Дизайн систем машинного обучения

4. Подготовка и отбор признаков

План курса

1) ~~Практическое применение машинного обучения~~
2) ~~Основы проектирования ML-систем~~
3) ~~Обучающие данные~~
4) **Подготовка и отбор признаков — Вы находитесь здесь**
5) Выбор модели, разработка и обучение модели
6) Оценка качества модели
7) Развертывание систем
8) Диагностика ошибок и отказов ML-систем
9) Мониторинг и обучение на потоковых данных
10) Жизненный цикл модели
11) Отслеживание экспериментов и версионирование моделей
12) Сложные модели: временные ряды, модели над графами
13) Непредвзятость, безопасность, управление моделями
14) ML инфраструктура и платформы
15) Интеграция ML-систем в бизнес-процессы

Подготовка данных

Сделать так, чтобы алгоритму было проще учиться:

- Аугментация данных Data Augmentation
- Синтетические данные
- Очистка данных Data Cleaning
- Масштабирование/нормализация Scaling Normalisation
- Конструирование признаков Feature Engineering
- Отбор признаков Feature Selection

Аугментация данных

- Data augmentation
- Расширение датасета
- Адаптация к сдвигу данных
- Более устойчивые модели
- Во время обучения
- Во время предсказания TTA

https://albumentations.ai/

Аугментация

- Картинки
  - Поворот, обрезка, яркость, искажение
- Звуки
  - Добавление шума, замедление
- Текст
  - Двойной перевод на другой язык, синонимы, шум, перестановки
- Табличные данные
  - Сэмплирование, автоэнкодеры, добавление шума

Аугментация - библиотеки

- Albumentation https://albumentations.ai/
- Nlpaug https://github.com/makcedward/nlpaug
- EDA
- DTA
- Torchaudio
- Audiomentations и др
- SpecAugment

Синтетические данные - CV

- https://unity.com/ru/products/computer-vision

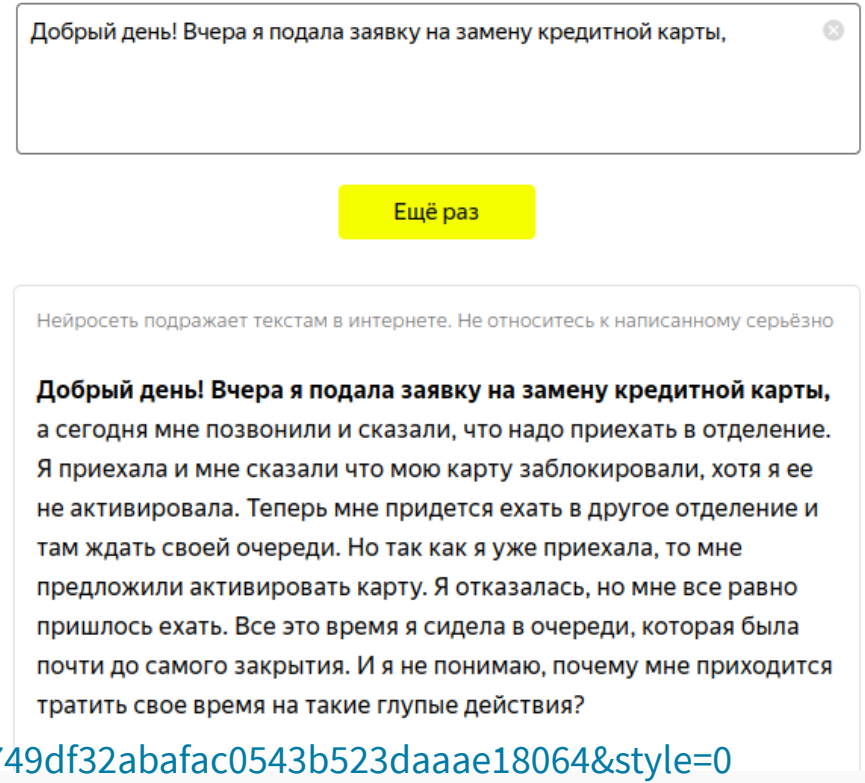
Синтетические данные - текст

- Большие языковые модели
- Перевод с другого языка
- Фильтрация датасета
- Генератор — например Faker

```python
from faker import Faker
fake = Faker()

fake.name()
# 'Lucy Cechtelar'

fake.address()
# '426 Jordy Lodge
#  Cartwrightshire, SC 88120-6700'
```



https://yandex.ru/lab/yalm/share?id=fd45c8a87749df32abafac0543b523daaae18064&style=0

Синтетические данные

- SDV
  - Табличные данные
  - Временные ряды
  - PyTorch
- Gretel
  - Табличные данные
  - Временные ряды (PyTorch)
  - Текст
  - Tensorflow

Признаки — строим или выучиваем?

