Подбор гиперпараметров модели

Солодова София Михайловна М8О-309Б-23
Cardiovascular Disease Dataset

Постановка задачи

ЗАДАЧА

Бинарная классификация – прогноз наличия сердечно-сосудистых заболеваний

ОБЪЕМ ДАННЫХ

70 000 наблюдений
12 признаков

ЦЕЛЬ РАБОТЫ

- Сравнить методы подбора гиперпараметров
- Проанализировать эффективность AutoML (TPOT)
- Исследовать интерпретируемость моделей

ПОДГОТОВКА ДАННЫХ

01 Удаление идентификатора (id)

02 Разделение на train/test(80/20)

03 Создание пайплайна с CatBoostClassifier

Проблема

- Несбалансированные классы
- Разнородные признаки

Особенности

- CatBoost сам обрабатывает категориальные признаки (one-hot не требуется)
- Pipeline для последующего использования в Grid/Random search

Grid Search

Принцип работы

Полный перебор всех комбинаций параметров
Точный, но вычислительно дорогой

Математическая основа

```
for params in parameter_grid:
    model = Model(**params)
    score = cross_val_score(model, X, y)
    update_best_if_needed(score, params)
```

Преимущества

Гарантированно находит оптимальную комбинацию в заданном пространстве
Прост в реализации

Недостатки

Экспоненциальный рост времени выполнения
Неэффективен для больших пространств параметров

Random Search

Принцип работы

Случайная выборка из пространства параметров
Вероятностный подход

Математическая основа

```
for i in range(n_iter):
    random_params = sample(parameter_distribution)
    score = evaluate_model(random_params)
```

Преимущества

Быстрее Grid Search
Эффективен для high-dimensional пространств
Хорош для первоначального исследования

Недостатки

Не гарантирует нахождение оптимума
Может пропускать важные области

Байесовская оптимизация (Optuna)

Принцип работы

Строит вероятностную модель целевой функции
Балансирует exploration/exploitation

Математическая основа

Использует surrogate model (часто Gaussian Process)
Acquisition function (EI, UCB) для выбора следующей точки

Преимущества

Умный поиск - учится на предыдущих итерациях
Эффективен для дорогих функций
Хорош для сложных пространств

Недостатки

Сложность реализации
Зависимость от выбора surrogate model

Genetic Programming (TPOT)

Принцип работы

Эволюционные алгоритмы для оптимизации пайплайнов
"Скрещивание" и "мутация" конвейеров

Основные операции

Selection - отбор лучших пайплайнов
Crossover - комбинация частей пайплайнов
Mutation - случайные изменения

Преимущества

Автоматический подбор не только параметров, но и алгоритмов
Находит неочевидные решения
Генерирует готовый Python код

Недостатки

Долгое время выполнения для сложных задач
Требует значительных ресурсов

SHAP (SHapley Additive exPlanations)

Принцип работы

Теория игр Шепли для распределения "вклада" между признаками
Справедливое распределение влияния на prediction

Математическая основа

$$SHAP\_value(i) = \Sigma\ [f(S \cup \{i\}) - f(S)] * |S|!(M-|S|-1)!/M!$$

Преимущества

Теоретически обоснованный метод
Локальная и глобальная интерпретируемость
Работает с любыми моделями

Типы объяснений

Force plot - вклад в конкретное предсказание
Summary plot - глобальная важность признаков
Dependence plot - зависимость предсказания от признака

Сравнение методов

Метод	Точность	Время	Лучшая модель	Ключевые параметры
GridSearch	0,7770	~20min	CatBoost	Depth=6 lr=0.05
RandomSearch	0,7772	~15min	CatBoost	Subsample=0.9
Optuna	0,7764	~10min	CatBoost	Iterations=660
Tpot	0,7752	~1h	MLPClassifier + preprocessing	Auto-pipline

Анализ TPOT

01 **MaxAbsScaler** - масштабирование

02 **RFE, ExtrsTreesClassifier** – отбор признаков

03 **FeatureUnion** – комбинация преобразований

04 **XGBClassifier** – финальная модель

Особенности

Автоматический подбор preprocessing steps
Комбинация нескольких алгоритмов
Готовый код для использования

