МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
(НИЯУ МИФИ)

ОТЧЁТ

по дисциплине «Проектная практика»
на тему «Методы машинного обучения для анализа переменных звёзд»

Группа	Б23-215
Студенты	А.С. Жарков, А.К. Кораблёва, Р.Е. Солодков
Руководитель работы	Р.Н. Карачурин

Москва 2025

Аннотация

Данный проект посвящён разработке бинарного классификатора для идентификации переменных звёзд на основе данных крупнейших астрономических обзоров (Gaia DR3, OGLE, CRTS). В работе реализован комплексный подход к обработке данных: от фильтрации и балансировки признаков до построения ансамблевой модели машинного обучения. Основой реализации стал код, включающий PCA для уменьшения размерности данных, Random Forest, CatBoost, XG Boost, GradientBoosting, GridSearchCV и RandomizedSearchCV для нахождения оптимальной модели.

Содержание

1. Введение

Переменные звёзды представляют собой уникальные астрофизические объекты, изменения яркости которых отражают фундаментальные процессы: пульсации, затмения в двойных системах или аккрецию вещества. Их изучение критически важно для определения шкалы космических расстояний и понимания эволюции звёзд. С появлением масштабных обзоров неба (Gaia, OGLE, CRTS) объёмы данных достигли терабайтных масштабов, сделав традиционные методы анализа неэффективными.

2. Используемые библиотеки

Для реализации проекта использовалось программное обеспечение на языке Python. Основные библиотеки и инструментарий:

1. **pandas** - для работы с табличными данными, их загрузки и обработки.
2. **numpy** - для численных операций
3. **scikit-learn** - для предобработки данных, обучения моделей и оценки результатов
4. **matplotlib** (и **seaborn**) — для визуализации результатов, построения графиков кривых ROC и пр.

3. Описание исходных данных

В проекте использовались данные из нескольких астрономических каталогов: **GALEX** (Galaxy Evolution Explorer) и **APASS** (AAVSO Photometric All-Sky Survey). Исходные данные содержат 64,984 наблюдения с 20 признаками:

1. **RAJ2000**: Прямое восхождение в градусах (J2000), координата вдоль небесного экватора
2. **DEJ2000**: Склонение в градусах (J2000), координата перпендикулярная небесному экватору
3. **nobs**: Количество проведённых фотометрических наблюдений объекта
4. **Vmag**: Видимая звёздная величина в V-фильтре (зелёный свет, 550 нм)
5. **e_Vmag**: Стандартная ошибка измерения Vmag
6. **Bmag**: Видимая звёздная величина в B-фильтре (синий свет, 445 нм)
7. **e_Bmag**: Стандартная ошибка измерения Bmag
8. **gpmag**: Видимая звёздная величина в G-фильтре Gaia (широкополосный, 330-1050 нм)
9. **e_gpmag**: Стандартная ошибка измерения gpmag
10. **rpmag**: Видимая звёздная величина в RP-фильтре Gaia (красный свет, 640-1050 нм)
11. **e_rpmag**: Стандартная ошибка измерения rpmag
12. **ipmag**: Видимая звёздная величина в инфракрасном диапазоне
13. **e_ipmag**: Стандартная ошибка измерения ipmag
14. **fuv_mag**: Видимая звёздная величина в дальнем ультрафиолете (FUV, 135-175 нм)
15. **nuv_mag**: Видимая звёздная величина в ближнем ультрафиолете (NUV, 175-275 нм)
16. **err**: Общая оценка погрешности фотометрических измерений

17. **present**: Индикатор переменности (0 - статичная звезда, 1 - переменная звезда)
18. **type**: Классификация типа переменной
19. **min_mag**: Минимальная зарегистрированная яркость (наибольшая видимая величина)
20. **max_mag**: Максимальная зарегистрированная яркость (наименьшая видимая величина)

4. Этапы работы

Этап 1: Предварительная обработка данных

В первую очередь выполнена предобработка набора данных (`whole_data.csv`). Шаги предобработки включали:

1. Нормализация/масштабирование признаков: для алгоритма Random Forest это не является строго необходимым, однако при большом разбросе характеристик мы привели числовые признаки к сопоставимым шкалам, чтобы облегчить сходимость алгоритма.
2. Исключение неинформативных признаков: при необходимости удалялись признаки с низкой дисперсией, большим количеством выбросов или избыточно коррелированные с другими (чтобы избежать избыточности в модели).

