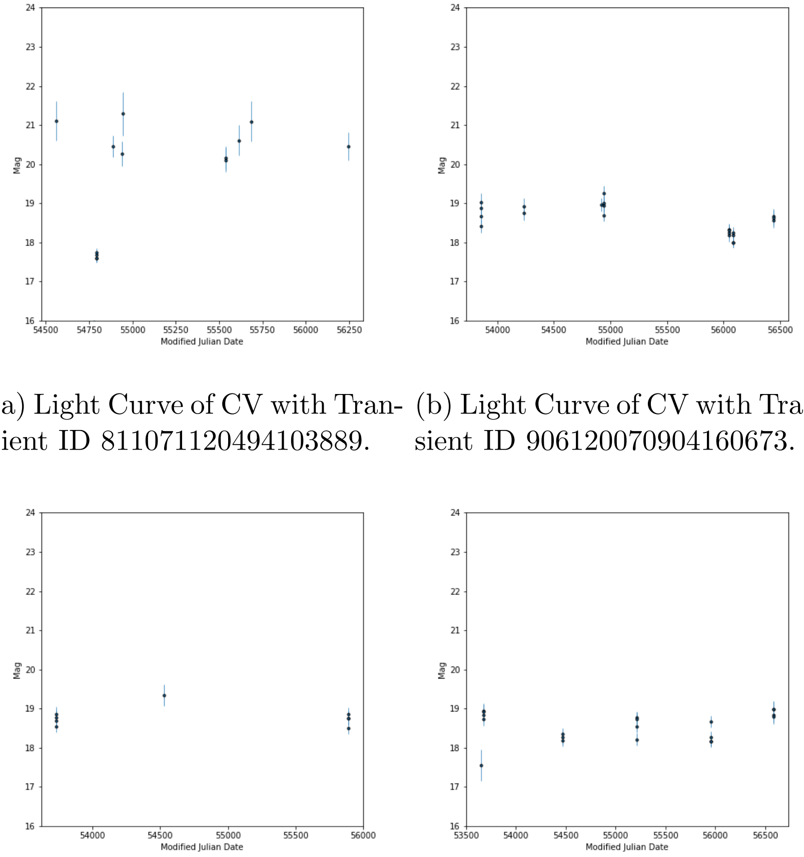
(a) Light Curve of Blazar with Transient ID 1504181260554131698.

(b) Light Curve of Blazar with Transient ID 1607081520484130475.

(c) Light Curve of Blazar with transient ID 1609101690374105884.

(d) Light Curve of Blazar with transient ID 1611061091234116212.

Figure 2.3: Blazar Light Curves.



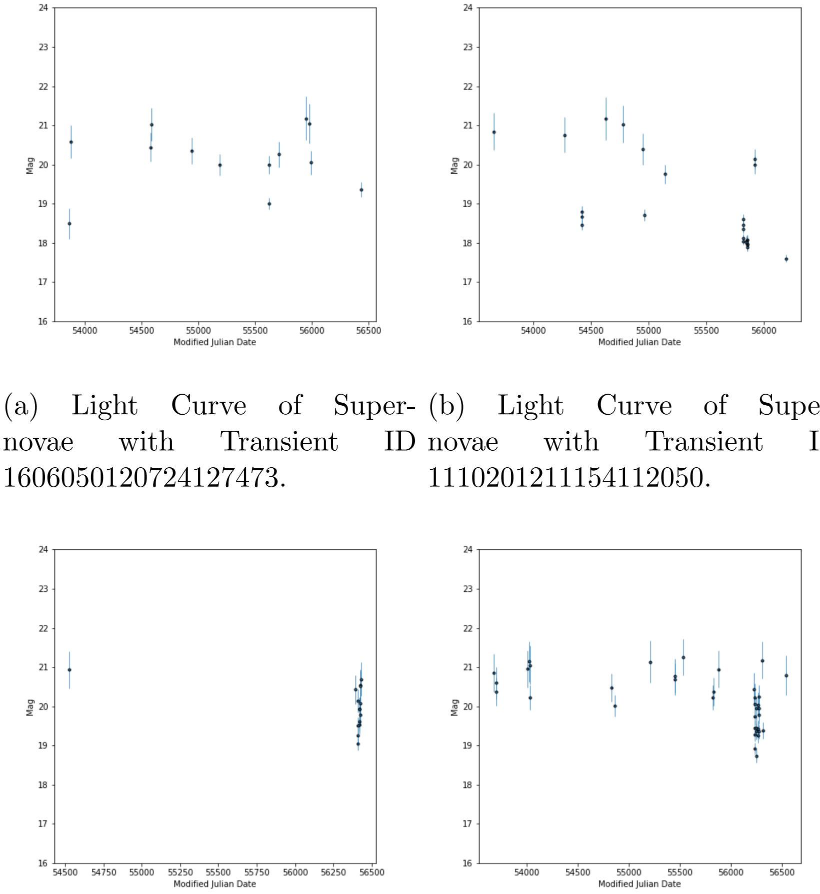
(a) Light Curve of CV with Transient ID 811071120494103889.

(b) Light Curve of CV with Transient ID 906120070904160673.

(c) Light Curve of CV with transient ID 1601030090334163193.

(d) Light Curve of CV with transient ID 1001131380174108966.

Figure 2.4: Cataclysmic Variable Stars Light Curves.



(a) Light Curve of Supernovae with Transient ID 1606050120724127473.

(b) Light Curve of Supernovae with Transient ID 1110201211154112050.

(c) Light Curve of Supernovae with transient ID 1304291090754107901.

(d) Light Curve of Supernovae with transient ID 1211071070144131502.

Figure 2.5: Supernovae Transient Object Light Curves.

Chapter 3

Methodology

In this chapter, the methodology proposed to classify transient events with machine learning and using light curves is presented. Mainly, the pipeline consists of four stages: filtering out irrelevant light curves (Section 3.1, oversampling the remaining curves by using the magnitude errors as a probability density function (Section 3.2), extracting descriptive features from the curves (Section 3.3), processing feature vectors (Section 3.4), and lastly, performing classification using a three machine learning algorithms (Section 3.5). Specific details on the experiments designed to build the best classifiers are discussed at the end of the chapter (Section 3.6).

3.1 Data Filtering

In this step, noisy and incomplete data gathered from the Catalina Real Time Survey is filtered out to increase the accuracy of the classification algorithms. The following actions are executed on both the transient and non-transient light curves:

- **Remove Sources with Few Observations:** Astronomical objects with few observations don't contain enough data-points for the feature extraction phase. This is why two sub-sets of data were created by filtering out light curves with a small number of observations: one containing light curves with 5 observations minimum, and another one containing 10 observations minimum per light curve.

- **Remove Duplicates:** All observation duplicates for a same astronomical source where the time-stamp was repeated, were removed. These duplicates correspond to blended observations that include the luminosity of nearby astronomical sources, which would add noise to the data if not filtered out.

Since the amount of non-transient light curves is much higher than the number of transient curves, only a sub-set of the former are used for classification. Two sub-sets of non-transient light curves are randomly sampled, one using the non-transient set that have at least 5 observations, and another one using the non-transient set of objects that has at least 10 observations. Each newly sampled set of non-transient light curve has the same amount of light curves as the analogous set of transient objects.

Metrics for the resulting data-sets are presented in Section 2.2.2.

3.2 Data Oversampling

The number of light curves present for each transient class is not equal, or even balanced, as presented in Section 2.2.2. In order to test the behaviour of the classification algorithms on classes with the same number of objects, an oversampling step was designed and implemented.

In this phase, for each observation point of a 'real' light curve a new observation was sampled using a probability distribution function. This created a new light curve with the same number of observation points as the one used for sampling. Specifically, each observation magnitude and error were used to model a Gaussian probability distribution function centered on the 'real' magnitude value, with a standard deviation equal to the magnitude error. This re-sampling process was repeated for each transient light curve several times, creating a new data set containing balanced classes. This process was executed for both sub-sets of data mentioned in Section 3.1.

3.3 Feature Extraction

Light curve's observations are not sampled at regular intervals nor all instances of a certain class have the same number of observations (see Section 2.2. This makes it very challenging to use the time-series data directly for classification using traditional methods. To solve this difficulty, a set of characteristic features are extracted from each light curve instead, using statistical and model-specific fitting techniques. Some measurements used in this project are formally introduced in [29], though they are used in many studies, including [31] and [30].

In total 31 features were used in this project, each one calculated for every light curve. These measurements can be classified in the following four groups, and are described below: moment-based, magnitude-based, percentile-based and fitting-based.

Moment-based features use the magnitude for each light curve:

- **skew:** Skewness.
- **kurtosis:** Kurtosis.
- **small_kurtosis:** Small sample kurtosis.
- **std:** Standard deviation.
- **beyond1std:** Percentage of magnitudes beyond one standard deviation from the weighted mean. Each weights is calculated as the inverse of the corresponding photometric error.
- **stetson_j:** The Welch-Stetson J variability index [38]. A robust standard deviation.
- **stetson_k:** The Welch-Stetson K variability index [38]. A robust kurtosis measure.

