

Датасет: Факторы успеваемости студентов

ВЫПОЛНИЛА: ГЛЕБОВА МАРИЯ, ГРУППА М8О-307Б-23



Цель: Исследовать методы подбора гиперпараметров для модели Random Forest и сравнить их эффективность

Задачи:

1. Выбрать модель и изучить её гиперпараметры
2. Подготовить датасет для обучения
3. Подобрать гиперпараметры тремя методами (Grid Search, Random Search, Optuna)
4. Сравнить результаты методов
5. Создать калькулятор с интерпретацией модели (LIME и SHAP)

Выбор модели - Random Forest

Причины выбора:

- Хорошо работает для классификации и регрессии
- Устойчив к переобучению
- Обрабатывает разные типы признаков
- Много гиперпараметров для настройки
- Позволяет оценивать важность признаков

```
# Создадим набор из гиперпараметров, которых много!
# Используем ClassifyRandomForest
{
    "n_estimators": 100,
    "max_depth": null,
    "min_samples_split": 2,
    "min_samples_leaf": 1,
    "max_features": "sqrt",
    "criterion": "gini",
    "bootstrap": true,
    "max_samples": null,
    "class_weight": null,
    "random_state": null,
    "n_jobs": -1
}

# Описание:
# Количество деревьев в лесу
# Максимальная глубина дерева (更深 - без ограничений)
# Минимальное количество образцов для разделения узла
# Минимальное количество образцов в листе
# Количество признаков для рассмотрения при разделении (sqrt, log2, None или число)
# Использование для разделения качества разделения (gini, entropy, log loss)
# Использовать ли bootstrap выборку при построении деревьев
# Количество образцов для обучения каждого дерева (если bootstrap = true)
# Критерий для вычисления качества разделения (gini)
# Случайные признаки для каждого дерева (если max_features < n)
# Количество параллельных задач (-1 - все доступные ядра)
```

Таблица: гиперпараметры Random Forest:

Гиперпараметр	Описание	Тип	Значение по умолчанию
0 n_estimators	Количество деревьев в лесу	int	100
1 max_depth	Максимальная глубина дерева (更深 - без ограничений)	int или None	None
2 min_samples_split	Минимальное количество образцов для разделения	int или float	2
3 min_samples_leaf	Минимальное количество образцов в листе	int или float	1
4 max_features	Количество признаков для рассмотрения при разделении	str, int, float или None	sqrt
5 criterion	Функция или измерение качества разделения (gini, entropy, log loss)	str	gini
6 bootstrap	Использовать ли bootstrap выборку при построении деревьев	bool	True
7 max_samples	Количество образцов для обучения каждого дерева	int, float или None	None
8 class_weight	Веса классов (для несбалансированных данных)	dict, list или str	None
9 random_state	Случайный сокращник для построения деревьев	int	None
10 n_jobs	Количество параллельных задач (-1 - все доступные ядра)	int	None

Датасет и подготовка данных

- **Датасет:** Факторы успеваемости студентов

- **Размер:** 6607 записей, 20 признаков

- **Целевая переменная:** Exam_Score (балл на экзамене)

- **Классы:** 3 категории (Низкая, Средняя, Высокая успеваемость)

- **Предобработка:**

- Заполнение пропусков
- Кодирование категориальных признаков
- Разделение на train/test (80/20)

■ Информация о датасете:			
<class 'pandas.core.frame.DataFrame'>			
RangeIndex: 6607 entries, 0 to 6606			
data columns (total 20 columns):			
#	Column	Non-null Count	Type
0	Hours_Studied	6607 non-null	int64
1	Attendance	6607 non-null	int64
2	Parental_Involvement	6607 non-null	object
3	Access_to_Resources	6607 non null	object
4	Extracurricular_Activities	6607 non null	object
5	Sleep_Hours	6607 non null	int64
6	Previous_Scores	6607 non null	int64
7	Motivation_Level	6607 non null	object
8	Internet_Access	6607 non-null	object
9	Tutoring_Sessions	6607 non-null	int64
10	Family_Income	6607 non-null	object
11	Teacher_Quality	6607 non-null	object
12	School_Type	6607 non null	object
13	Peer_Influence	6607 non null	object
14	Physical_Activity	6607 non-null	int64
15	Learning_Disabilities	6607 non null	object
16	Parental_Education_Level	6607 non null	object
17	Distance_From_Home	6549 non-null	object
... имя файла: student-mat.csv			
None			

✓ Датасет загружен из: student-mat.csv

■ Всего записей: 6607 строк, 20 признаков

■ Первые строки:

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previous_Scores	Motivation
0	23	84	Low	High	No	7	43	Low
1	19	64	Low	Medium	No	8	59	Low
2	24	58	Medium	Medium	Yes	7	51	Medium
3	20	60	Low	Medium	Yes	8	50	Medium
4	19	59	Medium	Medium	Yes	6	45	Medium
5	19	88	Medium	Medium	Yes	8	80	Medium
6	29	84	Medium	Low	Yes	7	68	Medium
7	25	78	Low	High	Yes	6	50	Medium
8	17	54	Medium	High	No	6	40	Medium
9	23	58	Medium	Medium	Yes	8	41	Medium

