Hidden Technical Debt in Machine Learning Systems

Шемчик Евгений, БПМИ172

Технический долг

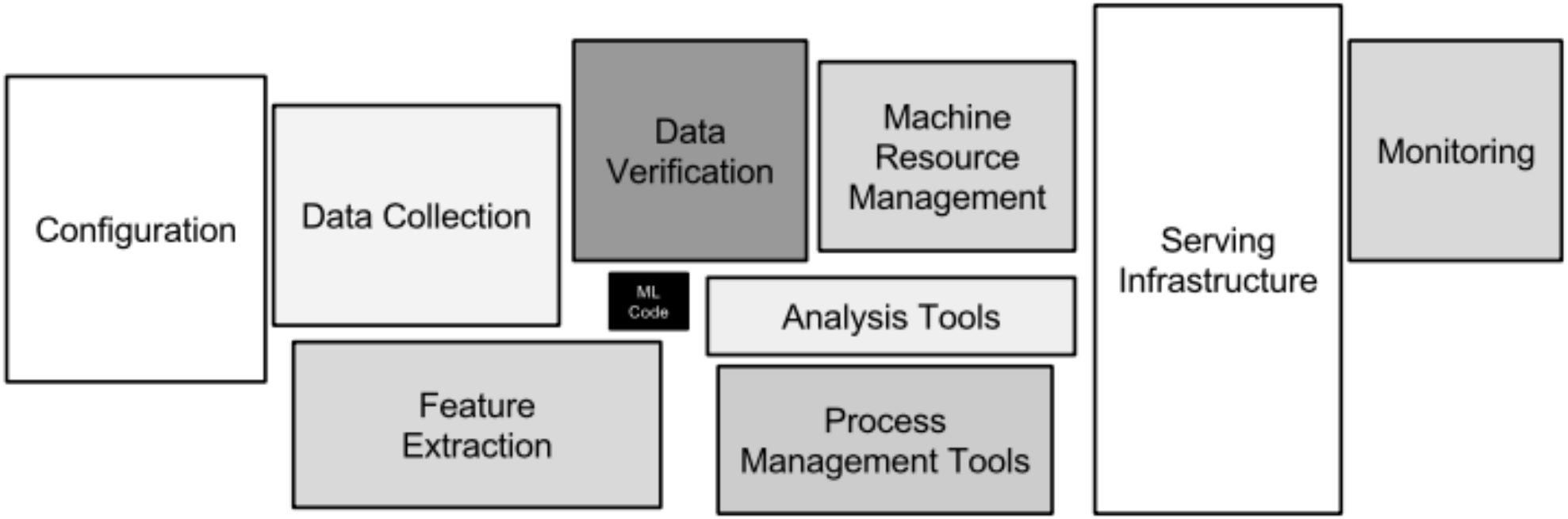
- Скорость разработки VS качество разработки

- Технический долг – накопленные проблемы в коде / архитектуре / инфраструктуре сервиса, требующие дополнительных трудозатрат на их устранение.

Общие проблемы техдолга в ML

- ML-система по определению зависит от внешних данных
- Недостаток абстракций (по сравнению, например с паттернами ООП)
- Недостаточное тестирование
- Отсутствие мониторингов
- Отсутствие культуры уменьшения техдолга

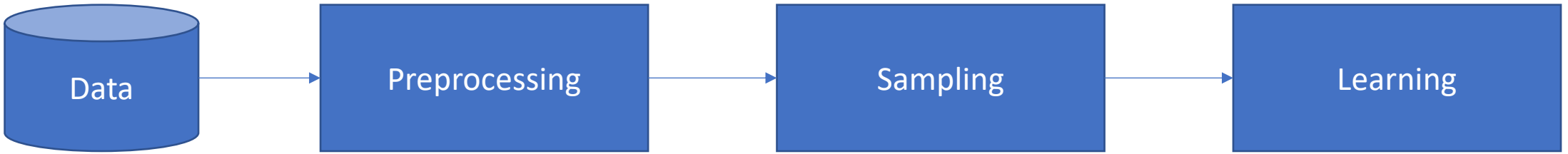
Распределение человеческих ресурсов при разработке ML-системы