Magnitude-based features also use the magnitude for each source:

- **max_slope:** Maximum absolute slope (delta magnitude over delta time) between two consecutive observations.
- **amplitude:** Difference between maximum and minimum magnitudes.
- **median_absolute_deviation:** Median discrepancy of magnitudes from the median magnitude.
- **median_buffer_range_percentage:** Percentage of points within 10% of the median magnitude.
- **pair_slope_trend:** Percentage of all pairs of consecutive magnitude measurements that have positive slope.

- **pair_slope_trend_last_30:** Percentage of the last 30 pairs of consecutive magnitudes that have a positive slope, minus percentage of the last 30 pairs of consecutive magnitudes with a negative slope.

Percentile-based features use the sorted flux distribution for each source. Flux is calculated as $F = 10^{0.4mag}$. Defining $F_{a,b}$ as the difference between the b and a flux percentiles:

- **percent_amplitude:** Largest percentage difference between the absolute maximum magnitude and the median.
- **percent_differenc_flux_percentile:** Ratio of $F_{5,95}$ and the median flux.
- **flux_percentile_ratio_mid20:** Ratio $F_{40,60}/F_{5,95}$
- **flux_percentile_ratio_mid35:** Ratio $F_{32.5,67.5}/F_{5,95}$
- **flux_percentile_ratio_mid50:** Ratio $F_{25,75}/F_{5,95}$
- **flux_percentile_ratio_mid65:** Ratio $F_{17.5,82.5}/F_{5,95}$
- **flux_percentile_ratio_mid80:** Ratio $F_{10,90}/F_{5,95}$

Fitting-based features also use the magnitude for each source:

- **poly1_a:** Coefficient of the linear term in monomial curve fitting.
- **poly2_a:** Coefficient of the cuadratic term in cuadratic curve fitting.
- **poly2_b:** Coefficient of the linear term in cuadratic curve fitting.
- **poly3_a:** Coefficient of the cubic term in cubic curve fitting.
- **poly3_b:** Coefficient of the cuadratic term in cubic curve fitting.
- **poly3_c:** Coefficient of the linear term in cubic curve fitting.
- **poly4_a:** Coefficient of the quartic term in quartic curve fitting.
- **poly4_b:** Coefficient of the cubic term in quartic curve fitting.
- **poly4_c:** Coefficient of the cuadratic term in quartic curve fitting.

- **poly4_d:** Coefficient of the linear term in quartic curve fitting.

The computed features were grouped in the following sets:

- 21 feats: This sub-set includes all moment-based, magnitude-based and percentile-based features.
- 31 feats: This sub-set includes all features, except quartic curve fitting parameters (poly4).
- 27 feats: This sub-set includes all moment-based, magnitude-based, percentile-based and fitting-based features.

Each of these sets of features was used to extract a feature vector for every light curve, in order to compare which worked the best, through different experiments (details in Section 3.6).

All code required to compute the measurements is written as part of the project's requirements. Broad description of the files with the implementation may be found in Appendix A.

Having a calculated vector of measurements results in a homogeneous set of data that can be used nicely for classification next.

3.4 Data Preprocessing

A small data pre-processing step was executed previous to classification, where feature scaling was applied to the feature vectors. This is a method where all feature values were re-scaled, so that each one contributes proportionately to the final result when using machine learning algorithms. Feature scaling was applied independently to all the values of the same feature.

Two feature scaling procedures are used in this project, both in different scenarios, since they can't be applied consecutively. The first scaling procedure standardizes each feature, by moving its mean to zero, and scaling to have unit variance. Conversely, the second scaling feature transforms features by scaling them to be in the range $[0, 1]$.

It is important to note that the pre-processing step was (and always must be) applied to the data previous to training, validation and testing.

3.5 Classification

Five classification tasks were performed with the homogenized feature vectors computed in the pre-processing step:

1. Binary Classification: Distinguish transients from non-transients.
2. 6-Transient Classification: Recognize objects as belonging to one of the following transient types: AGN, Blazar, CV, Flare, HPM and Supernovae.
3. 7-Transient Classification: Recognize objects as belonging to one of the following transient types: AGN, Blazar, CV, Flare, HPM, Other and Supernovae. The Other class is created by using objects from ambiguous and under-represented classes.
4. 7-Class Classification: Recognize objects as belonging to one of the following classes: AGN, Blazar, CV, Flare, HPM, Non-Transient and Supernovae.
5. 8-Class Classification: Recognize objects as belonging to one of the following classes: AGN, Blazar, CV, Flare, HPM, Other, Non-Transient and Supernovae. The Other class is created by using objects from ambiguous and under-represented classes, just like in 7-Transient Classification.

The non-transient data-set contains more light curves than the transient data-set, so a sub-sample of objects of the former class was used in all tasks where the non-transient class was classified. Such sub-set contained the same amount of light curves as the biggest transient class in every specific task.

Regarding the algorithms used, three different models were tested in all tasks: neural networks, random forests and support vector machines. These algorithms were chosen because they were found to be popular in previous studies (see Section 1.3. Moreover, these three algorithms can perform quickly classification under the low dimensional data that is being used, and their use in production pipelines could be studied too. Details on the inner workings of these algorithms can be found in [39]

None of the machine learning algorithms used in this study was coded from scratch, as that was not the focus of this research. On the other hand, multiple open-source libraries exist which provide APIs to easily create and train already implemented models. TensorFlow [40] and SciKit Learn [41] are examples of these. For this project SciKit-Learn was used, as it provides APIs to allows training, testing and validating the chosen models with few lines of code.

3.6 Experimental Framework

3.6.1 Cross Validation

To build and train highly accurate classification models, the hyper-parameters used to train the algorithms must be tuned to obtain the best classification boundaries. Considering that the initialization values of these parameters may derive how well the algorithm will perform, an experimentation framework was devised to obtain the best configurations and validate the result.

Each of the classifiers was trained by executing the following sequence of steps:

1. Use grid search to build several variations of each classification model, all with different hyper-parameters.
2. Train model variations with the training data-set.
3. Test trained model variations on a validation data-set.
4. Pick the best performing model variation on validation data.

Since the number of objects in the data sub-sets is small, cross validation was used to validate the results (see Section 1.2.3.2). 2-fold cross validation was used for this project, which means that the previous sequence of steps was repeated twice for each machine learning algorithms and the results were averaged per algorithm. The evaluation metrics used to assess the performance of the model was the F1-Score and Recall (see Section 1.2).

Previous to training the models, a test data-set was extracted from the feature data-set, representing a 33% of the samples. After training and validation, the best performing models were tested with this data.

3.6.2 Classification Experiments

Multiple experiments were executed to test which was the configuration of parameters that worked the best. Specifically, each experiment consisted of a unique combination of the parameters used in this proposal, which were:

- Task: There are five tasks in total, all described in Section 3.5.
- Min. Obs: Two sub-sets of information exists, those with light curves containing at least 5 observations, and those with light curves that have minimum 10 observations (Section 3.1).
- Oversampled: Light curves can be divided in two data-sets: those which are unbalanced and those oversampled to become balanced (Section 3.2).
- Num. Features: Three sub-sets of extraction features are described in Section 3.3 (21 feats, 27 feats and 31 feats).
- Feature Scaling: Two feature scaling methods are tested, as described in Section 3.4.
- Model: As described in Section 3.5, three classification algorithms were tested for all classification tasks.

According to this, each classification task is tested with 72 different models. All except binary classification, which used balanced classes by default, and only 36 models were tested for this case. The results of such process is presented in the following chapter.

Chapter 4

Results

This chapter presents the results obtained by applying the methodology described in Chapter 3. Detailed results for all the classification tasks are presented, as well as general results extracted from the analysis of all tasks as a whole.

The performance of running each classifier with a given sub-set of parameters is shown in this chapter for each classification task, making it easy for the reader to compare the results among different runs. Additionally, score metrics for the best performing model are presented, which give information on the classification performance of each distinct class in the task. Results for tasks which contain an unbalanced amount of objects (all multi-class) are also included, with a comparison to their analogous unbalanced experiments. All these quantitative descriptors are analyzed and summarized in each chapter's sub-section.

Though the experiments were run with two different feature scaling methods, results presented in this chapter are only displayed for the feature scaling alternative which resulted in higher performance. Feature scaling methods are important to obtain the highest performance possible, but their analysis was less relevant compared to the one of the rest of parameters used in this project.

4.1 Binary Classification

The following table presents the Recall obtained after running the binary classification experiment. Results are shown for each classifier, with the data-set's light curves filtered by 5 and 10 observations minimum, and using 21, 27 and 31 features.

