



PLAsTiCC Astronomical Classification

CAN YOU HELP MAKE SENSE OF THE UNIVERSE?

Gene Olafsen

Introduction

- ▶ Help some of the world's leading astronomers grasp the deepest properties of the universe.
- ▶ The human eye has been the arbiter for the classification of astronomical sources in the night sky for hundreds of years. But a new facility -- the [Large Synoptic Survey Telescope \(LSST\)](#) -- is about to revolutionize the field, discovering 10 to 100 times more astronomical sources that vary in the night sky than we've ever known. Some of these sources will be completely unprecedented!

Dataset Source

- ▶ The Photometric LSST Astronomical Time-Series Classification Challenge (PLAsTiCC) asks Kagglers to help prepare to classify the data from this new survey. Competitors will classify astronomical sources that vary with time into different classes, scaling from a small training set to a very large test set of the type the LSST will discover.

Features

- ▶ `object_id`: unique object identifier. Int32
- ▶ `ra`: right ascension, sky coordinate: co-longitude in degrees. Float32
- ▶ `dec1`: declination, sky coordinate: co-latitude in degrees. Float32
- ▶ `gal_l`: galactic longitude in degrees. Float32
- ▶ `gal_b`: galactic latitude in degrees. Float32
- ▶ `ddf`: A flag to identify the object as coming from the DDF survey area (with value DDF = 1 for the DDF, DDF = 0 for the WFD survey). Note that while the DDF fields are contained within the full WFD survey area, the DDF fluxes have significantly smaller uncertainties. Boolean

Features (cont.)

- ▶ `hostgal_specz`: the spectroscopic redshift of the source. This is an extremely accurate measure of redshift, available for the training set and a small fraction of the test set. Float32
- ▶ `hostgal_photoz`: The photometric redshift of the host galaxy of the astronomical source. While this is meant to be a proxy for `hostgal_specz`, there can be large differences between the two and should be regarded as a far less accurate version of `hostgal_specz`. Float32
- ▶ `hostgal_photoz_err`: The uncertainty on the `hostgal_photoz` based on LSST survey projections. Float32

Features (cont.)

- ▶ `distmod`: The distance to the source calculated from `hostgal_photoz` and using general relativity.
Float32
- ▶ `mwebv`: MW E(B-V). this 'extinction' of light is a property of the Milky Way (MW) dust along the line of sight to the astronomical source, and is thus a function of the sky coordinates of the source `ra`, `decl`. This is used to determine a passband dependent dimming and redenning of light from astronomical sources as described in subsection 2.1, and based on the [Schlafly et al. \(2011\)](#) and [Schlegel et al. \(1998\)](#) dust models.
Float32

Features (cont.)


- ▶ **target:** The class of the astronomical source. This is provided in the training data. Correctly determining the target (correctly assigning classification probabilities to the objects) is the 'goal' of the classification challenge for the test data. Note that there is one class in the test set that does not occur in the training set: `class_99` serves as an "other" class for objects that don't belong in any of the 14 classes in the training set. `Int8`

Time Series

- ▶ Data that is recorded at regular intervals of time, is called a time series.
- ▶ "Time" is the component which makes time-series problems more difficult to work with in a machine learning context.
- ▶ "Time" does actually play a part in many machine learning problems to the extent that a prediction (in the future) is being made of an observation that is currently being made. The prediction is made by the model where all previous observations used in training are treated equally.

Decomposition of a time series into 4 constituent parts

- ▶ **Level.** The baseline value for the series if it were a straight line.
- ▶ **Trend.** The optional and often linear increasing or decreasing behavior of the series over time.
- ▶ **Seasonality.** The optional repeating patterns or cycles of behavior over time.
- ▶ **Noise.** The optional variability in the observations that cannot be explained by the model.
- ▶ All time series have a level, most have noise, and the trend and seasonality are optional.

- 
- ▶ These constituent components can be thought to combine in some way to provide the observed time series. For example, they may be added together to form a model as follows:
 - ▶ $y = \text{level} + \text{trend} + \text{seasonality} + \text{noise}$
 - ▶ These components may also be the most effective way to make predictions about future values, but not always.
 - ▶ In cases where these classical methods do not result in effective performance, these components may still be useful concepts, and even input to alternate methods

Differences in Working with Time Series Data

- ▶ If you have experience working in machine learning, you must make some adjustments when working with time series. Below are seven key differences to keep in mind when making the transition.