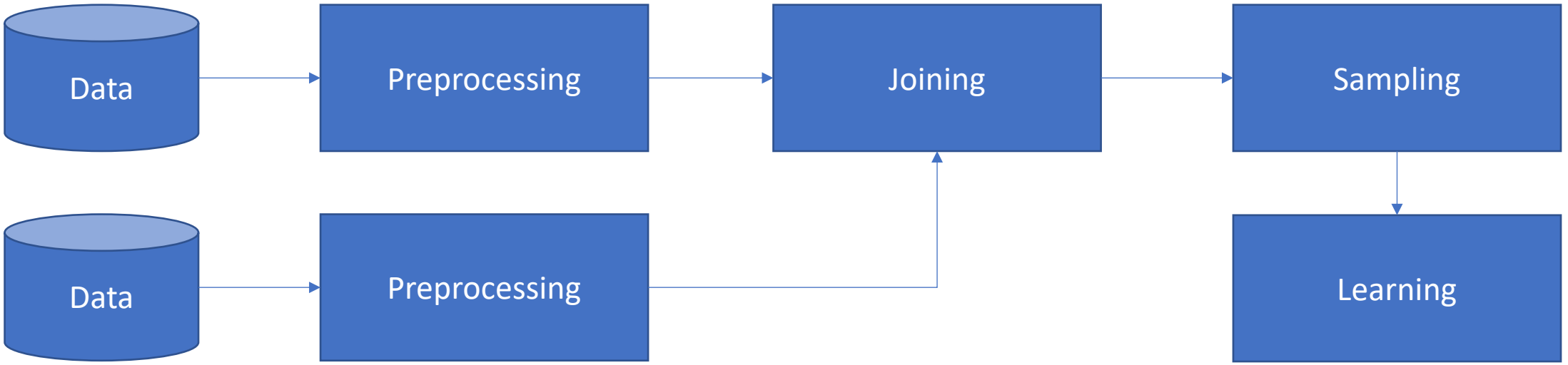
Техдолг в коде: использование пакетов для решения более общих задач

- Ограничены возможности использовать особенности задачи
- 95% составляет вспомогательный код
- Хорошая практика: «обёртки» для внешних библиотек