Classifier	Min. observations	Recall		
		21 feat.	27 feat.	31 feat.
SVM	5	85.28	85.49	85.42
	10	85.41	**85.85**	**85.85**
Random Forest	5	86.45	88.32	88.22
	10	86.34	89.02	**89.39**
Neural Network	5	84.55	85.14	85.07
	10	85.24	85.81	**86.75**

Table 4.1: Recall for the Binary Classification Task. Top score for each classifier is in bold

The following information can be resumed from Table 4.1:

- Top recall of 89.39% was obtained by the Random Forest when using light curves 10 observations minimum.
- Random Forest was the best classifier in all variations.
- Increasing from 27 to 31 features slightly decreased recall when using light curves with 5 observations minimum.
- Increasing from 27 to 31 features increaseimproved performance when using light curves with 10 observations minimum.
- The best performances were obtained when using 10 obs. and 31 features.
- Neural Network was the worst performing algorithm, and only surpassed the SVM (by $\sim$ 1%) with light curves of 10 observations min. and 31 features.

The next table presents the specific scores on the classification performance of the best performing algorithm (Random Forest, 10 obs, 31 feats):

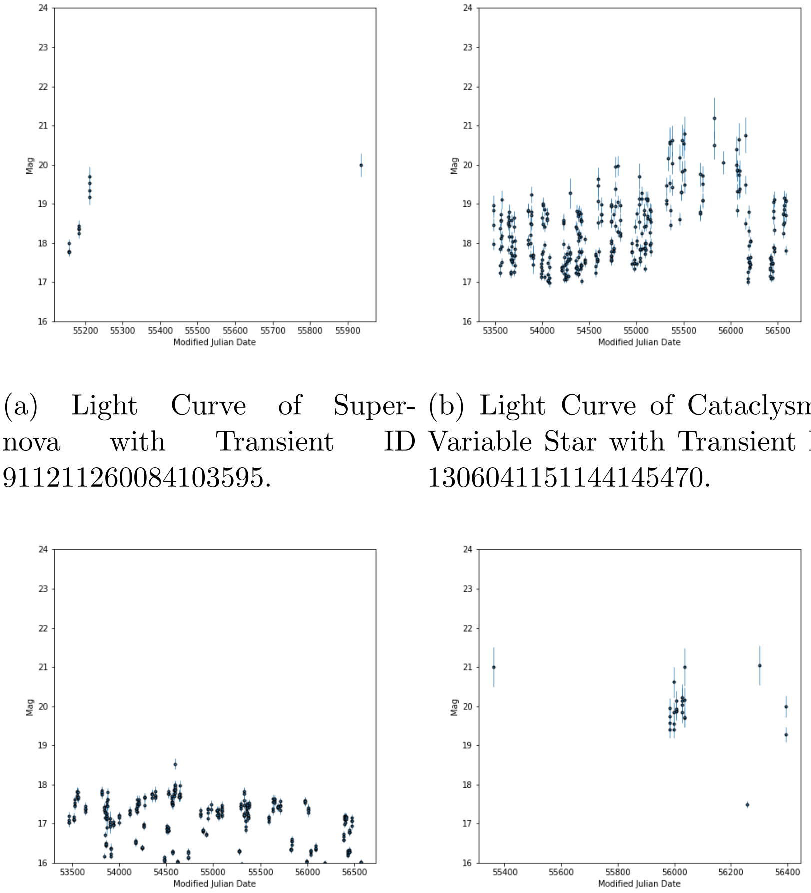
	precision	recall	f1-score	support
non-transient	87.41	92.03	89.66	1230
transient	91.59	86.75	89.10	1230
avg/total	89.50	89.39	89.38	2460

Table 4.2: Precision, Recall and f1-score for the Binary Classification Task

The following information can be resumed from Table 4.2:

- Recognition of Non-Transients and Transients had a similar f1-score.
- Non-Transient classification had higher Recall but lower precision than Transient classification.

Visualizations of sample transient and non-transient event light curves are presented next. Figure 4.1 presents the correctly classified transient event light curves, while Figure 4.2 introduces the light curve graphs of incorrectly classified transient event light curves. From these tables it can be recognized that transient events which are correctly classified tend to have a observable high variation for groups of points, while incorrectly classified ones have low variation and observation points tend to be very close together. Conversely, Figure 4.3 presents the light curves of correctly classified non-transient objects, while Figure 4.4 shows the light curve graphs of non-transient events which are incorrectly classified. In this case, non-transient objects that were correctly predicted tend to have a linear tendency when looked as a whole, which means that their observation magnitudes aren't located too far away from a mean value. This is specially the case with image (d) of Figure 4.3. Incorrectly classified non-transient light curves, on the other hand, tend to be very sparse, with a variety of magnitudes and error values, which might be the reason why these samples were incorrectly classified. It is important to notice that incorrectly classified non-transient objects look like correctly classified transients, while the opposite is also true.



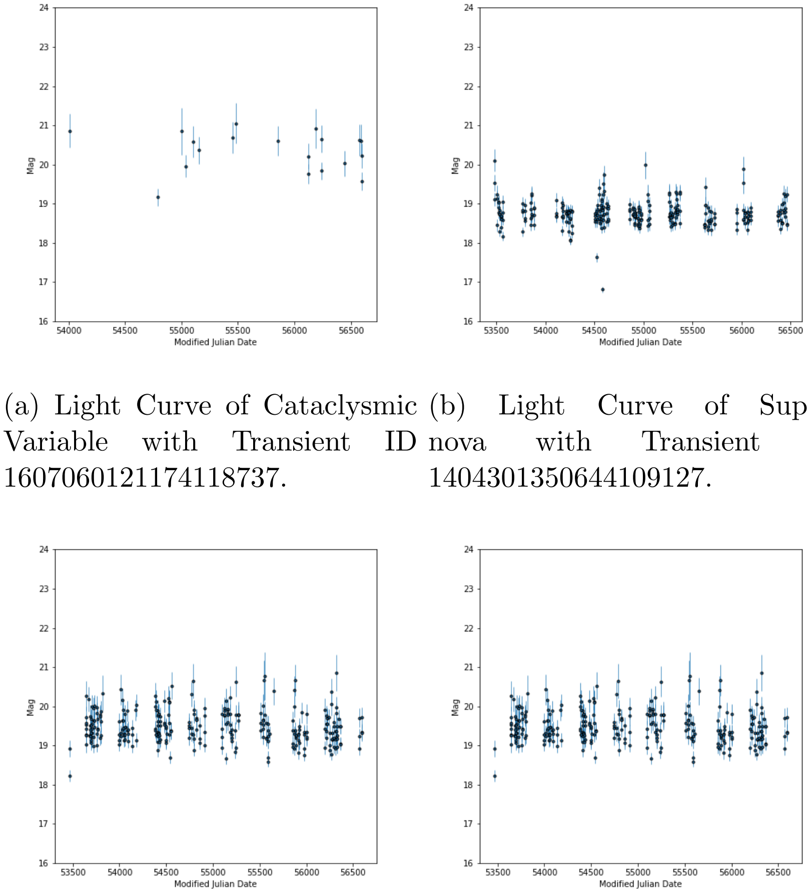
(a) Light Curve of Supernova with Transient ID 911211260084103595.

(b) Light Curve of Cataclysmic Variable Star with Transient ID 1306041151144145470.

(c) Light Curve of Cataclysmic Variable Star with Transient ID 904161120864151070.

(d) Light Curve of Supernova Transient ID 1202251090624111413.

Figure 4.1: Light Curves of four correctly classified transient objects.



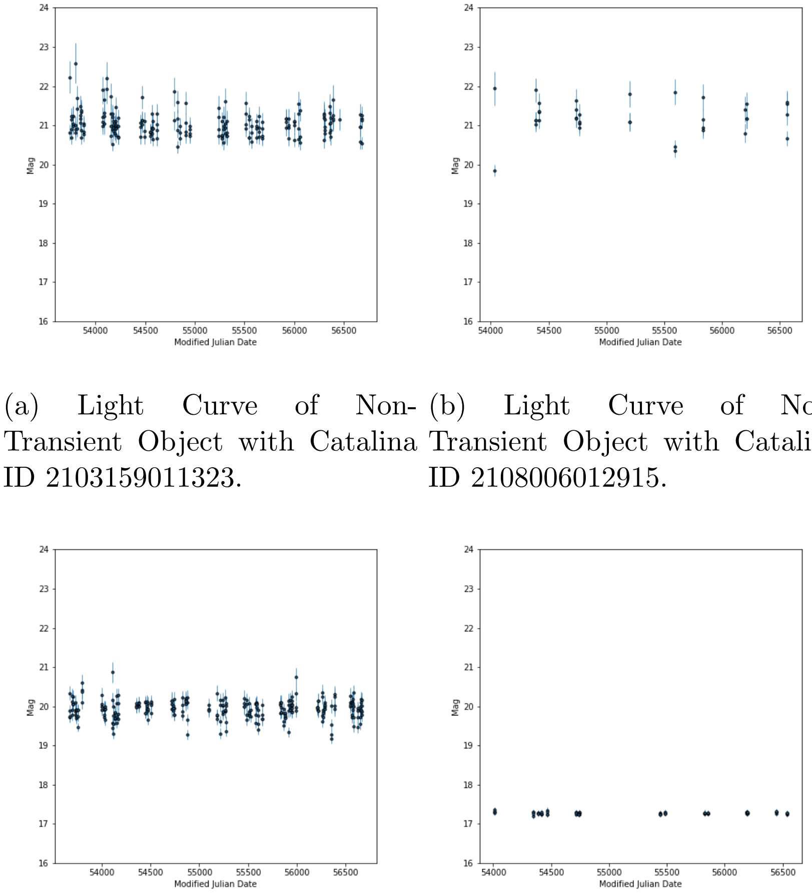
(a) Light Curve of Cataclysmic Variable with Transient ID 1607060121174118737.