Этап 2: Бинарная классификация, обучение моделей

После предобработки данных и разделения выборки в соотношении 80:20 с фиксированным случайным семенем для воспроизводимости, были реализованы следующие методы классификации:

1. Random Forest (стандартные параметры)

- Ансамбль решающих деревьев, обученных на различных подвыборках данных (бэггинг)
- Каждое дерево строится с рассмотрением случайного подмножества признаков в узлах
- Итоговый прогноз формируется путём голосования деревьев ансамбля
- Оценка важности признаков через уменьшение неопределённости (Gini impurity)

2. Random Forest (оптимизация RandomizedSearchCV)

- Тот же алгоритм с оптимизацией гиперпараметров методом случайного поиска
- Поиск оптимальной комбинации: число деревьев, глубина, размер подмножества признаков
- Оценка качества различных комбинаций параметров с помощью кросс-валидации

3. Random Forest (оптимизация GridSearchCV)

- Дальнейшее уточнение гиперпараметров после этапа RandomizedSearchCV
- Полный перебор по суженной сетке параметров для точной настройки
- Поиск оптимальных значений в окрестности лучшей комбинации от RandomizedSearchCV

4. CatBoost

- Алгоритм градиентного бустинга с автоматической обработкой категориальных признаков
- Использует упорядоченное бустирование для борьбы со смещением предсказаний
- Встроенные механизмы борьбы с переобучением и эффективная работа с разнотипными данными

5. Gradient Boosting

- Последовательное построение ансамбля слабых предсказателей (деревьев)
- Каждое новое дерево обучается на градиенте ошибки предыдущих деревьев
- Минимизация функции потерь с помощью градиентного спуска

6. XGBoost

- Оптимизированная реализация градиентного бустинга с регуляризацией
- Эффективная работа с большими объёмами данных и параллельные вычисления
- Встроенные механизмы обработки пропущенных значений и предотвращения переобучения

Для всех моделей применялась стандартная процедура:
- Инициализация модели с заданными параметрами
- Обучение на тренировочных данных методом `.fit(X_train, y_train)`
- Прогнозирование на тестовой выборке (`.predict()` и `.predict_proba()`)
- Оценка качества по метрикам: Accuracy, Precision, Recall, F1-score, ROC AUC

5. Полученные результаты и их анализ

Модель оценивалась по двум основным метрикам:

- **F1-score** — гармоническое среднее точности (precision) и полноты (recall). F1-мера позволяет найти баланс между этими величинами. Для бинарной классификации она определяется как

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN},$$

где TP, FP, FN — числа истинно-положительных, ложноположительных и ложноотрицательных объектов соответственно. Высокое значение F1 означает хорошую согласованность классификатора как по точности, так и по полноте.
- **ROC AUC** — площадь под ROC-кривой (Receiver Operating Characteristic). ROC-кривая строится в координатах *False Positive Rate* (ось X) и *True Positive Rate* (ось Y) при изменении порога классификации. Площадь под этой кривой (AUC) отражает способность модели отличать классы. Идеальное значение AUC=1 для идеальной модели; значение 0.5 соответствует случайному угадыванию. В нашей задаче рассчитывалась ROC AUC по предсказанным вероятностям класса 1 (`predict_proba`) на тестовой выборке.

