

# MAC система для автоматизации ML задач - AetherML.

## Описание задачи

Необходимо создать прототип мультиагентной системы (сокращенно MAC) пред назначенной для построения пайплайнов машинного обучения с помощью AutoML фреймворков. Предложенная MAC должна не только использовать существующие AutoML-фреймворки, но и расширять их, находя и интегрируя новые, более эффективные модели из актуальных научных статей. Ключевая идея такого подхода — переход от использования готовых инструментов к их активному улучшению В качестве “альтернативно решения” можно рассматривать концепцию “мета-оптимизатор пайплайнов”, которая не использует один фреймворк, а динамически комбинирует и оркестрирует несколько фреймворков для создания гибридного, оптимального пайплайна. Например это может выглядеть следующим образом - «Использовать Fast-пресет AutoGluon для быстрой предобработки, затем применить ансамблирование от Auto-sklearn 2.0». Агенты координируют выполнение спланированного пайплайна, управляя вызовами разных фреймворков и передачей данных между ними, а затем финальный гибридный пайплайн сравнивается с результатами работы каждого фреймворка по отдельности, чтобы доказать эффективность подход:

1. В качестве списка automl фреймворков можно взять решения представленные в статье "AMLB: an AutoML Benchmark" (например AutoGluon, FEDOT и так далее)
2. Базовая реализация MAC - использование готовых AutoML библиотек с помощью промтов и function calling
3. Продвинутая реализация MAC - делается несколько уточняющих шагов (multi-hop logic), в числе которых веб-скрейпинг или извлечение новых алгоритмов или моделей из статей, дальнейшая кодовая реализация этих моделей и последующая интеграция их в выбранную AutoML библиотеку.
4. В качестве бенчмарка на которой будет происходить валидация качества предложенных MAC-систем необходимо использовать датасеты из AMLB: an AutoML Benchmark для задач классификации и регрессии.

## Описание этапности реализации (опционально, команда может выбрать свою стратегию)

Этап	Задачи	Результат этапа
1. Базовый прогон и анализ	<p>Запустить выбранные AutoML-фреймворки на отобранных датасетах.</p> <p>Зафиксировать базовые метрики (точность, F1) и проанализировать типы ошибок (например, на каких классах ошибаются модели).</p>	Результаты — "бейзлайн" для сравнения и источник гипотез для улучшения.
2. Поиск идей для улучшений	Агенты выполняют веб-скрапинг arXiv.org и конференций (e.g., NeurIPS, ICML) по ключевым словам, сгенерированным на основе анализа этапа 1 (например, "improving tree-based models on tabular data").	Найденные статьи станут основой для генерации кода на следующем этапе.
3. Генерация и интеграция кода	<p>На основе найденных статей агенты генерируют код новых моделей, методов предобработки или стратегий ансамблирования.</p> <p>Интегрируют их в конвейер выбранного AutoML-фреймворка (например, как кастомный эстиматор в Auto-sklearn).</p>	Созданный артефакт — расширенный пайплайн — будет валидирован на следующем этапе.

---

<b>4. Валидация и отчетность</b>	<p>Запустить расширенный пайплайн на тех же датасетах. Сравнить метрики с бейзлайном.</p> <p>Сгенерировать отчет с объяснением, за счет чего было достигнуто улучшение (или почему его не произошло).</p>	<p>Финальный отчет демонстрирует успешность или диагностирует неудачу всей цепочки работы агентов.</p>
----------------------------------	---	--

---

#### **Рекомендации по реализации:**

- **Длительность задачи** - 2-3 дня
- **Размер команды** - 5-6 человек уровня middle+
- Использование внешних моделей - не ограничены гигачатом
- Использование локальных моделей - ollama
- **Инструменты:** LangGraph/AutoGen для оркестрации агентов, SerpAPI/arXiv API для поиска, Docker для изоляции фреймворков.

#### Критерии качества теоретической реализации:

1. **Выбор датасетов.** Например доля датасетов с малым объемом данных, но высоким отношением числа признаков к наблюдениям или сложной структурой (например, KDDCup09\_upselling). Для регрессии - например доменные датасеты из материаловедения, где часто встречаются маленькие, но сложные выборки .
2. **Сложность логики.** Количество и осмысленность шагов reasoning-a (multi-hop logic), выполненных системой.
3. **Иновационность:** Креативность в предложенных улучшениях или комбинациях методов.

#### Критерии качества технической реализации:

1. **Качество модели.** Улучшение метрики (Accuracy, F1, RMSE) по сравнению с базовым прогоном лучшего из фреймворков.
2. **Стабильность работы.** Доля успешно завершенных экспериментов без «падений» агентов.

#### **Описание потенциальных “ролей” агентов в МАС-системе (опционально, команда может выбрать свою стратегию)**

1. **Data agent.** Подготовка и валидация данных из AMLB бенчмарка. В качестве этапов которые должен реализовать этот агент:
  - a. Загрузка и валидация данных
  - b. Разбиение на трейн и тест
  - c. Создание метапризнаков
  - d. Версионирование и мониторинг запусков экспериментов на конкретном датасете
2. **AutoML agent.** Запуск конкретных automl фреймворков на выбранном датасете. В качестве этапов которые должен реализовать этот агент:
  - a. Запуск фреймворка
  - b. Конфигурирование его гиперпараметров запуска
  - c. Сбор метрик и мониторинг числа “падений” фреймворка и их причин
3. **Research agent.** Запуск конкретных automl фреймворков на выбранном датасете. В качестве этапов которые должен реализовать этот агент:
  - a. Поиск статей на архив по ключевым словам
  - b. Извлечение идей и алгоритмов из статей
  - c. Ранжирование методов по “потенциальной эффективности”
4. **Research agent.** Запуск конкретных automl фреймворков на выбранном датасете. В качестве этапов которые должен реализовать этот агент:
  - a. Поиск статей на архив по ключевым словам
  - b. Извлечение идей и алгоритмов из статей
  - c. Ранжирование методов по “потенциальной эффективности”
5. **Code agent.** Генерация кода на основе результатов research агента:
  - a. Создание кода на основе результатов скрапинга рисерч агента
  - b. Создание совместных модулей с существующим AutoML
  - c. Создание документации и примеров кода