(b) Light Curve of Supernova with Transient ID 1404301350644109127.

(c) Light Curve of Supernova with Transient ID 1603021070274145695.

(d) Light Curve of AGN? with Transient ID 1603021070274145695.

Figure 4.2: Light Curves of four incorrectly classified transient objects.



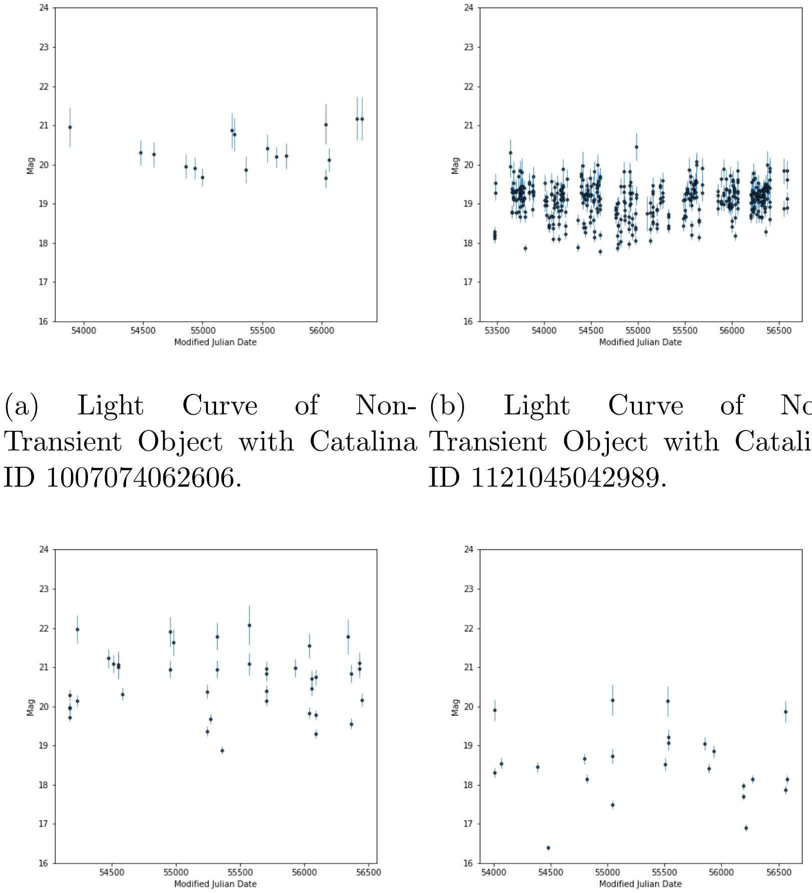
(a) Light Curve of Non-Transient Object with Catalina ID 2103159011323.

(b) Light Curve of Non-Transient Object with Catalina ID 2108006012915.

(c) Light Curve of Non-Transient Object with Catalina ID 2122055008174.

(d) Light Curve of Non-Transient Object with Catalina ID 2002316003404.

Figure 4.3: Light Curves of four correctly classified non-transient objects.



(a) Light Curve of Non-Transient Object with Catalina ID 1007074062606.

(b) Light Curve of Non-Transient Object with Catalina ID 1121045042989.

(c) Light Curve of Non-Transient Object with Catalina ID 2013203017370.

(d) Light Curve of Non-Transient Object with Catalina ID 1001128050211.

Figure 4.4: Light Curves of four incorrectly classified non-transient objects.

4.2 6-Transient Classification

The following tables present the Recall obtained after running the 6-transient classification experiment with the unbalanced and oversampled (balanced) data-sets. Results are shown for each classifier, with the data-set's light curves filtered by 5 and 10 observations minimum, and using 21, 27 and 31 features.

Classifier	Min. observations	Recall		
		21 feat.	27 feat.	31 feat.
SVM	5	67.69	68.31	68.31
	10	**69.59**	69.49	69.49
Random Forest	5	74.03	77.38	77.02
	10	74.70	**77.85**	77.56
Neural Network	5	71.30	70.60	71.92
	10	70.96	71.36	**74.21**

Table 4.3: Recall for the 6-Transient Classification Task with unbalanced classes. Top score for each classifier is in bold.

Classifier	Min. observations	Recall		
		21 feat.	27 feat.	31 feat.
SVM	5	69.01	68.57	68.40
	10	71.06	**71.16**	71.06
Random Forest	5	72.80	75.09	75.09
	10	72.34	**78.15**	78.05
Neural Network	5	70.07	69.19	70.42
	10	71.06	71.95	**72.64**

Table 4.4: Recall for the 6-Transient Classification Task with balanced classes. Top score for each classifier is in bold

The following information can be resumed from Tables 4.3 & 4.4:

- Top recall of 78.15% was obtained by the Random Forest when using balanced-class light curves with 10 observations min. and 31 features.
- Random Forest was the best classifier in all variations.
- Worst performing algorithm was the SVM, with the lowest scores when trained with unbalanced data.
- SVMs
  - Performed better with oversampled balanced data.

  - Number of features didn't seem to affect much on performance.

- Random forests
  - Performed significantly better with more than 21 features.
  - Performed slightly better with 27 features.
  - Performed better with balanced data, when using 10 obs. minimum. Otherwise it performs worse.

- Neural networks
  - Had overall better performance when using unbalanced data.
  - Performed better when using 10 obs. and 31 features.

The next tables present different scores on the classification performance of the best performing algorithm, when using both the balanced data (Random Forest, 10 obs, 27 feats) and the balanced oversampled data (Random Forest, 10 observations, 27 features).

	Precision	Recall	f1-Score	Support
AGN	84.78	82.78	83.87	141
Blazar	70.45	40.79	51.67	76
CV	80.25	75.58	77.84	258
Flare	65.00	41.94	50.98	62
HPM	96.24	96.97	96.60	132
SN	70.33	84.73	76.86	347
avg/total	77.91	77.85	77.19	1016

Table 4.5: Precision, Recall and f1-score for the 6-Transient Classification Task with unbalanced classes

	Precision	Recall	f1-Score	Support
AGN	81.21	85.82	83.45	141
Blazar	55.00	57.89	56.41	76
CV	82.70	75.97	79.19	258
Flare	53.66	70.97	61.11	62
HPM	94.12	96.97	95.52	132
SN	78.61	75.22	76.88	347
avg/total	78.74	78.15	78.31	1016

Table 4.6: Precision, Recall and f1-score for the 6-Transient Classification Task with balanced classes

The following information can be resumed from Tables 4.5 & 4.6:

- HPM was by far the best recognized class.
- AGN, CV, HPM and SN were the classes with the highest f1-score.
- Blazar and Flare were the classes with the lowest precision and recall.
- Average precision, recall and f1-score improved when using oversampled data.
- Blazar and Flare precision decreased with oversampled data, but their recall increased with the same data.

4.3 7-Transient Classification

The following tables presents the Recall obtained after running the 7-transient classification experiment with the unbalanced and oversampled (balanced) data-sets. Results are shown for each classifier, with the data-set's light curves filtered by 5 and 10 observations minimum, and using 21, 27 and 31 features.

Classifier	Min. observations	Recall		
		21 feat.	27 feat.	31 feat.
SVM	5	55.98	57.22	57.36
	10	59.19	**59.59**	59.43
Random Forest	5	62.13	65.31	65.31
	10	63.25	**66.91**	66.50
Neural Network	5	60.26	60.06	60.40
	10	62.11	**62.20**	61.54

Table 4.7: Recall for the 7-Transient Classification Task with unbalanced classes. Top score for each classifier is in bold.

Classifier	Min. observations	Recall		
		21 feat.	27 feat.	31 feat.
SVM	5	57.15	58.05	57.91
	10	**60.33**	60.00	59.84
Random Forest	5	61.85	64.89	64.34
	10	63.17	**67.24**	67.15
Neural Network	5	59.50	58.19	58.81
	10	59.92	**61.46**	61.22

Table 4.8: Recall for the 7-Transient Classification Task with balanced classes. Top score for each classifier is in bold.

