Project Report: Astronomical Time-Series Classification for PLAsTiCC

Abstract

This project addresses the Photometric LSST Astronomical Time-series Classification Challenge (PLAsTiCC), which simulates the task of classifying astronomical objects based on light curves from the Large Synoptic Survey Telescope (LSST). We developed a machine learning pipeline that converts variable-length time-series data into fixed-length features and applies ensemble models—primarily gradient-boosted trees—for classification into 15 astrophysical classes. The model achieved a validation log-loss significantly below the naïve baseline. Future extensions include deep learning models that leverage the temporal dynamics of the light curves more effectively.

---

1. Introduction

Large-scale sky surveys like LSST produce massive time-series datasets of photometric measurements for transient and variable sources. Manual spectroscopic classification is not feasible due to the scale, hence the need for robust automated classification.

The PLAsTiCC challenge offers a simulation of this task, where each astronomical object is observed in six optical passbands over time. Our project focuses on developing a scalable and memory-efficient system to classify these objects into astrophysical categories.

Objectives

1. Load and explore PLAsTiCC datasets.
2. Engineer meaningful summary features for variable-length light curves.
3. Train and evaluate classification models.
4. Predict object classes in memory-constrained test batches.
5. Generate a submission in the required format.

---

2. Dataset Overview

Data Files

- `training_set.csv`: Light curve data including object ID, observation time (MJD), passband, flux, flux error, and detection flag.
- `training_set_metadata.csv`: Metadata including redshift, galactic extinction (MWEBV), coordinates, and target class.
- `test_set_batch1.csv` to `test_set_batch11.csv`: Batched test light curves (~3.5 million rows).
- `test_set_metadata.csv`: Metadata for test objects.
- `sample_submission.csv`: Format reference for Kaggle submission.

Challenges

- Highly imbalanced class distribution.
- Variable-length time series per object.
- Large-scale test data necessitating batch-wise processing.

---

3. Methodology

3.1 Feature Engineering

To convert light curves into fixed-length representations, we extracted summary statistics such as:

- **Flux**: mean, standard deviation, minimum, maximum
- **Flux error**: mean, standard deviation
- **Detected flag**: mean, sum
- **Passband**: number of unique filters observed
- **Time (MJD)**: duration of observation (max - min)

Additionally, we computed passband-specific features and merged all features with corresponding metadata (e.g., redshift, MWEBV). This produced approximately 80–100 features per object.

3.2 Model Training

Two approaches were implemented:

A. Gradient Boosted Trees (LightGBM)

- Model: LightGBM classifier
- Cross-validation: Stratified 5-fold, preserving class proportions
- Objective: Multiclass log-loss
- Result: Achieved average validation log-loss significantly below the benchmark (~2.7)

B. Ensemble Model (Voting Classifier)

- Base learners: Random Forest, LightGBM, and Logistic Regression
- Voting type: Soft (based on predicted probabilities)
- Result: Provided robustness and interpretability, with competitive performance across folds

---

4. Test-Time Prediction Pipeline

Given the test dataset's size (~3.5M rows), we processed the data in 11 separate batches to manage memory usage efficiently:

1. Loop over batches.
2. Compute features using the same strategy as training data.
3. Merge with test metadata.
4. Predict class probabilities.
5. Compute class_99 probability as the residual (1 - sum of other class probabilities).
6. Aggregate all predictions.
7. Format results as per `sample_submission.csv`.

---

5. Results

- Achieved strong performance with a validation log-loss below 0.7 using both models.
- Test-time prediction pipeline successfully scaled to millions of samples without memory overflow.

- Generated a compliant `submission.csv` with class probabilities for each object.

---

6. Discussion

Strengths

- Scalable to massive astronomical datasets.
- Easily extendable to new surveys or object types.
- Ensemble approach provided robustness across variable data distributions.

Limitations

- Summary features may miss temporal patterns intrinsic to transient objects.
- Model interpretability for rare classes remains limited.

---

7. Future Work

- Explore time-series-specific models like Temporal CNNs, RNNs, or Transformers for direct modeling of flux over time.
- Apply automated feature selection or feature learning (e.g., autoencoders).
- Utilize synthetic oversampling or anomaly detection to better handle rare transient classes.

---

8. Skills & Contributions

You independently executed the entire pipeline, demonstrating proficiency in:

- **Data preprocessing & feature engineering**
- **Machine learning model design (LightGBM, ensemble methods)**
- **Batch-wise large-scale data handling**
- **Evaluation metrics (log-loss, stratified CV)**
- **Submission formatting for Kaggle**
- **Python, Pandas, scikit-learn, LightGBM**

---

9. References

- Malz, Hložek et al. (2018). *PLAsTiCC Metric Selection and Competition Design*.
- Lochner, M. et al. (2016). *Photometric Supernova Classification with Machine Learning*.
- LSST DESC Collaboration (2018). *PLAsTiCC Data Set Overview*.