- Классическое машинное обучение:
  - Ручное конструирование и отбор признаков
- Deep Learning
  - Признаки выучиваются и конструируются автоматически
- Разумно сочетаем оба подхода
  - Если мы что-то знаем, нужно сказать об этом модели
  - Выход простых моделей — признаки для сложных
  - Например, регрессия или SARIMA как признак для CatBoost

Масштабирование

- Для линейных моделей обязательно
  - sklearn.preprocessing
- Нормализация картинок для нейронной сети
  - torchvision.transforms.Normalize
- Исправление распределения признаков
  - PowerTransformer
  - QuantileTransformer

Пропущенные значения

- MCAR — Полностью случайные пропуски
- MAR — Случайные пропуски, зависящие от других признаков
  - Например, мальчиков и девочек осматривали разные врачи, кто-то был менее внимателен
- MNAR — Пропуски, зависящие от значения признака
  - Например, кто-то не раскрывает свой доход, потому что он маленький

Как работать с пропусками

- Удаление строк с пропусками
  - Если мало пропущеных значений
- Удаление столбцов с пропусками
  - Если значение заполнено менее чем в 5% строк, например
- Заполнение пропущеных значений Imputation
- MNAR — факт пропуска — значимый признак
- MCAR MAR — обычно пропуски можно игнорировать

Библиотеки для импутации

- missingno
- sklearn.impute
- fancyimpute
- GAIN
- transdim
- PyPOTS
- Imputer

Дискретизация и наоборот

- Количественную переменную в категориальную
- optbinning
- Разбиваем на квантили pandas.qcut
- Category Encoder
- Hashing trick

Преобразования

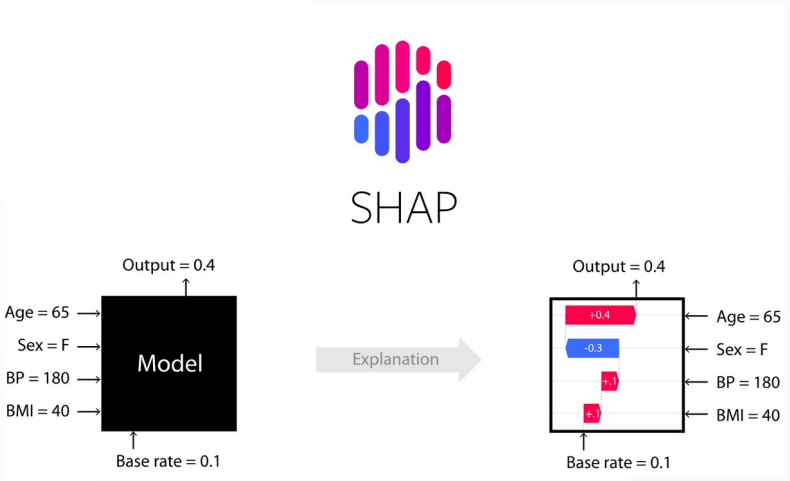
- Разбиение на элементы
  - Например, Дата: год, месяц, день, день недели, час
  - Например, адрес: страна, город, улица, номер дома
- Переход в полярные координаты
  - Расстояние от центра города
- Переход в декартовы координаты
- Снижение размерности PCA UMAP T-SNE
- Преобразование Фурье и т.д.

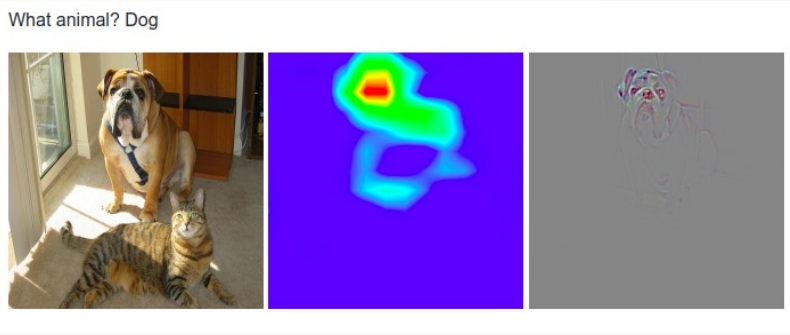
Выбросы / аномалии

- Выбросы outliers мешают линейным моделям
- Обращаться как с пропущенными значениями
- OCAR OAR ONAR ;-)
- Детектировать PyOD luminaire
- Удалять
- Ограничивать
- Откуда они берутся?