The following information can be resumed from Tables 4.3 & 4.4:

- Top recall of 67.24% was obtained by the Random Forest when using balanced-class light curves with 10 observations min. and 27 features.
- Random Forest outperformed all other classifiers in all variations.
- Worst performing algorithm was the SVM, with the lowest scores when trained with 5 obs.
- In comparison to 6-Transient Classification, more observations were required for better performance, instead of more balanced data.
- SVMs

  - Performed better when using light curves with 10 observations min.
  - Increased performance when using oversampled data, for the same number of observations.
  - Number of features didn't seem to affect much on performance.

- Random forests
  - Performed significantly better with more than 21 features, and slightly better with 27 features rather than 31.
  - Using 10 observations min. scored higher performance when using oversampled data.

- Neural networks
  - Performance increased significantly when using 10 observations min.
  - Recall increased slightly when using unbalanced data.

The next tables present different scores on the classification performance of the best performing algorithm, when using both the balanced data (Random Forest, 10 obs, 27 feats) and the balanced oversampled data (Random Forest, 10 observations, 27 features).

	Precision	Recall	f1-Score	Support
AGN	76.52	71.63	73.99	141
Blazar	56.51	33.77	42.28	77
CV	77.20	74.81	75.98	258
Flare	69.23	43.55	53.47	62
HPM	92.75	96.97	94.81	132
Other	46.99	36.45	41.05	214
SN	58.82	78.03	67.08	346
avg/total	66.67	66.91	65.95	1230

Table 4.9: Precision, Recall and f1-score for the 7-Transient Classification Task with balanced classes

	Precision	Recall	f1-Score	Support
AGN	70.27	73.76	71.97	141
Blazar	44.83	50.65	47.56	77
CV	79.34	74.42	76.80	258
Flare	50.00	64.52	56.34	62
HPM	90.14	96.97	93.43	132
Other	48.70	43.93	46.19	214
SN	68.05	66.47	67.25	346
avg/total	67.31	67.24	67.16	1230

Table 4.10: Precision, Recall and f1-score for the 7-Transient Classification Task with balanced classes

The following information can be resumed from Tables 4.9 & 4.10:

- AGN, CV, HPM and SN were the classes with the highest f1-score.
- HPM was by far the best distinguished class.
- Blazar and Flare were the classes with the lowest precision and recall.
- Average precision, recall and f1-score improved when using oversampled data.
- Blazar and Flare precision decreased with overampled data, but their recall increases with the same data.
- Individual precision and recall was more similar when using balanced data.

4.4 7-Class Classification

The following tables present the Recall obtained after running the 7-transient classification experiment with the unbalanced and oversampled (balanced) data-sets. Results are shown for each classifier, with the data-set's light curves filtered by 5 and 10 observations minimum, and using 21, 27 and 31 features.

Classifier	Min. observations	Recall		
		21 feat.	27 feat.	31 feat.
SVM	5	63.79	64.68	63.98
	10	**66.59**	65.59	65.79
Random Forest	5	70.83	75.24	74.92
	10	71.73	**77.31**	77.24
Neural Network	5	68.27	68.59	68.07
	10	**69.68**	69.53	**69.68**

Table 4.11: Recall for the 7-Class Classification Task with unbalanced classes. Top score for each classifier is in bold.

Classifier	Min. observations	Recall		
		21 feat.	27 feat.	31 feat.
SVM	5	65.07	**66.22**	66.03
	10	65.79	65.49	65.57
Random Forest	5	68.07	72.62	72.87
	10	69.38	75.92	**76.06**
Neural Network	5	63.53	65.32	65.20
	10	**66.64**	66.01	66.08

Table 4.12: Recall for the 7-Class Classification Task with balanced classes. Top score for each classifier is in bold.

The following information can be resumed from Tables 4.11 & 4.12:

- Top recall of 77.31% was obtained by the Random Forest when using unbalanced-class light curves with 10 observations min. and 27 features.
- Random Forest outperformed all classifiers, in all variations.
- Worst performing algorithm was the SVM, with the lowest scores when trained with unbalanced data.
- SVMs
  - Performed better with balanced data and more than 21 features.
  - Performed better with min. 5 observations per light curve.

- Random forests
  - Performed significantly better with more than 21 features.
  - Performed better with unbalanced data.
  - Performed better when using 10 obs. minimum.
- Neural networks
  - Had overall better performance when using unbalanced data.
  - Performed better when using 10 obs. minimum.

The next tables present different scores on the classification performance of the best performing algorithm, when using both the balanced data (Random Forest, 10 obs, 27 feats) and the balanced oversampled data (Random Forest, 10 observations, 27 features).

	Precision	Recall	f1-Score	Support
AGN	83.46	75.18	79.10	141
Blazar	70.27	33.77	45.61	77
CV	81.22	77.13	79.13	258
Flare	70.59	38.71	50.00	62
HPM	95.12	88.64	91.76	132
Non-Transient	72.73	92.49	81.42	346
SN	73.31	75.43	74.36	346
avg/total	77.53	77.31	76.50	1362

Table 4.13: Precision, Recall and f1-score for the 7-Class Classification Task with unbalanced classes.

	Precision	Recall	f1-Score	Support
AGN	80.82	83.69	82.23	141
Blazar	61.11	57.14	59.06	77
CV	83.26	75.19	79.02	258
Flare	38.46	56.45	45.75	62
HPM	85.21	91.67	88.32	132
Non-Transient	78.24	82.08	80.11	346
SN	76.19	69.36	72.62	346
avg/total	76.83	76.06	76.26	1362

Table 4.14: Precision, Recall and f1-score for the 7-Class Classification Task with balanced classes

The following information can be resumed from Tables 4.13 & 4.13:

- AGN, CV, HPM and SN were the classes with the highest f1-score.
- HPM was by far the best distinguished class.
- Blazar and Flare were the classes with the lowest precision and recall.
- Average precision, recall and f1-score decreased with oversampled data.
- Non-Transient, Blazar and Flare precision decreased with balanced data, but their recall increase with the same data.
- Random Forest trained with unbalanced data was only slightly better than the one trained with balanced data.

4.5 8-Class Classification

The following tables presents the Recall obtained after running the 7-transient classification experiment with the unbalanced and oversampled (balanced) data-sets. Results are shown for each classifier, with the data-set's light curves filtered by 5 and 10 observations minimum, and using 21, 27 and 31 features.

Classifier	Min. observations	Recall		
		21 feat.	27 feat.	31 feat.
SVM	5	55.20	54.77	54.67
	10	58.66	58.97	**59.16**
Random Forest	5	62.03	64.37	65.12
	10	63.60	**68.23**	68.17
Neural Network	5	59.15	59.41	58.88
	10	61.32	61.13	**62.14**

Table 4.15: Recall for the 8-Class Classification Task with unbalanced classes. Top score for each classifier is in bold.

Classifier	Min. observations	Recall		
		21 feat.	27 feat.	31 feat.
SVM	5	55.68	56.11	56.11
	10	58.21	58.34	**58.40**
Random Forest	5	60.11	62.77	62.40
	10	61.95	67.09	**67.47**
Neural Network	5	55.09	55.79	54.13
	10	59.54	**60.11**	59.48

Table 4.16: Recall for the 8-Class Classification Task with balanced classes. Top score for each classifier is in bold.

The following information can be resumed from Tables 4.15 & 4.16:

- Top recall of 67.23% was obtained by the Random Forest when using balanced-class light curves with 10 observations min. and 27 features.
- Random Forest was the best classifier in all variations.
- Worst performing algorithm was the SVM, with the lowest scores when trained with unbalanced data.
- SVMs
  - Using 10 observations min. increases performance.

  - Using 10 observations min. had higher performance when using unbalanced classes.
  - Recall obtained with 27 - 31 features was very similar, and was better than using 21.

- Random forests
  - Much better performance than other algorithms.
  - Second top score of 68.17% using balanced-class light curves with 10 observations min. and 27 features.
  - Performed better with unbalanced data.
  - Performed significantly better with more than 21 features.

- Neural networks
  - Better recall when using 10 obs.
  - Higher performance when using unbalanced data.
  - Number of features used don't affect much on performance.

The next tables present different scores on the classification performance of the best performing algorithm, when using both the balanced data (Random Forest, 10 obs, 27 feats) and the balanced oversampled data (Random Forest, 10 observations, 27 features).

	Precision	Recall	f1-Score	Support
AGN	71.53	73.05	72.28	141
Blazar	66.67	34.21	45.22	76
CV	76.36	76.36	76.36	258
Flare	65.71	37.10	47.42	62
HPM	92.91	89.39	91.12	132
Non-Transient	67.25	89.34	76.73	347
Other	48.00	33.64	39.56	214
SN	62.53	65.42	63.94	347
avg/total	67.53	68.23	66.95	1577

Table 4.17: Precision, Recall and f1-score for the 8-Class Classification Task with unbalanced classes.

