Лекция 2. Структура ML-проекта. Подготовка к решению задачи и начальная предварительная обработка данных

Практикум по программированию, 5 семестр

Иван Евгеньевич Бугаенко,
ассистент каф. ПМиФИ

Формат работы

Курс состоит из:

- 8 лекционных занятий (16 баллов)
- 7 лабораторных работ (35 баллов + 9 доп. баллов)
- домашнего задания (~~20~~15 баллов) = сбор всех лаб в одну работу
- демоэкзамена (~~20~~25 баллов)

Итого: 100 баллов

Все материалы выкладываются на wiki.pmifi.ru

Рейтинг ведется в таблице

Можно ли шаблонизировать ML-проект?

Каждая ML-задача уникальна, однако можно выделить базовый пайплайн по обработке данных

Основные шаги по работе с ML-проектом

0. Описание задачи и признаков
1. Предварительная обработка данных
2. Генерация новых признаков
3. Выбор ML-моделей и метрик качества
4. Обучение моделей, кросс-валидация и подбор гиперпараметров
5. Получение предсказаний на новых данных и вычисление метрик

Шаг 0. Описание задачи и признаков

На данном шаге исследователь собирает всю входную информацию о проекте:

- формулирует постановку задачи
- выполняет описание признакового пространства (определяет входные и целевые признаки, ограничения на значения, выписывает зависимости прикладной области)
- формирует виртуальное окружение (импорт библиотек, seed, фиксация версий библиотек для работы)
- + в самом конце – содержание работы (table of content)

Цель: сделать так, чтобы читатель понял, для чего решается задача, какая это вообще задача, а также чтобы работа была воспроизводима (можно было ее запустить с теми же параметрами)

Шаг 1. Предварительная обработка данных

На данном шаге исследователь выполняет базовые действия по работе с данными:

- загружает данные в рабочее пространство, выводит их и приводит признаки к необходимому типу (+ агрегация, парсинг дат, отбрасывание столбцов) – начальный этап обработки данных
- разделяет выборку на обучение и тест
- обрабатывает пропущенные значения
- выполняет кодировку категориальных признаков
- решает проблему наличия выбросов и аномалий в данных
- устраняет дисбаланс классов (для задачи классификации)

Цель: обработать данные так, чтобы можно было обучить baseline-модель

Baseline-модель – это такая наивная модель ML, которая обучалась на базовом варианте обработанных данных. Используется для проверки эффективности инструментов предобработки

Советы по начальному этапу обработки датасета

- всегда обращайте внимание на кодировку файла с данными и на разделитель (если речь про .csv и .tsv файлы)
- если датасет содержит даты, то передавайте их в параметр parse_dates
- обязательно выводите фрагмент датасета через DataFrame.head(), чтобы убедиться, что все корректно почиталось
- анализируйте при помощи DataFrame.info(), корректно ли парсятся типы данных при загрузке датасета (сверяйтесь с табличкой, содержащей инфу о семантике признаков); если типы не соответствуют, выполняйте ручное приведение
- для принятия решения об отбрасывании категориального признака смотрите на его семантику и количество уникальных значений (Series.nunique())
- обращайте внимание на признаки, которые неизвестны на момент получения прогноза, они подлежат удалению

Советы по начальному этапу обработки датасета

- агрегаты данных разбивайте на компоненты, если это не критично для задачи (например, дату можно разбить на день, месяц и год, если не решается задача анализа временных рядов)
- контролируйте, чтобы во входных признаках не было утечки данных

Типичные ошибки при делении выборки на train/test

- допускается утечка информации → рекомендуется сначала выполнять деление выборки, а потом выполнять действия только с train-датасетом (и результаты переносить на test)
- игнорируется временная структура данных → нужно контролировать, чтобы данные о "будущем" не попадали в обучающую выборку
- нарушение баланса классов → использовать стратификацию выборки при разделении (например, поле stratify=...)
- слишком большой/маленький train/test → контролировать объем выборки