Важность признаков

- SHAP
- Interpret
- Captum
- Ferret
- grad-cam





Отбор признаков

- Выкидывать по одному RFECV
- Случайно перемешивать BoostARoota
- Смотреть, кто источник ошибки
- См Feature selection
- Осторожно с категориальными признаками
- Осторожно со скоррелированными признаками
- См http://www.feat.engineering/selection.html

Даталики

- Информация из тестового набора может «просачиваться» в обучающий набор. Это называют протечкой данных или DataLeak
- Даталик — любая информация, которая доступна модели при обучении, но недоступна при инференсе
- Даталики «завышают» метрику качества модели
- Далатики могут «отвлекать» внимание модели

Типичные источники даталиков

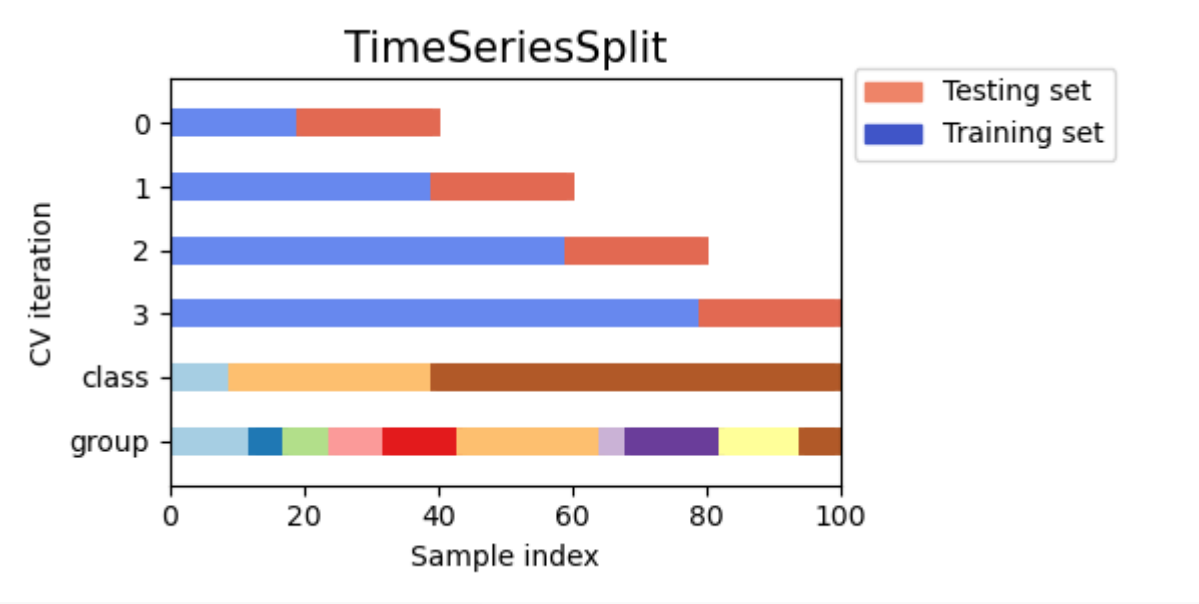
- Дубликаты
- Разбиение
- Масштабирование перед разбиением
- Импутация
- Генерация данных
- Групповая стратификация

Даталики: дубликаты

- Даталик: если дубликаты попадают в трейн и в тест
- Что делать:
  - Удалять дубликаты (искажаем распределение, если много)
  - Стратифицировать разбиение (все дубликаты в один из фолдов)
- Обычно дубликаты ухудшают качество модели
  - Deduplicating Training Data Makes Language Models Better

Даталики: разбиение

- Временные ряды нельзя разбивать случайно



https://scikit-learn.org/stable/modules/cross_validation.html#time-series-split

Даталики: масштабирование

- Масштабирование проводим после разбиения на трейн и тест
- Или внутри каждого фолда

Даталики: импутация

- Импутированные значения — информация о выборке в целом
- Импутацию проводим после разбиения на трейн и тест
- Или внутри каждого фолда

Даталики: пространственная корреляция

- Например, пространственная корреляция
  - Цены на недвижимость в одном здании
  - Посещаемость магазинов в одном и том же торговом центре
  - Если часть попадет в трейн, а часть в тест — будет даталик
- Решение — стратифицировать по группам
  - Например, кластеризация MeanShift
  - «Разделительная полоса» между трейном и тестом

Проблема — сложная стратификация

- Если нужно стратифицировать
  - Пространственно
  - По значению целевого признака
  - …
- Скорее всего, придется писать руками (несложно)
- Multi-Way Survey Stratification and Sampling
- Кое-что есть в R →
- Если найдете для python — напишите в чате, пожалуйста

Лики в процессе генерации данных

- Например:
  - Признак, производный от целевой переменной
  - Порядковый номер строчки
  - Подписи и оборудование на рентгеновских снимках
  - Качество и объем данных
  - Заглядывание в будущее

Дополнительные материалы

- Albumentations: Fast and Flexible Image Augmentations
- MEMO: Test Time Robustness via Adaptation and Augmentation
- EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks

Все будет в телеграм-канале