	Precision	Recall	f1-Score	Support
AGN	67.30	75.89	71.33	141
Blazar	54.93	51.32	53.06	76
CV	80.41	76.36	78.33	258
Flare	37.36	51.84	44.44	62
HPM	82.88	91.67	87.05	132
Non-Transient	72.32	79.83	75.89	347
Other	45.79	40.65	43.07	214
SN	69.18	58.21	63.22	347
avg/total	67.57	71.33	67.24	1577

Table 4.18: Precision, Recall and f1-score for the 8-Class Classification Task with balanced classes.

The following information can be resumed from Tables 4.17 & 4.18:

- AGN, CV, HPM and Non-Transients were the classes with the highest f1-score.
- Other, Blazar and Flare were the classes with the lowest precision and recall.
- Average precision, recall and f1-score improved slightly when using oversampled data.
- Blazar and Flare precision decreased with oversampled data, but their recall increased with the same data.
- Non-Transient recall decreased in about 10% with oversampled data.

4.6 Consolidated Results

The following results were obtained after analyzing individual outcomes of the five experiments. Regarding the classification algorithms:

- All trained models except the SVM in Table 4.11 performed better when using light curves with 10 observations min.
- In all tasks, Random Forests were the best performing algorithm.

- Overall, Random Forests performed better when using 27 features; using 31 features give similar results.
- In general, SVM were the worst performing algorithm for all tasks.
- SVMs performed similarly when using 21, 27 and 31 features.
- Neural Networks performed better most of the time when using 27-31 features.
- Neural Networks performed worse with balanced data.

Concerning the results obtained on the classification tasks:

- For the binary classification task, Transient class recognition scored a higher recall than non-transient recognition.
- For the binary classification task, using light curves with observations and features improved the recall.
- In all multi-class classification tasks, AGN, CV and HPM were the classes with higher scores.
- In all multi-class tasks that included the Other class (6-Transient & 7-Transient Classification), such class was the worst performing one.
- In all multi-class tasks that included the Non-Transient class (7-Class & 8-Class Classification), all models performed worse when using balanced data.
- Introducing the Other class in the 7-Transient task decreased significantly the performance, in comparison to the 6-Transient classification task.
- Introducing the Non-Transient class in the 7-Class task decreased slightly the performance ($\sim 2\%$ precision and recall), in comparison to the 6-Transient classification task.
- Introducing the Other class in the 8-Class task decreased slightly the performance, in comparison to the 7-Transient classification task.
- Introducing the Non-Transient class in the 8-Class task decreased slightly the performance, in comparison to the 7-Class classification task.

Chapter 5

Conclusions and Future Work

In this chapter, analyses of the results obtained in Chapter 4 are presented through a set of conclusions (Section 5.1) and future work recommendations are also described (Section 5.2). The conclusions presented here respond to how well the project objectives stated in the introduction were achieved, and the group of improvements for the methodology proposed in this project provide following actions that could potentially boost the performance of transient event recognition.

5.1 Conclusions

This project presents an approach for the automatic recognition of transient events with the use of machine learning techniques. Such proposal was developed under the scope of forthcoming astronomical synaptic surveys such as the LSST.

The method introduced in this project consists in oversampling filtered light curves, then extracting characteristic features from them, and then using those features as inputs to machine learning algorithms. The features extracted from light curves are either statistical descriptors of the observations, or coefficients obtained from polynomial curve fittings applied to the light curves. Mentioned machine learning algorithms were trained with the resulting features, so that they could perform classification with high reliability.

Three machine learning models tested with the data mentioned in a variety of classification experiments. These experiments consisted in training and testing binary classification among transients and non-transients, and multi-class classification among several types of transients (sometimes including transients). Detailed description of these tasks can be found in Section 3.5. Overall, the best classifier for all tasks was the Random Forest, followed by Neural Networks and then Support Vector Machines.

State of the art results were obtained when testing the trained models to unseen sets of data, specifically in binary and transient multi-class classification. Recall scores obtained from the best classifier for each task, are equivalent to those results found in [27] and [30]. This implies that the methodology proposed in this project, which is a new methodology proposal, work correctly for the classification tasks propounded. The detailed scores obtained for each task were:

- Binary Classification: 89.39% recall.
- 6-Transient Classification: 78.15% recall.
- 7-Transient Classification: 67.24% recall.
- 7-Class Classification: 77.31% recall.
- 8-Class Classification: 68.23% recall.

Regarding the parameters introduced in this project (e.g. min. number of observations per light curve, num. features, etc), it is evident that training with light curves that have 10 observations minimum always outperforms using 5 observations. This seems to be true in all scenarios, due to the lower noise that curves with more observations provide. Additionally, using 27 features per light curve seems to give the best results overall when classifying only transient sources, while using 31 features works better when including non-transients to the classification tasks. A reason for this behaviour may be that non-transient light curves have a higher observation mean than transients, and higher rank polynomial fitting adjusts better to these curves. Finally, it is important to note that using oversampled light curves rather than the original ones works better when doing transient recognition only. As non-transient light curves are never oversampled (only sub-sampled), classifiers might over-fit on transient recognition, giving lower results.

The methodology proposed in this project, together with the positive results lay the foundations on the development of more robust Transient Object

Classifiers. This research becomes a base for the work of Research groups at Universidad de los Andes and CCPM keep exploring the field, groups that also look forward to contribute to the new age of synoptic surveys.

5.2 Future Work

Though the results obtained demonstrate that the methodology proposed works well, the scores obtained are far from perfect. Results are not significantly better than what the state of the art already obtained, which means that there are still more improvements to develop in order to obtain better classifiers.

One of the hardest tasks in transient recognition using photometric data is obtaining clean information. Astronomical data-sources tend to be obscure and hard to deal with, which makes it difficult to understand the information that is being gathered. Having said that, a critical improvement on transient recognition using their light curves would be to train the models with cleaner data, this means, data with lower amount of duplicates and errors, and using a more reliable non-transient data-set. Additionally, it would be beneficial to have a data-set containing more light curves, with more balanced classes. Several transient types in this project contain less than 1 light curve, which is not nearly enough to have correct classification of those, and which is why only 6 transient classes were clearly labeled. Artificially balancing classes with techniques like the one presented in this project may slightly improve performance, but counting with more real sources would significantly improve it.

Using additional features could also benefit the performance of the classifiers. Specifically, using more statistical curve descriptors such as the ones in [30], which have already shown success cases, may improve classification even further. Moreover, a variety of curve-fitting techniques can be used to have different representations of the light curves which may fit better than the polynomial ones, thus closing the gap towards a perfect classification.

Regarding the classification techniques, new methodologies can also be tested for a better detection of transient objects. A more detailed hyper-parameter grid search can be applied in order to find the values of those parameters closest to the global maximum performance. Furthermore, since the algorithms tested in this project classified transient objects in different ways, their results could be mixed into a single bagging algorithm which performs much better than its individual parts.

Appendix A

Program repository

In this appendix, the program repository used to fulfill the objectives proposed in Chapter 1 is described. The main project characteristics are described first (Section A.1). Following, the folder structure is presented, taking into account that all paths shown are relative to the main folder of the repository (Section A.2. Finally, a description of the notebooks used to implement the methodology proposed in Chapter 3, concludes this appendix (Section A.3).

A.1 Program characteristics

All the code developed in this project was built using Python 3.4. It is contained a git repository, which has been uploaded to the GitHub platform; it may be found and downloaded from `http://github.com/diegoalejogm/crts-transient-recognition`.

The main functionality of the program was built using Jupyter Notebooks (`http://jupyter.org/`). This platform allows sharing notebooks that run as web applications and contain live code with narrative text. The project's principal functionality is then divided in various notebook files, all of which are well documented and provide a quick learning curve on how the project internals work. Most notebooks file names start with numbers for sequential execution of the methodology proposed in Chapter 3, and those that start with labels explore the data-set described in Chapter 2.

A running Jupyter Notebook environment can be found within Universidad de los Andes's local network, and is hosted in the domain `http:`

`//astro.virtual.uniandes.edu.co:8080/`

The notebooks developed use several dependencies, which are all listed in the file named *requirements.txt*, found in the repository's main directory. As a suggestion, all these dependencies can be downloaded using the 'pip' package management system.

A.2 Folder Structure

The main project folder contains files and folders. All files present in this directory concern the configuration of the project. Conversely, the three directories contain fundamental files required for the program execution. Each of these folders is described next:

- **notebooks**: This directory contains all Jupyter Notebooks used for the execution of the data pipeline and classification framework. It also contains auxiliary code files, all of which are documented, that are used by the notebooks. Additional notebooks for data exploration can be found in the */exploration/* subfolder.
- **data**: This directory contains the files with the photometric data, and the transient catalogue information. Light curves information may be found in the *lightcurves* subfolder, and feature vector data may be found in the *features* subfolder once it has been created by the execution pipeline. All files in this folder are named explicitly regarding the information they contain.
- **results**: This directory contains one file for the results of each experiment executed.

