



УНИВЕРСИТЕТ ИТМО

Лекция 4. Разработка приложений на основе подходов машинного обучения

Михаил А. Каканов¹ Олег А. Евстафьев¹

¹Факультет систем управления и робототехники, Университет ИТМО
{makakanov, oaevstafev}@itmo.ru

Октябрь 2021

Курс «Прикладной искусственный интеллект»

1. Жизненный цикл
2. Обработка данных
3. Разработка модели
4. Разработка программы

1. Жизненный цикл

2. Обработка данных

3. Разработка модели

4. Разработка программы

Цикл разработки состоит из трех фаз:

1

Обработка данных: сбор и подготовка данных.

2

Разработка модели: обучение и обслуживание модели машинного обучения.

3

Разработка программы: интеграция модели в конечный продукт.

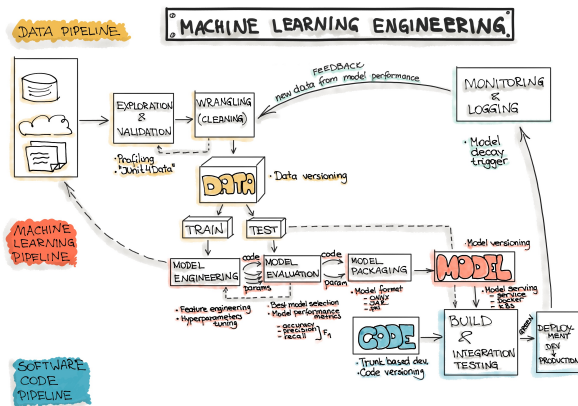


Рисунок 1 — Цикл разработки приложения на основе подходов машинного обучения © MLOps

1. Жизненный цикл

2. Обработка данных

3. Разработка модели

4. Разработка программы

Обработка данных состоит из следующих этапов:

1. **Сбор данных** — сбор данных с помощью различных механизмов и форматов, таких как Spark, HDFS, CSV и т. д. Этот этап может также включать генерацию синтетических данных или обогащение данных.
2. Исследование и проверка.
3. Очистка данных.
4. Маркировка данных.
5. Разделение данных.

Обработка данных состоит из следующих этапов:

1. Сбор данных.
2. **Исследование и проверка** — включает профилирование данных для получения информации о содержании и структуре данных. Результатом этого этапа является набор метаданных, таких как max, min, avg значений. Операции валидации данных - это определяемые пользователем функции обнаружения ошибок, которые сканируют набор данных с целью выявления некоторых ошибок.
3. Очистка данных.
4. Маркировка данных.
5. Разделение данных.

Обработка данных состоит из следующих этапов:

1. Сбор данных.
2. Исследование и проверка.
3. **Очистка данных (Data Wrangling)** — процесс переформатирования определенных атрибутов и исправления ошибок в данных, таких как подстановка отсутствующих значений.
4. Маркировка данных.
5. Разделение данных.

Обработка данных состоит из следующих этапов:

1. Сбор данных.
2. Исследование и проверка.
3. Очистка данных.
4. **Маркировка данных** — операция, при которой каждой точке данных присваивается определенная категория.
5. Разделение данных.

Обработка данных состоит из следующих этапов:

1. Сбор данных.
2. Исследование и проверка.
3. Очистка данных.
4. Маркировка данных.
5. **Разделение данных** — разделение данных на обучающие, проверочные и тестовые наборы, которые будут использоваться на основных этапах машинного обучения для создания ML-модели.

Обработка данных состоит из следующих этапов:

1. Сбор данных.
2. Исследование и проверка.
3. Очистка данных.
4. Маркировка данных.
5. Разделение данных.

Обработка данных состоит из следующих этапов:

1. Сбор данных.
2. Исследование и проверка.
3. Очистка данных.
4. Маркировка данных.
5. Разделение данных.

Источники данных, например:

- ▶ изображения,
- ▶ текстовые файлы на файловой системе,
- ▶ журналы, разбросанные по разным машинам,
- ▶ записи в базе данных.

Источники данных, например:

- ▶ изображения,
- ▶ текстовые файлы на файловой системе,
- ▶ журналы, разбросанные по разным машинам,
- ▶ записи в базе данных.

Источники данных, например:

- ▶ изображения,
- ▶ текстовые файлы на файловой системе,
- ▶ журналы, разбросанные по разным машинам,
- ▶ записи в базе данных.

Источники данных, например:

- ▶ изображения,
- ▶ текстовые файлы на файловой системе,
- ▶ журналы, разбросанные по разным машинам,
- ▶ записи в базе данных.