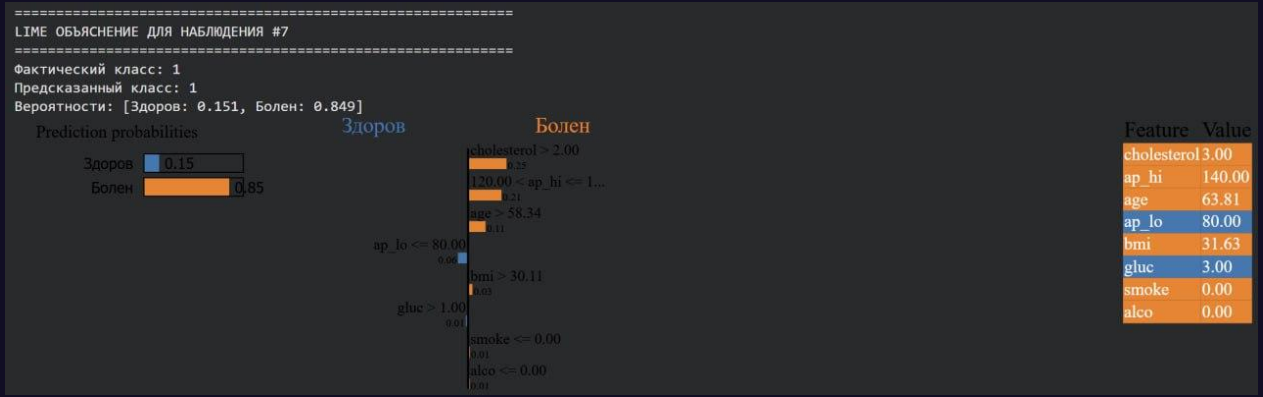
Интерпретация моделей (LIME)

Пример анализа

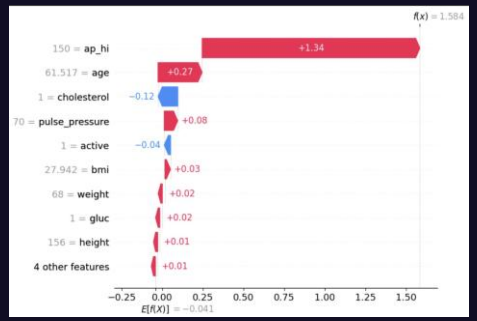
- Наблюдение #7: модель уверенно предсказывает "Cardio" (85%)
- Ключевые факторы: холестерин, артериальное давление, возраст пациента

Визуализация

- Веса признаков для конкретного предсказания
- Факторы "за" и "против" класса

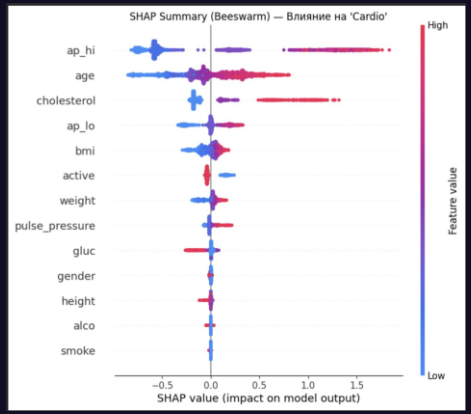


Практическое применение SHAP



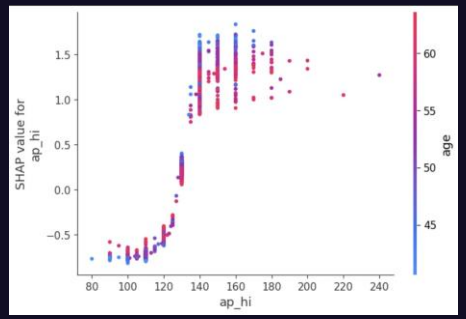
Локальные объяснения:

- Почему конкретный пациент отнесен к классу "Cardio"
- Вклад каждого признака в предсказание



Summary plot:

- Важность признаков
- Распределение влияния признаков на модель



Dependence plots:

- Как изменение признака влияет на предсказание
- Взаимодействия между признаками

Выводы

01 **GridSearch** – наиболее стабильный и интерпретируемый результат

02 **RandomSearch** – лучшая точность

03 **Optuna** – лучший баланс точности и времени

04 **TPOT** – в данной задаче показал худшие результаты

05 **Интерпретация** критически важна для медицинских приложений