A.3 Notebooks

A description of the most relevant notebooks files is presented next:

- *1. Filtering & Feature Extraction.ipynb*: In this notebook, data-sets are loaded and filtered by the manually specified number of observations and features to use, as explained in Section 3.1. Then, features are extracted using this data and resulting files are saved in the *features* sub-folder of the *data* directory.

- *2. Oversampling & Feature Extraction*: In this notebook, data-sets are loaded and filtered by the manually specified number of observations and features to use, as explained in Section 3.1. Then, light curves are oversampled to balance classes (Section 3.2). Finally, features are extracted from the resulting oversampled data-set, and they're saved in the *features* sub-folder of the *data* directory.

- *3. Pre-Processing & Classification*: In this notebook, all model training and testing experiments are executed. The sub-set of parameter variations is selected in the beginning, and then the experimental framework (Section 3.6) is executed on every combination of parameters selected. Results are stored in the global *results* folder.

Appendix B

Experiments repository

During the first weeks of the research, various data-sets were explored and tested as sources for this project, in order to decide which data-set to use for transient recognition. A repository containing all the experiments used in this research to select the final data source was created during that time, and the current appendix briefly describes its contents. The overall program characteristics are explained first (Section B.1), and specific folder content structure is explored next B.2.

Experimentation includes data download, image visualization, meta-data extraction, among other tasks.

B.1 Program characteristics

All the code developed in this project was built using Python 3.4. It is contained a git repository, which has been uploaded to the GitHub platform; it may be found and downloaded from `https://github.com/diegoalejogm/transient-recognition-experiments`.

The main functionality of the program was built using Jupyter Notebooks (`http://jupyter.org/`) too (see Appendix A). Experimentation of each data-source's is contained in a separate folder, each of which contains various *notebook* files. These notebooks are documented and their file names describe their functionality.

All experiments use various dependencies, and they're all listed in a single file named *requirements.txt*, found in the repository's main directory.

B.2 Folder Structure

The main folder contains a single directory named *notebooks*. Within it, multiple directories are found, each of which contains the code required for experimentation with a specific data source. These folders is described next:

- **CRTS**: This directory contains all Jupyter Notebooks used for the experimentation of a different sub-set of the CRTS data. Experimentation tasks found in this folder are download, exploration and visualization of the data.

- **CRTS2**: This directory contains all Jupyter Notebooks used for the experimentation of the CRTS data described in the Chapter 2 of this project. It contains additional notebooks regarding data consolidation.

- **PESSTO**: This directory contains Jupyter Notebooks that explore the Public ESO Spectroscopic Survey of Transient Objects (PESSTO) data-set [42]. All these notebooks examine the catalogue's images and associated meta-data.

- **PTF**: This directory contains the Jupyter Notebooks that explore the data-set of the Palomar Transient Factory's catalogue [43].

Bibliography

[1] S. G. Djorgovski, C. i Donalek, A. Mahabal, B. Moghaddam, M. Turmon, M. Graham, A. Drake, N. Sharma, and Y. Chen. Towards an Automated Classification of Transient Events in Synoptic Sky Surveys. *ArXiv e-prints*, October 2011.

[2] Frederick R. Chromey. *To Measure the Sky: An Introduction to Observational Astronomy*. Cambridge University Press, 2010.

[3] Bradley W. Carroll and Dale A. Ostlie. *An Introduction to Modern Astrophysics (2nd Edition)*. Pearson, 2006.

[4] NASA. Light curves and what they can tell us. `https://imagine.gsfc.nasa.gov/science/toolbox/timing1.html`, August 2013.

[5] American Association of Variable Star Observers. About light curves. `https://www.aavso.org/about-light-curves`, May 2010.

[6] D. G. York. The sloan digital sky survey: Technical summary. *ArXiv e-prints*, 2000.

[7] J. E. Gizis and M. F. Skrutskie. The Two Micron All-Sky Survey: Removing the Infrared Foreground. In M. Harwit and M. G. Hauser, editors, *The Extragalactic Infrared Background and its Cosmological Implications*, volume 204 of *IAU Symposium*, page 197, January 2001.

[8] D. K. Nadyozhin. Physics of supernovae: theory, observations, unresolved problems, 2008.

[9] Christian Knigge. The evolution of cataclysmic variables, 2011.

[10] Henric Krawczynski and Ezequiel Treister. Active galactic nuclei - the physics of individual sources and the cosmic history of formation and evolution, 2013.

[11] J. Isler. How i fell in love with quasars, blazars and our incredible universe. `https://www.ted.com/talks/jedidah_isler_how_i_fell_in_love_with_quasars_blazars_and_our_incredible_universe#t-242069`, 2015. Accessed 10/09/17.

[12] Kevin P. Murphy. *Machine learning: a probabilistic perspective.* MIT Press, 2013.

[13] Stuart J. Russell and Peter Norvig. *Artificial intelligence: A Modern Approach.* Prentice Hall, 2009.

[14] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction.* The MIT Press, 2012.

[15] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning.* The MIT Press, 2016.

[16] Christopher M. Bishop. *Pattern recognition and machine learning.* Springer, 2011.

[17] R. Kessler, J. Marriner, M. Childress, R. Covarrubias, C. B. D'Andrea, D. A. Finley, J. Fischer, R. J. Foley, D. Goldstein, R. R. Gupta, K. Kuehn, M. Marcha, R. C. Nichol, A. Papadopoulos, M. Sako, D. Scolnic, M. Smith, M. Sullivan, W. Wester, F. Yuan, T. Abbott, F. B. Abdalla, S. Allam, A. Benoit-Levy, G. M. Bernstein, E. Bertin, D. Brooks, A. Carnero Rosell, M. Carrasco Kind, F. J. Castander, M. Crocce, L. N. da Costa, S. Desai, H. T. Diehl, T. F. Eifler, A. Fausti Neto, B. Flaugher, J. Frieman, D. Gruen, R. A. Gruendl, K. Honscheid, D. J. James, N. Kuropatkin, T. S. Li, M. A. G. Maia, J. L. Marshall, P. Martini, C. J. Miller, R. Miquel, R. Ogando, A. A. Plazas, A. K. Romer, A. Roodman, E. Sanchez, I. Sevilla-Noarbe, R. C. Smith, M. Soares-Santos, F. Sobreira, G. Tarle, J. Thaler, R. C. Thomas, D. Tucker, and A. R. Walker. The difference imaging pipeline for the transient search in the dark energy survey. *ArXiv e-prints*, 2015.

[18] Frank Masci, Russ Laher, Umaa Rebbapragada, Gary Doran, Adam Miller, Eric Bellm, Mansi Kasliwal, Eran Ofek, Jason Surace, David

Shupe, Carl Grillmair, Ed Jackson, Tom Barlow, Lin Yan, Yi Cao, S. Bradley Cenko, Lisa Storrie-Lombardi, George Helou, Thomas Prince, and Shrinivas Kulkarni. The ipac image subtraction and discovery pipeline for the intermediate palomar transient factory. *ArXiv e-prints*, 2016.

[19] Hung-Yu Jian, Lihwai Lin, Kai-Yang Lin, Sebastien Foucaud, Chin-Wei Chen, Tzihong Chiueh, R. G. Bower, Shaun Cole, Wen-Ping Chen, W. S. Burgett, P. W. Draper, H. Flewelling, M. E. Huber, N. Kaiser, R. P. Kudritzki, E. A. Magnier, N. Metcalfe, R. J. Wainscoat, and C. Waters. The pan-starrs1 medium-deep survey: Star formation quenching in group and cluster environments. *ArXiv e-prints*, 2017.

[20] C. Alard and R. H. Lupton. A method for optimal image subtraction. *ArXiv e-prints*, 1997.

[21] Fabian Gieseke, Steven Bloemen, Cas van den Bogaard, Tom Heskes, Jonas Kindler, Richard A. Scalzo, Valério A. R. M. Ribeiro, Jan van Roestel, Paul J. Groot, Fang Yuan, Anais Möller, and Brad E. Tucker. Convolutional neural networks for transient candidate vetting in large-scale surveys. *ArXiv e-prints*, 2017.

[22] Guillermo Cabrera-Vives, Ignacio Reyes, Francisco Förster, Pablo A. Estévez, and Juan-Carlos Maureira. Deep-hits: Rotation invariant convolutional neural network for transient detection. *ArXiv e-prints*, 2017.

[23] Jakub Klencki and Łukasz Wyrzykowski. Real-time detection of transients in ogle-iv with application of machine learning, 2016.

[24] Jakub Klencki, Łukasz Wyrzykowski, Zuzanna Kostrzewa-Rutkowska, and Andrzej Udalski. Robust filtering of artifacts in difference imaging for rapid transients detection, 2016.

