

Срок - полгода.

Какие данные необходимо собирать с оборудования?

- Данные с датчиков на станках (на каждом типе станков свой набор датчиков). Пример оформления сгенерирован и выложен в sensors.md.
- Данные по использованию станков: модель станка, марка расходных материалов, время эксплуатации станка, даты ближайших и крайних технических работ.
- Перечень всех деталей дополнительной информацией для БД. Критерии выделены в файле databases.csv
- Перечень всех поломок деталей. В том же файле databases.csv

Требования к данным:

Доступность: Данные должны передаваться в реальном времени через IoT-шлюзы и храниться в облаке (Google Cloud).

Релевантность: Фильтрация шума (например вибрации с близлежащей железной дороги).

Полнота: Охват всех критических узлов (ножи, двигатели) и временных периодов (например архив за 3+ года).

Точность: Использование датчиков с погрешностью $\leq 1\%$ (может и больше, зависит от допусков) и регулярная калибровка.

Непротиворечивость: Единые форматы времени (UTC), единицы измерения ($^{\circ}\text{C}$, мм/с).

Своевременность: Минимальные задержки передачи, особенно для критических параметров (например поставить датчики с аппаратным hard real-time)

Осмысленность: Логическая связь данных с процессами (например, рост вибрации \rightarrow износ подшипников).

Как эти данные могут помочь предсказать неисправности?

Благодаря данным с датчиков можно вовремя диагностировать проблемы в работе станков. Эти данные архивируются и формируют костяк исторических данных, которые пойдут на обучение и дообучение ML-модели. Также:

- Появится возможность предсказывать какие детали нужно иметь в запасе на основе частоты поломок.
- Чаще получится вовремя проводить техническое обслуживание станков и деталей.
- Можно выявлять некоторые систематические ошибки сотрудников (наличие связи между поломками и каким-то конкретным сотрудником)

Это возможно благодаря постоянной диагностике станков с помощью датчиков. Исторические данные могут выступать в качестве какой-то нормы, отклонения от которой можно фиксировать. Так же они помогают предположить, какой будет выпуск деталей в конкретный момент времени, сравнивая с имеющимися данными. Информация о поломках даёт возможность определить оптимальное время для технического обслуживания станка, чтобы минимизировать простои. Также можно ребалансировать нагрузку, чтобы максимально эффективно использовать имеющиеся ресурсы.

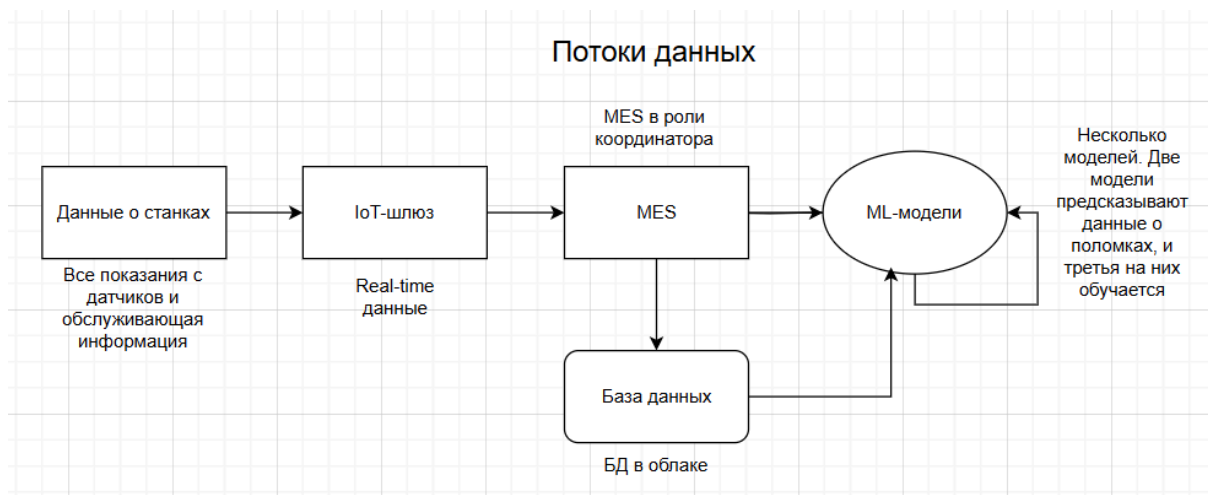
Метрики:

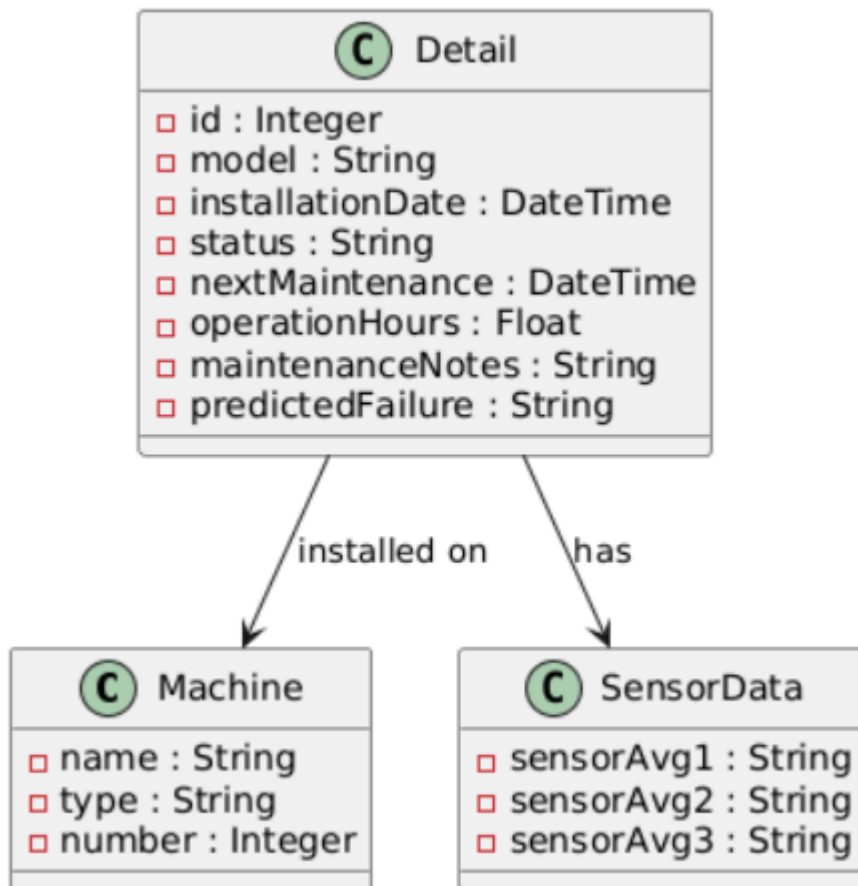
Технические: F1-score (баланс между ложными срабатываниями и пропусками).