Способ переноса данных в обучаемый формат различен для каждого проекта и каждой компании. Например:

- ▶ Возможно, вы обучаете свои изображения на ImageNet, и все изображения - это просто URL-адреса S3. Тогда все, что вам нужно сделать, это загрузить их в локальную файловую систему.
- ▶ Может быть, у вас есть куча текстовых файлов, которые вы сами где-то раздобыли. Вы хотите использовать Spark для их обработки на кластере и Pandas data frame для анализа/выбора подмножеств, которые будут использоваться в локальной файловой системе.
- ▶ Возможно, вы собираете журналы и записи из своей базы данных в озеро данных/хранилище данных (например, Snowflake). Затем вы обрабатываете эти выходные данные и преобразуете их в обучаемый формат.

Способ переноса данных в обучаемый формат различен для каждого проекта и каждой компании. Например:

- ▶ Возможно, вы обучаете свои изображения на ImageNet, и все изображения - это просто URL-адреса S3. Тогда все, что вам нужно сделать, это загрузить их в локальную файловую систему.
- ▶ Может быть, у вас есть куча текстовых файлов, которые вы сами где-то раздобыли. Вы хотите использовать Spark для их обработки на кластере и Pandas data frame для анализа/выбора подмножеств, которые будут использоваться в локальной файловой системе.
- ▶ Возможно, вы собираете журналы и записи из своей базы данных в озеро данных/хранилище данных (например, Snowflake). Затем вы обрабатываете эти выходные данные и преобразуете их в обучаемый формат.

Способ переноса данных в обучаемый формат различен для каждого проекта и каждой компании. Например:

- ▶ Возможно, вы обучаете свои изображения на ImageNet, и все изображения - это просто URL-адреса S3. Тогда все, что вам нужно сделать, это загрузить их в локальную файловую систему.
- ▶ Может быть, у вас есть куча текстовых файлов, которые вы сами где-то раздобыли. Вы хотите использовать Spark для их обработки на кластере и Pandas data frame для анализа/выбора подмножеств, которые будут использоваться в локальной файловой системе.
- ▶ Возможно, вы собираете журналы и записи из своей базы данных в озеро данных/хранилище данных (например, Snowflake). Затем вы обрабатываете эти выходные данные и преобразуете их в обучаемый формат.



Рисунок 2 — Источники данных

Ключевые моменты, которые следует запомнить:

- ▶ Вы должны тратить в 10 раз больше времени, чем хотите, на изучение набора данных.
- ▶ Данные - это лучший способ улучшить общую производительность вашего ML-проекта
- ▶ Keep It Simple Stupid: важно не усложнять ситуацию и не превращать управление данными в ракетостроение.

Ключевые моменты, которые следует запомнить:

- ▶ Вы должны тратить в 10 раз больше времени, чем хотите, на изучение набора данных.
- ▶ Данные - это лучший способ улучшить общую производительность вашего ML-проекта
- ▶ Keep It Simple Stupid: важно не усложнять ситуацию и не превращать управление данными в ракетостроение.

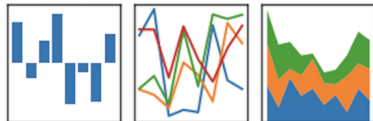
Ключевые моменты, которые следует запомнить:

- ▶ Вы должны тратить в 10 раз больше времени, чем хотите, на изучение набора данных.
- ▶ Данные - это лучший способ улучшить общую производительность вашего ML-проекта
- ▶ Keep It Simple Stupid: важно не усложнять ситуацию и не превращать управление данными в ракетостроение.

Цель исследования данных - понять и визуализировать природу данных, которые вы моделируете.

- ▶ Pandas - это рабочая лошадка Python для визуализации данных.

pandas
 $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



- ▶ Dask - это альтернатива, которая может ускорить обработку данных для больших наборов данных, с которыми Pandas не может справиться, за счет распараллеливания.
- ▶ Аналогичным образом RAPIDS ускоряет обработку больших наборов данных, хотя и за счет использования графических процессоров.

Цель исследования данных - понять и визуализировать природу данных, которые вы моделируете.

- ▶ Pandas - это рабочая лошадка Python для визуализации данных.
- ▶ Dask - это альтернатива, которая может ускорить обработку данных для больших наборов данных, с которыми Pandas не может справиться, за счет распараллеливания.



- ▶ Аналогичным образом RAPIDS ускоряет обработку больших наборов данных, хотя и за счет использования графических процессоров.