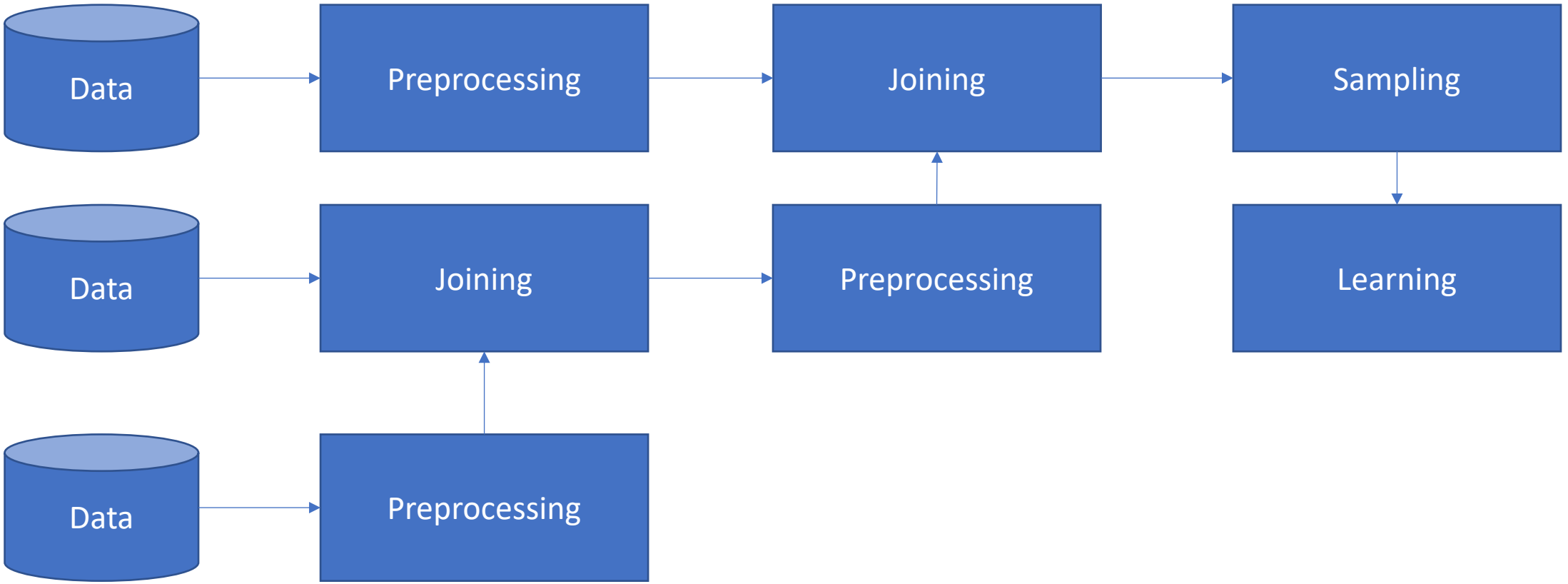
Техдолг в коде: pipeline jungles



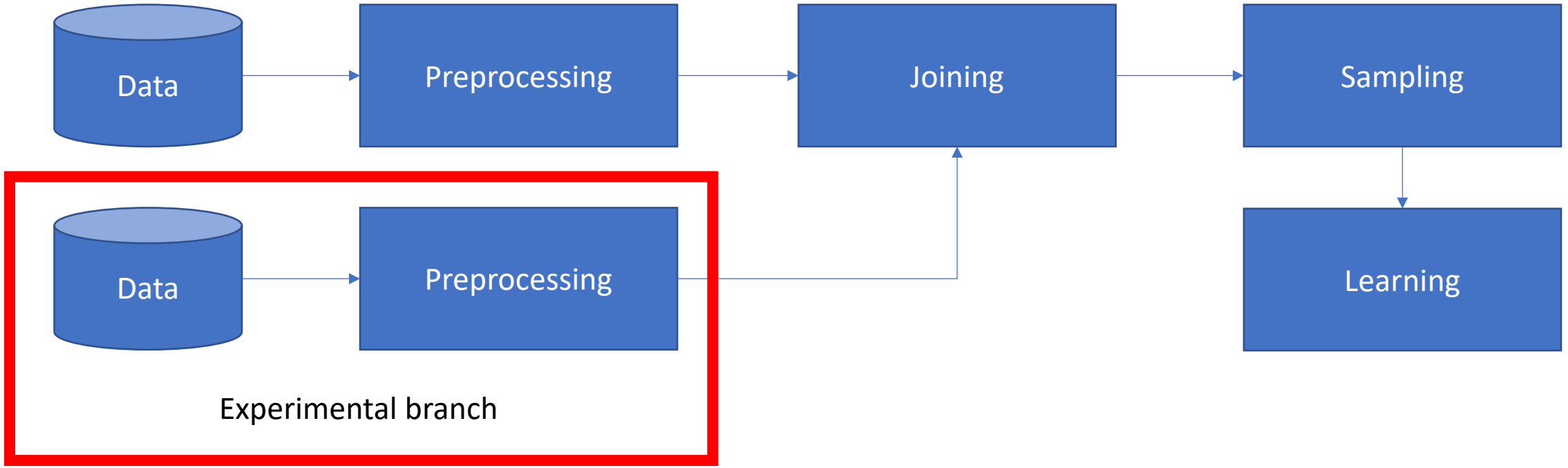
Техдолг в коде: pipeline jungles



Техдолг в коде: pipeline jungles



Техдолг в коде: перегруженность экспериментами



Вывод: не забывайте чистить экспериментальный код после принятия решения.

Техдолг в коде: общие ошибки

- Отсутствие изоляции логических частей кода
- Нарушения / недостатки системы абстракций
- Неоправданное применение нескольких языков программирования
- Plain-Old-Data Type Smell
- etc.

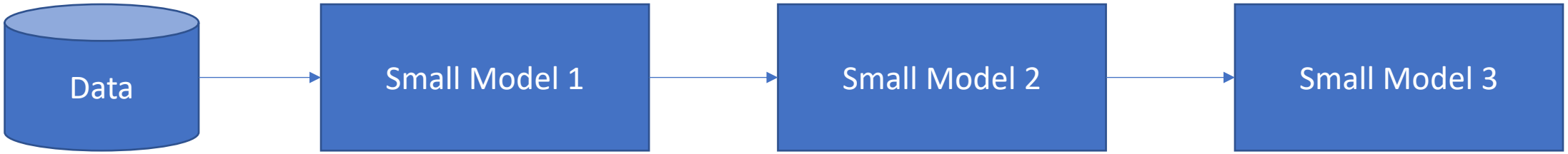
Техдолг в данных: внешние данные

- Отсутствие sanity check
- Отсутствует мониторинг изменений во внешних данных (например, мат. ожидания, дисперсии, etc)
- Отсутствуют end-to-end тесты для проверки корректности обработки данных
- Не пересматриваются пороги принятия решений