Бизнес: Сокращение простоев на 20% за квартал.

Операционные: Время отклика <30 секунд.

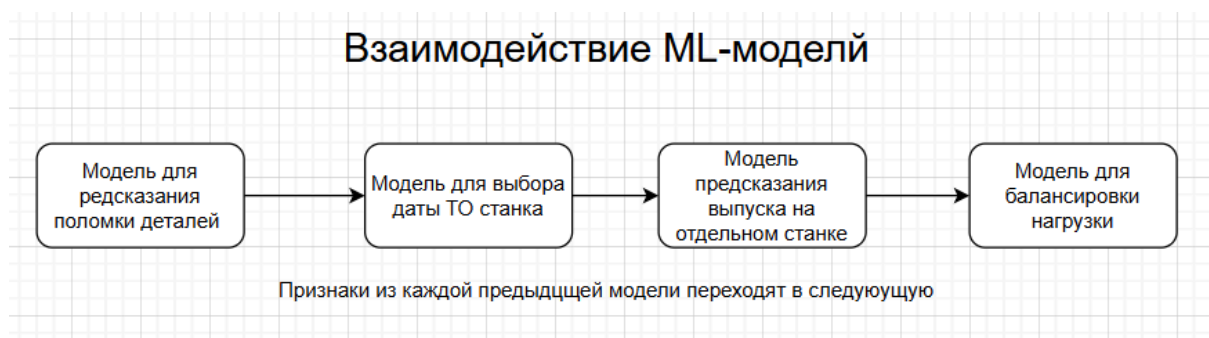
Хранение данных: данные загружаются в базы, для формирования исторических данных. Как правило, данные берутся из MES или с датчиков на станках, а потом загружаются в облако.





Класс деталей выступает в качестве базового для других классов, поскольку информация о нём учитывается во всех моделях при обучении (поломки деталей и их влияние на работоспособность).

Модели:



Так как цель предприятия повысить эффективность и стабильно достигать поставленных показателей, важно обращать внимание на выпуск каждого конкретного станка. Соответственно, важно понимать, какого значения может достигнуть каждый станок. И уже потом можно рассчитывать, как оптимизировать оказываемую нагрузку.

Градиентный бустинг неплохо себя показывает на регрессиях с нелинейными зависимостями. Поскольку неизвестно, как будет изменяться target, а данных много, и они разные, надёжнее считать, что зависимости нелинейны.

Регрессия для предсказания, через сколько сломается деталь. Градиентный бустинг.

Если отказаться от стремления узнать, относительно, точные сроки поломки детали, то можно рассматривать задачу как многоклассовую классификацию (срочность замены: критическая, срочная, несрочная, замена не требуется).

Регрессия для определения крайнего срока, когда нужно делать ТО станка, перед его поломкой. Градиентный бустинг

Регрессия для определения выпуска с одного станка. Градиентный бустинг.

Регрессия для определения оптимальной балансировки нагрузки на станков, для достижения поставленного показателя выпуска. Градиентный бустинг.

Предобработка

Все категории данных приводим к единым для своей категории параметров. Для времени - UTC, для показаний с датчиков - конвертация в систему СИ. Заменить данные с плавающей точкой на целочисленные там, где это возможно. Обработать пропуски значений (например в лекции рассказывали, что в SkLearn отсутствует встроенная обработка пропусков), выявить и устранить статистические аномалии (выбросы).

В качестве критериев оценки работоспособности модели может выступать количество часов в простое по отношению к среднему количеству часов в простое за какой-то промежуток времени из исторических данных. Также можно смотреть на разницу между количеством произведённых и количеством предсказанных деталей.

Внедрение и тестирование:

1. Подготовка инфраструктуры (Google Cloud, IoT-шлюзы, установка датчиков, написание модулей для MES) - месяц
2. Сбор Исторических данных и занесение их в базу данных - 2 недели
3. Обучение каждой модели + проверка на валидационной выборке - $4 * 1.5$ недели + $4 * 0.5$ недели = 2 месяца
4. Проверка на тестовых данных и калибровка - 2 недели
5. Аprobация на реальных данных - 2 месяца

Итог:

В результате, благодаря анализу данных с датчиков, информации о поломках и логированию этой информации по базам данных, можно предсказывать будущие поломки и минимизировать простои, что напрямую влияет на выпуск деталей и на прибыль компании. Также, если вести статистику по частоте поломок деталей, можно их заранее закупать в нужном объёме, что так же минимизирует простои.