**РЕШЕНИЕ**

**Предположение**

В реальном мире с не работающего мотора не приходят данные датчиков. То есть я предположил, что последний по номеру отсчет это последний рабочий период мотора и в следующий период он сломался. Дополнил файл Data\_80.csv колонкой Break и соответствующим образом заполнил. Получил обучающую выборку.

В постановке задачи указан **запрет** на получение дополнительных пояснений.

**Работа с данными**

Мы понимаем, что внутри временные ряды.

Посмотрев на данные, выяснили, что s1, s10, s16, s18, s19 и s5 это константы в обоих выборках. Исключаем из данных. В питоне.

Сдвигаем на один отсчет вперёд Setting1 – Setting2. В экселе.

Моторы вылетают начиная с 120+ цикла, поэтому один из параметров – с какого цикла учить модель … с 1/10/50/100? Тема к исследованию. В питоне.

**Моделирование**

С такой выборкой можно решать задачу регрессии – оценивать вероятность выхода из строя (0 – не выйдет, 1 – выйдет). И задачу классификации – оценивать в явном виде да\нет, 0 – не выйдет, 1 – выйдет.

Инструменты можно применить разные, от линейной регрессии (библиотека питона sklearn) до нейронных сетей (keras\tensorflow). Для такой задачи заходит бустинг и ann. Мне близок инструмент градиентного бустинга, вперёд использовал его в версии для классификации.

**Исследование**

1. Сделано быстрое решение – взят из запасов XGBoost драфт, быстро проведено моделирование и получен прогноз.
2. XGBoost драфт доработан, проведено исследование и получены СМ и прогноз.
   1. В СМ смотрим в правый нижний угол матрицы, необходимо, чтобы модель максимально предсказывала именно выход из строя.
3. В дополнение «развернул» временные данные, получилось дополнительное количество фич, повторил моделирование. Ну так себе.

Возможно, для бизнеса будет интереснее задача регрессии, а не классификации. Чтобы была возможность заблаговременно отправить ремонтников для профилактики.

1. **Меняю** модель – использую ANN на данных как в п.1.
2. Разворачиваю временные данные, получаю дополнительное количество фич, повторяю моделирование ANN.
3. Обернуть в регрессию п.1
4. Обернуть в регрессию п.2

Результаты

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Вариант** | **СМ** | **Ошибка1** | **Ошибка2** |  | **Польза** |
| 0 | - | - | - | - |  |
| 1 | 3207 | *5* | *13* | 3 | 0,9944 |
| 2 | 2409 | *0* | *17* | 2 | 0,9930 |
| 3 | 3210 | *2* | *13* | 3 | 0,9954 |
| 4 | 2808 | *7* | *9* | 4 | 0,9943 |
| 5 | 3209 | *3* | *12* | **4** | **0,9954** |
| 6 | 2813 | *2* | *12* | 1 | 0,9950 |

**Решение**

Вариант 5. Потому что машинное, а не глубокое обучение.

<https://github.com/IlyaEgorov800/Motor>

**Что ещё попробовать**

Запросить описание работы датчиков

Сделать визуализацию, поискать инсайты в графиках

Убрать лишние столбцы

Сгенерировать ещё фичи + - / \*

Сделать дополнительный параметрический тюнинг и\или автомл

Попробовать совсем другие ML и DL – например MLR из sklearn и rnn из keras\tensorflow

Из экстремального - поискать аномалии и через это определять поломку.