Техдолг в данных: лишние данные

- Устаревшие признаки
- Скоррелированные признаки
- Признаки дающие незначительные улучшения
  - Trade Off: качество модели VS затраты на поддержку признаков

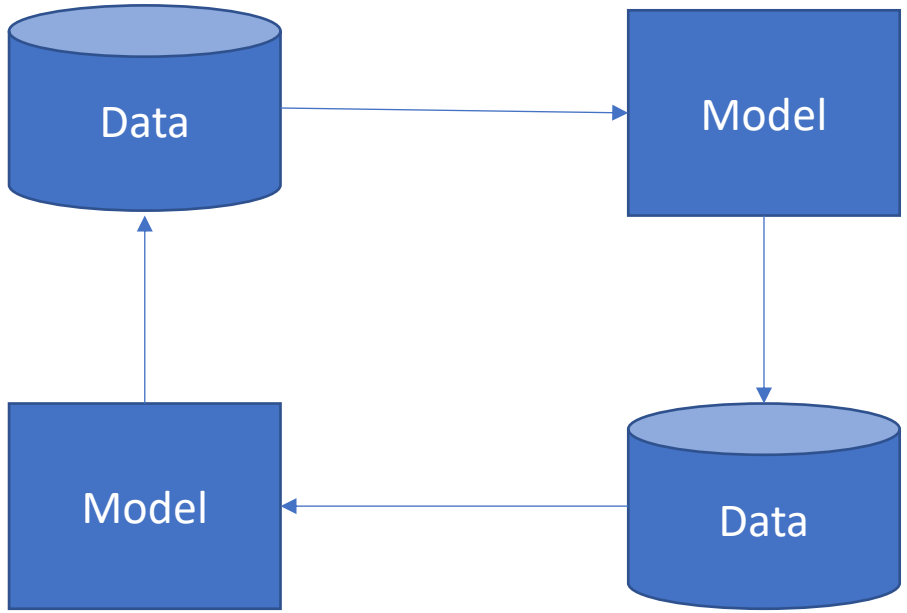
Техдолг в данных: пайплайны моделей



Fine-tune VS New Model

Техдолг в данных: feedback loop

- Явный feedback loop



Техдолг в данных: feedback loop

- Неявный feedback loop
- Пример:
  - Сервис показа рекламы в интернете хочет получать меньше жалоб на объявления
  - Блокировка лишних объявлений приведёт к уменьшению такой метрики
  - Классификатору выгодно перебанивать
  - Если он после этого учится на новых данных – проблема усугубляется

Техдолг в данных: visibility debt

- Доступ ко всем данным должен контролироваться, не должно быть неявных зависимостей.
- Аналогично в коде: отсутствие маркировки методов и полей как public/private приводит к некорректному использованию таких методов и нарушению системы абстракций.

Техдолг в моделях: нестабильность

- Поломка одного признака (перестал считаться / стал считаться неправильно), должна быть легко находима и не должна приводить к краху всей системы.
- Данные стоит хранить версионированно (как входные, так и выходные)

Техдолг в данных: неправильная конфигурация

- Не проверяется корректность конфигурации
- Изменения конфигурации тестируются скопом, без анализа эффектов отдельных изменений
- Для конфигураций должны выполняться базовые правила
  - It should be easy to specify a configuration as a small change from a previous configuration.
  - It should be hard to make manual errors, omissions, or oversights.
  - It should be easy to see, visually, the difference in configuration between two models.
  - It should be easy to automatically assert and verify basic facts about the configuration: number of features used, transitive closure of data dependencies, etc.
  - It should be possible to detect unused or redundant settings.
  - Configurations should undergo a full code review and be checked into a repository.

Выводы

- О техдолге стоит думать уже на этапе выстраивания процесса разработки
- От многих проблем могут спасти мониторинги и тестирование
- Игнорирование техдолга может привести к фактической остановке разработки новой функциональности
- Иногда приходится заново переписывать части системы ради избавления от техдолга

Вопросы

- Какие мониторинги стоит сделать для бинарного классификатора, который принимает решение на основе некоторого порога уверенности?
- В чём заключается сложность борьбы с неявным feedback loop?
- В каких ситуациях в контексте тех долга может возникнуть необходимость уменьшить качество модели?