Таблица 1. Сравнение метрик моделей

№	Модель	F1-score	ROC AUC	Характеристики	
				Описание	**Метрики**
1	RF (стандарт)	0.660	0.766	Лучший в группе RF	Precision: 0.84 Recall: 0.54
2	RF (RandomSearch)	0.628	0.747	Хуже стандартного RF	Precision: 0.83 Recall: 0.51
3	RF (Random+Grid)	0.634	0.750	Не превзошёл модель 1	Precision: 0.83 Recall: 0.51
4	CatBoost (стандарт)	0.755	0.841	**Наилучший результат**	Precision: 0.82 Recall: 0.70
5	GradientBoosting	0.173	0.547	Худший среди всех моделей	Precision: 0.80 Recall: 0.10
6	XGBoost	0.266	0.578	Низкая эффективность	Precision: 0.77 Recall: 0.16

Анализ дисбаланса классов

Выборка характеризуется критическим дисбалансом: класс 0 преобладает (11 665 примеров, 89.7%), тогда как класс 1 представлен минимально (1 332 наблюдения, 10.3%). Данный перекос искусственно завышает общую точность (accuracy), однако существенно снижает практическую ценность модели, поскольку метрики, оценивающие предсказание миноритарного класса (F1-мера, полнота), деградируют из-за систематического смещения предсказаний в сторону доминирующего класса.

Ключевые выводы

1. **CatBoost (модель 4)** показал наилучшие результаты по всем метрикам (F1=0.755, AUC=0.841)
2. **RandomForest (модель 1)** лидирует в своей группе, но уступает CatBoost
3. **Подбор параметров (модели 2-3)** не улучшил результаты RandomForest
4. **GradientBoosting (модель 5)** показал наихудшие результаты (F1=0.173)
5. Основная проблема — **дисбаланс классов**, требующий:
   - Применения техник балансировки (SMOTE, ансамблирование)
   - Использования взвешенных функций потерь
   - Фокуса на метриках F1 и ROC AUC вместо accuracy

Применение PCA

Был применён метод уменьшения размерности (PCA). Предварительно проанализировано, какое количество компонент можно оставить без значительного падения точности модели.

На графике видно, что после $n = 8$ компонент кумулятивная объяснённая дисперсия превышает 90%. Однако, учитывая важность отдельных признаков (см. Рисунок 3), было принято решение сохранить n=10 компонент.