Методы подбора гиперпараметров

****Grid Search:**** Полный перебор всех комбинаций

- Плюсы: гарантированно находит лучшую комбинацию
- Минусы: очень медленный

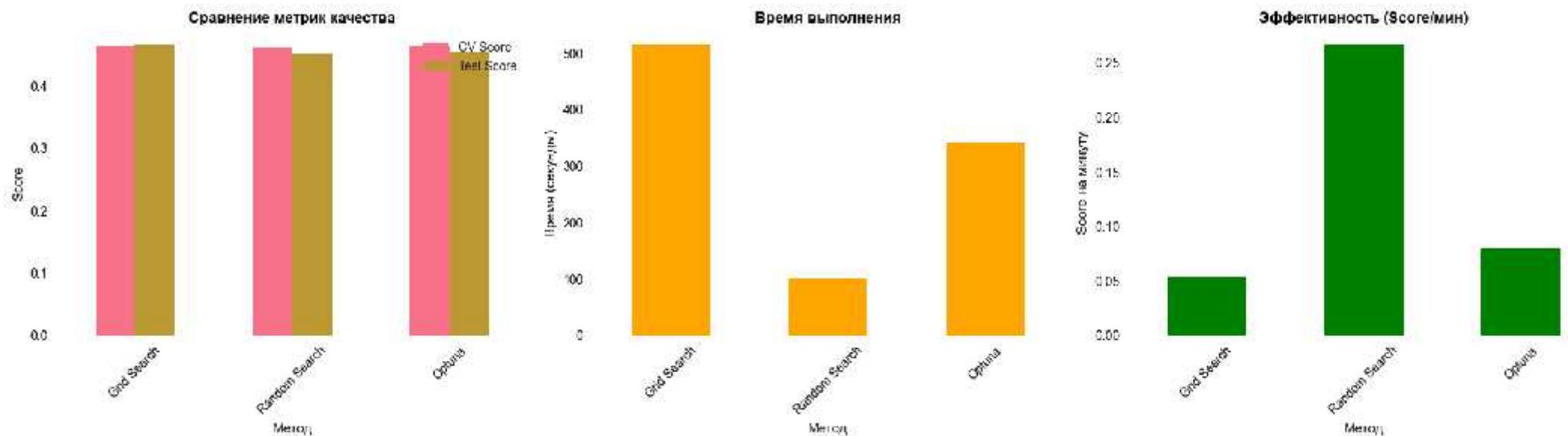
****Random Search:**** Случайный поиск из заданных диапазонов