[25] D. E. Wright, S. J. Smartt, K. W. Smith, P. Miller, R. Kotak, A. Rest, W. S. Burgett, K. C. Chambers, H. Flewelling, K. W. Hodapp, M. Huber, R. Jedicke, N. Kaiser, N. Metcalfe, P. A. Price, J. L. Tonry, R. J. Wainscoat, and C. Waters. Machine learning for transient discovery in pan-starrs1 difference imaging. *ArXiv e-prints*, 2015.

[26] L. du Buisson, N. Sivanandam, B. A. Bassett, and M. Smith. Machine learning classification of sdss transient survey images. *ArXiv e-prints*, 2014.

[27] Julian Faraway, Ashish Mahabal, Jiayang Sun, Xiaofeng Wang, Yi, Wang, and Lingsong Zhang. Modeling light curves for improved classification. *ArXiv e-prints*, 2014.

[28] R. Downes et al. A catalog and atlas of cataclysmic variables: The ?nal edition. *Journal of Astronomical Data*, 11:2, 2005.

[29] Joseph W. Richards, Dan L. Starr, Nathaniel R. Butler, Joshua S. Bloom, John M. Brewer, Arien Crellin-Quick, Justin Higgins, Rachel Kennedy, and Maxime Rischard. On machine-learned classification of variable stars with sparse and noisy time-series data. *ArXiv e-prints*, 2011.

[30] Antonio D'Isanto, Stefano Cavuoti, Massimo Brescia, Ciro Donalek, Giuseppe Longo, Giuseppe Riccio, and Stanislav G. Djorgovski. An analysis of feature relevance in the classification of astronomical transients with machine learning methods. *ArXiv e-prints*, 2016.

[31] Michelle Lochner, Jason D. McEwen, Hiranya V. Peiris, Ofer Lahav, and Max K. Winter. Photometric supernova classification with machine learning. *ArXiv e-prints*, 2016.

[32] Tom Charnock and Adam Moss. Deep recurrent neural networks for supernovae classification. *ArXiv e-prints*, 2016.

[33] Trisha Hinners, Kevin Tat, and Rachel Thorp. Machine learning techniques for stellar light curve classification, 2017.

[34] Joshua S. Bloom and Joseph W. Richards. Data Mining and Machine-Learning in Time-Domain Discovery & Classification, 2011.

[35] Z. Ivezic, J. A. Tyson, B. Abel, E. Acosta, R. Allsman, Y. AlSayyad, S. F. Anderson, J. Andrew, R. Angel, G. Angeli, R. Ansari, P. Antilogus, K. T. Arndt, P. Astier, E. Aubourg, T. Axelrod, D. J. Bard, J. D. Barr, A. Barrau, J. G. Bartlett, B. J. Bauman, S. Beaumont, A. C. Becker, J. Becla, C. Beldica, S. Bellavia, G. Blanc, R. D. Blandford, J. S. Bloom, J. Bogart, K. Borne, J. F. Bosch, D. Boutigny,

W. N. Brandt, M. E. Brown, J. S. Bullock, P. Burchat, D. L. Burke, G. Cagnoli, D. Calabrese, S. Chandrasekharan, S. Chesley, E. C. Cheu, J. Chiang, C. F. Claver, A. J. Connolly, K. H. Cook, A. Cooray, K. R. Covey, C. Cribbs, W. Cui, R. Cutri, G. Daubard, G. Daues, F. Delgado, S. Digel, P. Doherty, R. Dubois, G. P. Dubois-Felsmann, J. Durech, M. Eracleous, H. Ferguson, J. Frank, M. Freemon, E. Gangler, E. Gawiser, J. C. Geary, P. Gee, M. Geha, R. R. Gibson, D. K. Gilmore, T. Glanzman, I. Goodenow, W. J. Gressler, P. Gris, A. Guyonnet, P. A. Hascall, J. Haupt, F. Hernandez, C. Hogan, D. Huang, M. E. Huffer, W. R. Innes, S. H. Jacoby, B. Jain, J. Jee, J. G. Jernigan, D. Jevremovic, K. Johns, R. L. Jones, C. Juramy-Gilles, M. Juric, S. M. Kahn, J. S. Kalirai, N. Kallivayalil, B. Kalmbach, J. P. Kantor, M. M. Kasliwal, R. Kessler, D. Kirkby, L. Knox, I. Kotov, V. L. Krabbendam, S. Krughoff, P. Kubanek, J. Kuczewski, S. Kulkarni, R. Lambert, L. Le Guillou, D. Levine, M. Liang, K-T. Lim, C. Lintott, R. H. Lupton, A. Mahabal, P. Marshall, S. Marshall, M. May, R. McKercher, M. Migliore, M. Miller, D. J. Mills, D. G. Monet, M. Moniez, D. R. Neill, J-Y. Nief, A. Nomerotski, M. Nordby, P. O'Connor, J. Oliver, S. S. Olivier, K. Olsen, S. Ortiz, R. E. Owen, R. Pain, J. R. Peterson, C. E. Petry, F. Pierfederici, S. Pietrowicz, R. Pike, P. A. Pinto, R. Plante, S. Plate, P. A. Price, M. Prouza, V. Radeka, J. Rajagopal, A. Rasmussen, N. Regnault, S. T. Ridgway, S. Ritz, W. Rosing, C. Roucelle, M. R. Rumore, S. Russo, A. Saha, B. Sassolas, T. L. Schalk, R. H. Schindler, D. P. Schneider, G. Schumacher, J. Sebag, G. H. Sembroski, L. G. Seppala, I. Shipsey, N. Silvestri, J. A. Smith, R. C. Smith, M. A. Strauss, C. W. Stubbs, D. Sweeney, A. Szalay, P. Takacs, J. J. Thaler, R. Van Berg, D. Vanden Berk, K. Vetter, F. Virieux, B. Xin, L. Walkowicz, C. W. Walter, D. L. Wang, M. Warner, B. Willman, D. Wittman, S. C. Wolff, W. M. Wood-Vasey, P. Yoachim, H. Zhan, and for the LSST Collaboration. Lsst: from science drivers to reference design and anticipated data products, 2008.

[36] A. J. Drake, S. G. Djorgovski, A. Mahabal, J. L. Prieto, E. Beshore, M. J. Graham, M. Catalan, S. Larson, E. Christensen, C. Donalek, and R. Williams. The catalina real-time transient survey. *ArXiv e-prints*, 2011.

[37] The University of Arizona. This is a test entry of type @ON-

LINE. https://catalina.lpl.arizona.edu/about/facilities/telescopes.

[38] P. B. Stetson. On the Automatic Determination of Light-Curve Parameters for Cepheid Variables. *pasp*, 108:851, October 1996.

[39] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)*. Springer, 2016.

[40] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2016.

[41] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *ArXiv e-prints*, 2012.

[42] S. J. Smartt, S. Valenti, M. Fraser, C. Inserra, D. R. Young, M. Sullivan, A. Pastorello, S. Benetti, A. Gal-Yam, C. Knapic, M. Molinaro, R. Smareglia, K. W. Smith, S. Taubenberger, O. Yaron, J. P. Anderson, C. Ashall, C. Balland, C. Baltay, C. Barbarino, F. E. Bauer, S. Baumont, D. Bersier, N. Blagorodnova, S. Bongard, M. T. Botticella, F. Bufano, M. Bulla, E. Cappellaro, H. Campbell, F. Cellier-Holzem, T. W. Chen, M. J. Childress, A. Clocchiatti, C. Contreras, M. Dall Ora, J. Danziger, T. de Jaeger, A. De Cia, M. Della Valle, M. Dennefeld, N. Elias-Rosa, N. Elman, U. Feindt, M. Fleury, E. Gall, S. Gonzalez-Gaitan, L. Galbany, A. Morales Garoffolo, L. Greggio, L. L. Guillou, S. Hachinger,

E. Hadjiyska, P. E. Hage, W. Hillebrandt, S. Hodgkin, E. Y. Hsiao, P. A. James, A. Jerkstrand, T. Kangas, E. Kankare, R. Kotak, M. Kromer, H. Kuncarayakti, G. Leloudas, P. Lundqvist, J. D. Lyman, I. M. Hook, K. Maguire, I. Manulis, S. J. Margheim, S. Mattila, J. R. Maund, P. A. Mazzali, M. McCrum, R. McKinnon, M. E. Moreno-Raya, M. Nicholl, P. Nugent, R. Pain, M. M. Phillips, G. Pignata, J. Polshaw, M. L. Pumo, D. Rabinowitz, E. Reilly, C. Romero-Canizales, R. Scalzo, B. Schmidt, S. Schulze, S. Sim, J. Sollerman, F. Taddia, L. Tartaglia, G. Terreran, L. Tomasella, M. Turatto, E. Walker, N. A. Walton, L. Wyrzykowski, F. Yuan, and L. Zampieri. Pessto : survey description and products from the first data release by the public eso spectroscopic survey of transient objects. *ArXiv e-prints*, 2014.

[43] J. Surace, R. Laher, F. Masci, C. Grillmair, and G. Helou. The palomar transient factory: High quality realtime data processing in a cost-constrained environment, 2015.