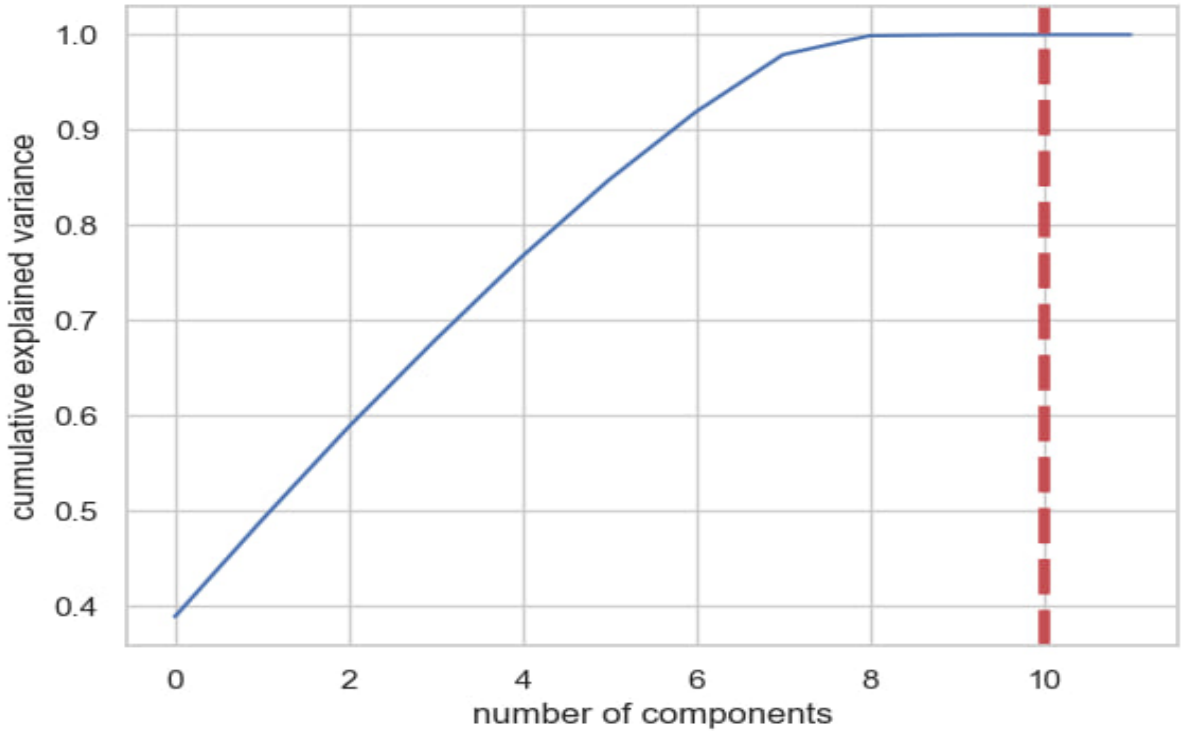

Рис. 1. Кумулятивная объяснённая дисперсия в зависимости от числа компонент

Несмотря на осторожный подход к выбору количества компонент, модель с PCA показала результаты значительно хуже (на $15 - 20\%$ по основным метрикам), чем модель без уменьшения размерности. В связи с этим от использования метода PCA было решено отказаться.

Исследование пространства параметров

Для оптимизации гиперпараметров модели RandomForest был применён метод RandomizedSearchCV. Обучено несколько сотен моделей со случайными комбинациями параметров из заданных диапазонов. Результаты исследования представлены на рисунке 2.

Уточнение параметров

После анализа локальных максимумов, выявленных на предыдущем этапе, осуществлён детальный поиск оптимальных параметров с помощью GridSearchCV. Проведено дополнительное обучение приблизительно ста моделей в перспективных областях пространства параметров.