- Плюсы: быстрее Grid Search
- Минусы: не гарантирует оптимальность

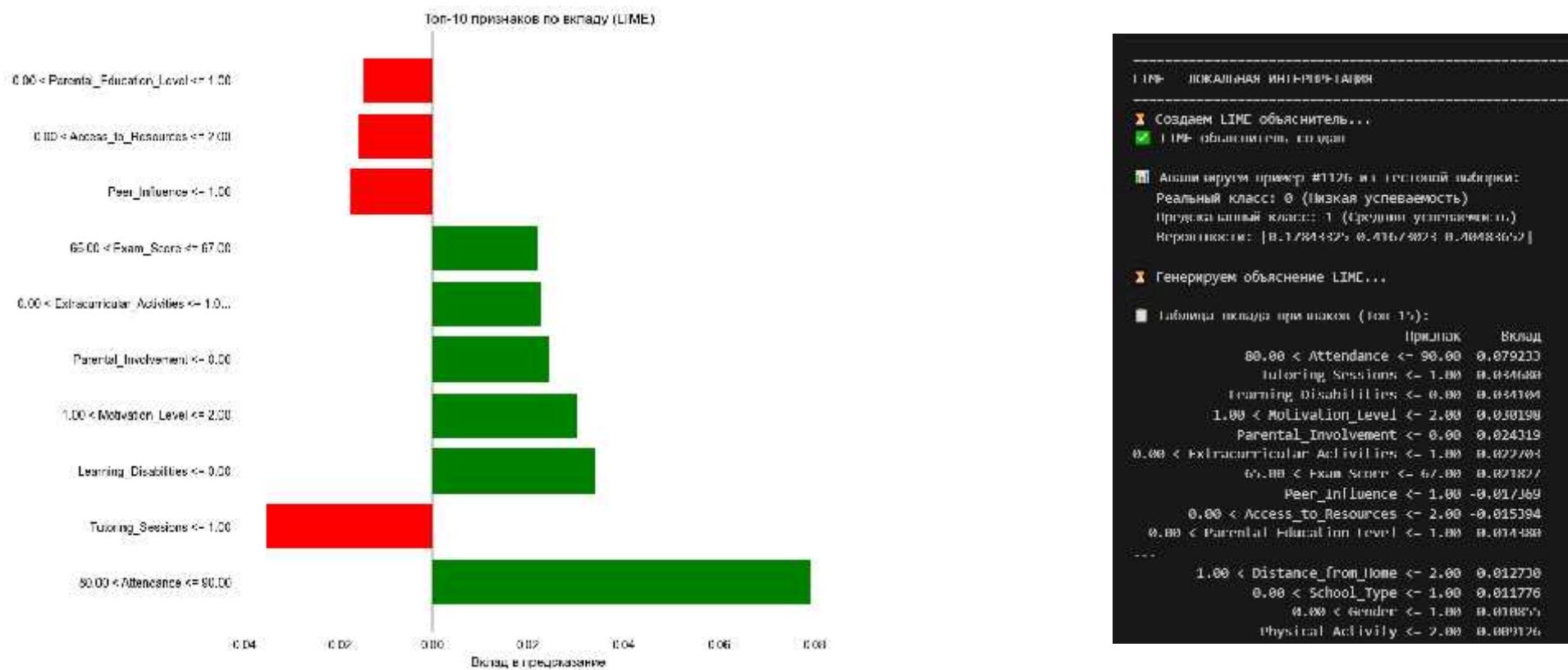
****Optuna:**** Байесовская оптимизация

- Плюсы: умный поиск, учится на результатах
- Минусы: сложнее в понимании

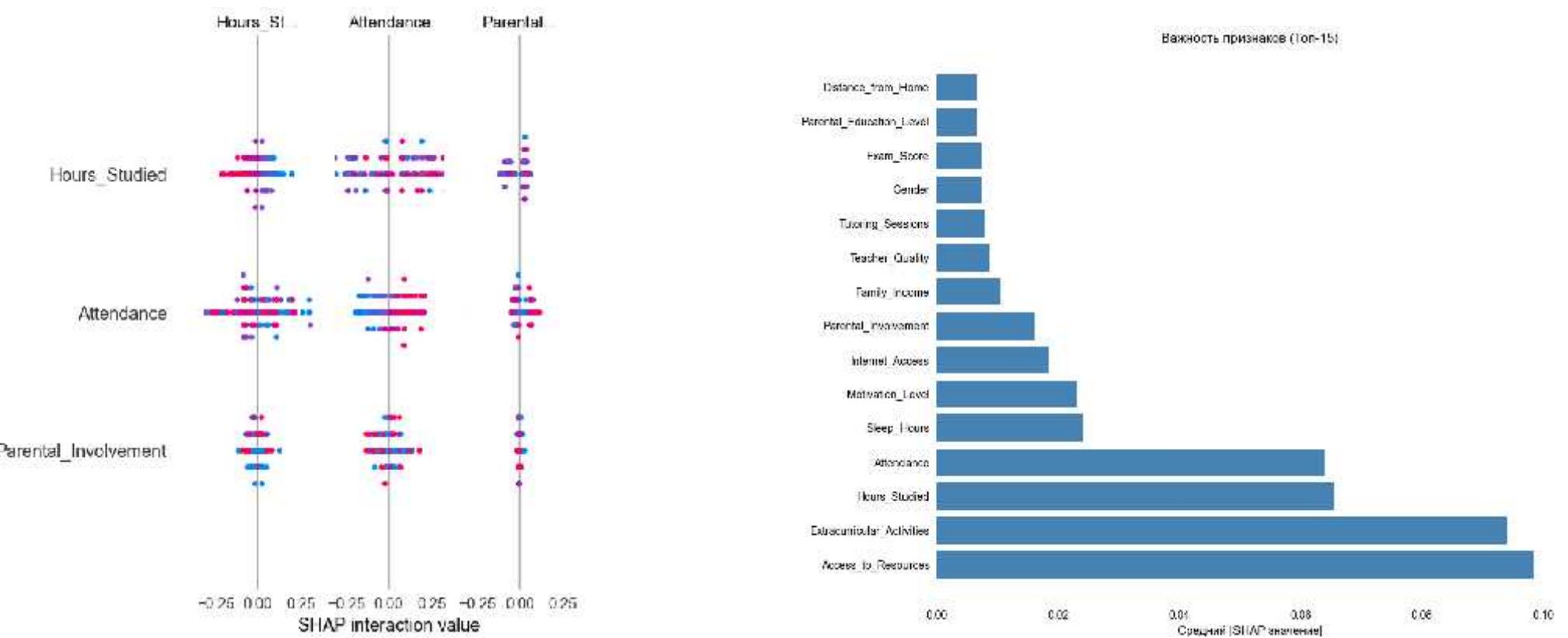
Сравнение результатов методов



Локальная интерпретация (LIME)



Глобальная интерпретация (SHAP)





Основные выводы:

1. Random Forest показал хорошие результаты для задачи классификации успеваемости
2. Optuna оказался наиболее эффективным методом подбора гиперпараметров
3. LIME и SHAP дополняют друг друга в интерпретации модели
4. Наиболее важные признаки: Previous_Scores, Attendance, Hours_Studied

Практическая значимость:

- Создан калькулятор для предсказания и интерпретации результатов
- Модель может использоваться для анализа факторов успеваемости студентов



Спасибо за внимание!