<https://blogs.oracle.com/datascience/7-ways-time-series-forecasting-differs-from-machine-learning>

Handle Features with Care

- ▶ In time series forecasting, you don't create features — at least not in the traditional sense. This is especially true when you want to forecast several steps ahead, and not just the following value.
- ▶ Use care

'Creating' Features

- ▶ 1. It is not clear what the future real values will be for those features.
- ▶ 2. If the features are predictable, i.e., they have some patterns, you can build a forecast model for each of them. However, keep in mind that using predicted values as features will propagate the error to the target variable, which may cause higher errors or produce biased forecasts.
- ▶ 3. A pure time series model may have similar or even better performance than one using features.

TSDB

(Time Series Database)

- ▶ If you have a very large time series dataset a TSDB may be appropriate.
- ▶ Some datasets come from events recorded with a timestamp, systems logs, financial data, data obtained from sensors (IoT), etc.
- ▶ Since TSDB works natively with time series, it is a great opportunity to apply time series technique to large-scale datasets.

Smaller May Be Better

- ▶ There are some benefits to having small- to medium-sized time series:
 - ▶ The datasets will fit the memory of your computer.
 - ▶ In some cases, you can analyze the entire dataset, and not just a sample.
 - ▶ The length of the time series is convenient for making plots that can be graphically analyzed. This is a very important point, because we rely heavily on plot analyses in the time-series analysis step.

Algorithmic Approach

- ▶ Many machine learning algorithms do not have the capability the ability to extrapolate patterns outside of the domain of training data, as they tend to be restricted to a domain that is defined by training data.
- ▶ An important property of a time series algorithm is the ability to derive confidence intervals. While this is a default property of time series models, most machine learning models do not have this ability because they are not all based on statistical distributions. Confidence intervals can be estimated, but they may not be as accurate.

Model Options

- ▶ There are many complex models or approaches that may be very useful in some cases. Generalized Autoregressive Conditional Heteroskedasticity (GARCH), Bayesian-based models, and VAR.
- ▶ Neural network models that can be applied to time series which use lagged predictors and can handle features, such as Neural Networks Autoregression (NNAR).
- ▶ Time-series models borrowed from deep learning, specifically in the RNN (Recurrent Neural Network) family, like LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) networks.

Evaluation Metrics

- ▶ Common evaluation metrics for forecasting are:
 - ▶ RMSE, which you may have used on regression problems
 - ▶ MAPE, as it is scale-independent and represents the ratio of error to actual values as a percent
 - ▶ MASE, which indicates how well the forecast performs compared to a naïve average forecast
- ▶ Once a forecasting model has been fit, it is important to assess how well it is able to capture patterns.
- ▶ NOTE: While evaluation metrics help determine how close the fitted values are to the actual ones, they **do not evaluate whether the model properly fits the time series.**

Residuals

- ▶ Residuals are a good evaluation metric.
- ▶ In trying to capture the patterns of a time series, you would expect the errors to behave as white noise, as they represent what cannot be captured by the model. White noise must have the following properties:
 - ▶ The residuals are uncorrelated ($Acf = 0$)
 - ▶ The residuals follow a normal distribution, with zero mean (unbiased) and constant variance
- ▶ If either of the two properties are not present, it means that there is room for improvement in the model.

Resolution

- ▶ Aim for the most granular level possible. (monthly, quarterly, etc.)
- ▶ When using aggregates, the model is learning patterns at a macro level. This not a bad choice, but there may be some patterns at the granular level that the model is not paying attention to.

Confidence

- ▶ Forecasts are predictions that always include confidence intervals, usually 80% and 95%. Alternatively, you could choose to use the standard deviation of the residuals as the sample standard deviation, allowing the confidence intervals to be calculated using an appropriate distribution, like the normal or exponential.
- ▶ For some models, e.g., neural networks, which are not based on probability distributions, you can run simulations of the forecasts and calculate confidence intervals from the distribution of the simulations.

High Accuracy vs High Error

- ▶ You are assuming that past patterns are indicators of what may occur in the future, and therefore they get replicated or projected.
- ▶ However, if patterns change, either gradually or abruptly, the forecasts may deviate highly from actual results. There is a chance that “black swan” or “gray swan” events may occur. According to Investopedia:
 - ▶ **Black swan:** An event or occurrence that deviates beyond what is normally expected of a situation and is extremely difficult to predict.
 - ▶ **Gray swan:** An event that can be anticipated to a certain degree, but is considered unlikely to occur and may have a sizable impact if it does occur.
- ▶ This frequently occurs in economic time series. When this occurs, it is preferable to first evaluate the impact, and then, if required, update the forecasts using recent data after the event has passed.