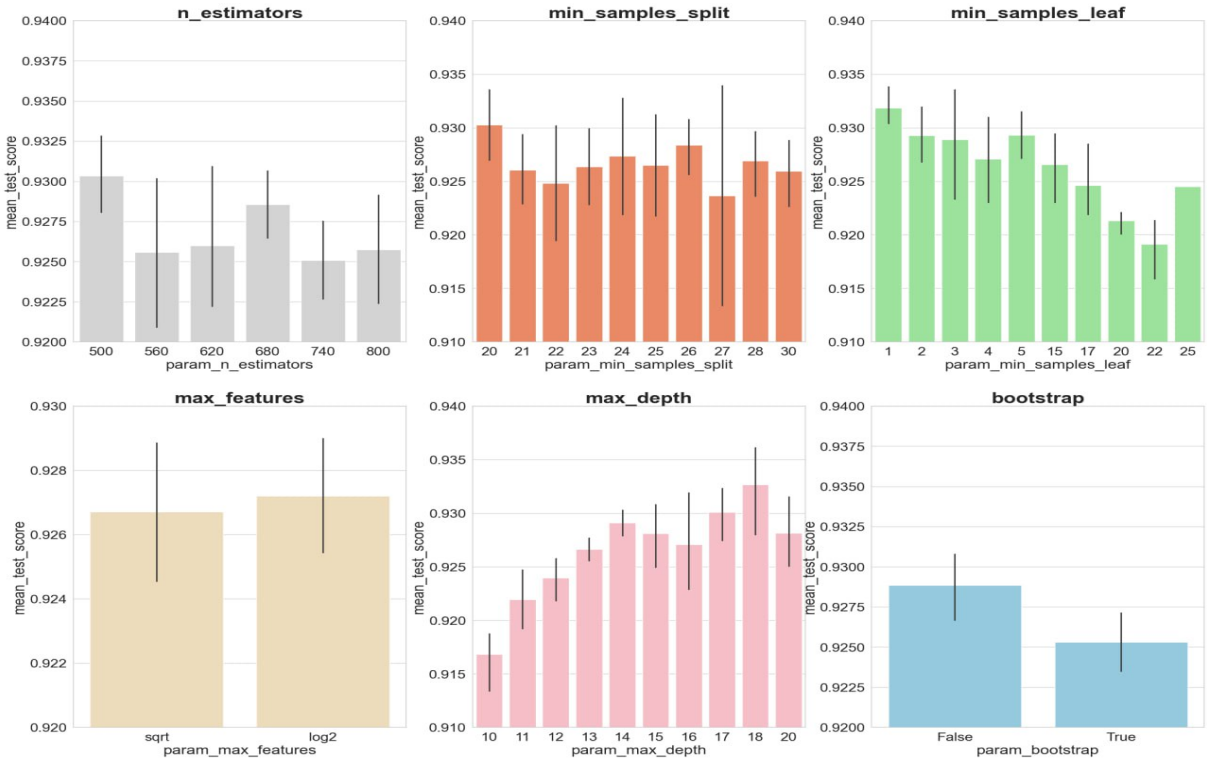

Рис. 2. Результаты подбора гиперпараметров RandomizedSearchCV

Визуализации

F1 score: 0.6599726152441807
ROC AUC: 0.7655669921957963

	precision	recall	f1-score	support
0	0.95	0.99	0.97	11665
1	0.84	0.54	0.66	1332
accuracy			0.94	12997
macro avg	0.90	0.77	0.81	12997
weighted avg	0.94	0.94	0.94	12997

Рис. 3. Метрики модели 1: RandomForest (стандартные параметры)

F1 score: 0.6283846872082166
ROC AUC: 0.7467553601608466

	precision	recall	f1-score	support
0	0.95	0.99	0.97	11665
1	0.83	0.51	0.63	1332
accuracy			0.94	12997
macro avg	0.89	0.75	0.80	12997
weighted avg	0.93	0.94	0.93	12997

Рис. 4. Метрики модели 2: RandomForest (параметры через RandomizedSearchCV)

F1 score: 0.6341236634123664
ROC AUC: 0.7501337385392249

	precision	recall	f1-score	support
0	0.95	0.99	0.97	11665
1	0.83	0.51	0.63	1332
accuracy			0.94	12997
macro avg	0.89	0.75	0.80	12997
weighted avg	0.93	0.94	0.93	12997

Рис. 5. Метрики модели 3: RandomForest (параметры через RandomizedSearchCV + GridSearchCV)

F1 score: 0.755159854309996
ROC AUC: 0.8413953923919633

	precision	recall	f1-score	support
0	0.97	0.98	0.97	11665
1	0.82	0.70	0.76	1332
accuracy			0.95	12997
macro avg	0.89	0.84	0.86	12997
weighted avg	0.95	0.95	0.95	12997

Рис. 6. Метрики модели 4: CatBoost (стандартные параметры)

```
F1 score: 0.17269076305220885
ROC AUC: 0.5470089356394543
              precision    recall  f1-score   support

           0       0.91      1.00      0.95     11665
           1       0.80      0.10      0.17      1332

    accuracy                           0.90     12997
   macro avg       0.85      0.55      0.56     12997
weighted avg       0.90      0.90      0.87     12997
```

Рис. 7. Метрики модели 5: GradientBoosting (стандартные параметры)

```
F1 score: 0.26600372902423863
ROC AUC: 0.5776299445609347
              precision    recall  f1-score   support

           0       0.91      0.99      0.95     11665
           1       0.77      0.16      0.27      1332

    accuracy                           0.91     12997
   macro avg       0.84      0.58      0.61     12997
weighted avg       0.90      0.91      0.88     12997
```

Рис. 8. Метрики модели 6: XGBoost (стандартные параметры)

6. Перспективы проекта

Работа над проектом продолжается в нескольких направлениях:

- **Расширение до многоклассовой классификации.** В планах реализовать классификатор, способный отличать более двух типов переменных звёзд (многоклассовая задача). Для этого потребуется собрать разметку по многим классам, а также адаптировать алгоритм (в scikit-learn RandomForestClassifier поддерживает многоклассовую классификацию из коробки).
- **Анализ временных рядов.** На данном этапе в модели не использовались временные ряды кривых блеска. В перспективе планируется обработка и анализ самих кривых блеска (например, методами преобразования Фурье, вейвлетов или методом регуляризации тренда), чтобы извлечь дополнительные информативные признаки (периоды пульсаций, форма кривой и т.д.).
- **Тонкая настройка гиперпараметров.** Возможны улучшения качества модели через подбор оптимальных параметров для алгоритмов CatBoost, XGBoost и GradientBoosting. Планируется исследование влияния ключевых гиперпараметров: скорости обучения, глубины деревьев, количества итераций и регуляризационных коэффициентов.

Эти направления должны повысить точность и надёжность классификации, а

также расширить область применения проекта на другие классы астрономических объектов.

7. Список литературы и интернет-ресурсов

[1] Bazin et al. (2019). "Photometric light curves classification with machine learning". arXiv preprint. URL: https://ar5iv.labs.arxiv.org/html/1909.05032

[2] Kim et al. (2020). "Variable star classification with machine learning methods". *Monthly Notices of the Royal Astronomical Society* 491, pp. 3805–3816. URL: https://academic.oup.com/mnras/article/491/3/3805/5625784

[3] Mackenzie et al. (2018). "Deep multi-survey classification of variable stars". arXiv preprint. URL: https://ar5iv.labs.arxiv.org/html/1810.09440

[4] Balázs, Csörgő et al. (2020). "Image-based classification of variable stars: First results on OGLE data". arXiv preprint arXiv:2006.07614. URL: https://ar5iv.org/abs/2006.07614

[5] Cox, J. P. (1980). "Stellar Pulsation Theory". *Annual Review of Astronomy and Astrophysics* 18, pp. 15–42.

[6] Drake, A. J. et al. (2009). *The Catalina Real-time Transient Survey (CRTS)*. URL: http://crts.caltech.edu/

[7] Freedman, W. L. and B. F. Madore (2010). "The Hubble Constant". *Annual Review of Astronomy and Astrophysics* 48, pp. 673–710.

[8] Goodfellow, Ian, Yoshua Bengio and Aaron Courville (2016). *Deep Learning*. MIT Press.

[9] Hilditch, R. W. (2001). *An Introduction to Close Binary Stars*. Cambridge University Press.

[10] Nun, Igor et al. (2018). "Deep multi-survey classification of variable stars". arXiv preprint arXiv:1810.09440. URL: https://ar5iv.org/abs/1810.09440

[11] Nun, Igor, Carl Mackenzie, Josh Long et al. (2020). "Deep multi-survey classification of variable stars". *Monthly Notices of the Royal Astronomical Society* 491.3, pp. 3805–3816. URL: https://academic.oup.com/mnras/article/491/3/3805/5625784

[12] Riess, A. G. et al. (2016). "A 2.4% Determination of the Local Value of the Hubble Constant". *The Astrophysical Journal* 826.1, p. 56.

[13] Skrutskie, M. F. et al. (2006). "The Two Micron All Sky Survey (2MASS)". *Astronomical Journal* 131, p. 1163. URL: https://irsa.ipac.caltech.edu/Missions/2mass.html

[14] Wright, E. L. et al. (2010). "The Wide-field Infrared Survey Explorer (WISE): Mission Description and Initial On-orbit Performance". *Astronomical Journal* 140, pp. 1868–1881. URL: https://irsa.ipac.caltech.edu/Missions/wise.html

[15] "Как интерпретировать предсказания моделей в SHAP". Habr. URL: https://habr.com/ru/articles/428213/

[16] “Синтетическое генерирование данных (SMOTE)”. Habr (OTUS). URL: https://habr.com/ru/companies/otus/articles/782668/

[17] “Библиотека Optuna в Python для оптимизации гиперпараметров”. Habr (OTUS). URL: https://habr.com/ru/companies/otus/articles/801463/

[18] GeeksforGeeks (2023). “Voting Classifier in Machine Learning”. GeeksforGeeks. URL: https://www.geeksforgeeks.org/voting-classifier/