Цель исследования данных - понять и визуализировать природу данных, которые вы моделируете.

- ▶ Pandas - это рабочая лошадка Python для визуализации данных.
- ▶ Dask - это альтернатива, которая может ускорить обработку данных для больших наборов данных, с которыми Pandas не может справиться, за счет распараллеливания.
- ▶ Аналогичным образом RAPIDS ускоряет обработку больших наборов данных, хотя и за счет использования графических процессоров.

RAPIDS

- ▶ Эффективная маркировка данных является основным компонентом производственных систем машинного обучения.
- ▶ Чтобы избежать ошибок аннотаторов, следует написать четкое руководство, в котором разъясняются правила для крайних случаев и высококачественных аннотаций.
- ▶ Один из вариантов - нанять собственных аннотаторов, что поможет повысить скорость и качество аннотаций. Однако это может быть дорого и трудно масштабируемо.
- ▶ Другим вариантом является краудсорсинг меток через такую платформу, как Amazon Mechanical Turk, которая быстро и дешево создается, но ее качество может быть хуже.
- ▶ ...или компании, предоставляющие полный спектр услуг по маркировке данных.

- ▶ Эффективная маркировка данных является основным компонентом производственных систем машинного обучения.
- ▶ Чтобы избежать ошибок аннотаторов, следует написать четкое руководство, в котором разъясняются правила для крайних случаев и высококачественных аннотаций.
- ▶ Один из вариантов - нанять собственных аннотаторов, что поможет повысить скорость и качество аннотаций. Однако это может быть дорого и трудно масштабируемо.
- ▶ Другим вариантом является краудсорсинг меток через такую платформу, как Amazon Mechanical Turk, которая быстро и дешево создается, но ее качество может быть хуже.
- ▶ ...или компании, предоставляющие полный спектр услуг по маркировке данных.

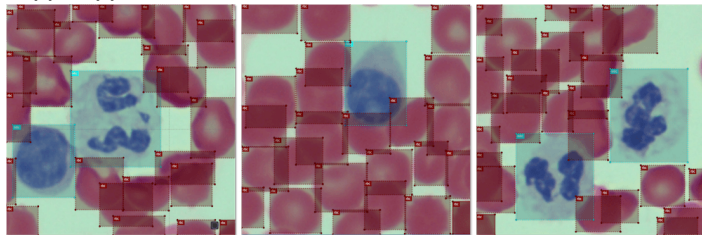
- ▶ Эффективная маркировка данных является основным компонентом производственных систем машинного обучения.
- ▶ Чтобы избежать ошибок аннотаторов, следует написать четкое руководство, в котором разъясняются правила для крайних случаев и высококачественных аннотаций.
- ▶ Один из вариантов - нанять собственных аннотаторов, что поможет повысить скорость и качество аннотаций. Однако это может быть дорого и трудно масштабируемо.
- ▶ Другим вариантом является краудсорсинг меток через такую платформу, как Amazon Mechanical Turk, которая быстро и дешево создается, но ее качество может быть хуже.
- ▶ ...или компании, предоставляющие полный спектр услуг по маркировке данных.

- ▶ Эффективная маркировка данных является основным компонентом производственных систем машинного обучения.
- ▶ Чтобы избежать ошибок аннотаторов, следует написать четкое руководство, в котором разъясняются правила для крайних случаев и высококачественных аннотаций.
- ▶ Один из вариантов - нанять собственных аннотаторов, что поможет повысить скорость и качество аннотаций. Однако это может быть дорого и трудно масштабируемо.
- ▶ Другим вариантом является краудсорсинг меток через такую платформу, как Amazon Mechanical Turk, которая быстро и дешево создается, но ее качество может быть хуже.
- ▶ ...или компании, предоставляющие полный спектр услуг по маркировке данных.

- ▶ Эффективная маркировка данных является основным компонентом производственных систем машинного обучения.
- ▶ Чтобы избежать ошибок аннотаторов, следует написать четкое руководство, в котором разъясняются правила для крайних случаев и высококачественных аннотаций.
- ▶ Один из вариантов - нанять собственных аннотаторов, что поможет повысить скорость и качество аннотаций. Однако это может быть дорого и трудно масштабируемо.
- ▶ Другим вариантом является краудсорсинг меток через такую платформу, как Amazon Mechanical Turk, которая быстро и дешево создается, но ее качество может быть хуже.
- ▶ ...или компании, предоставляющие полный спектр услуг по маркировке данных.

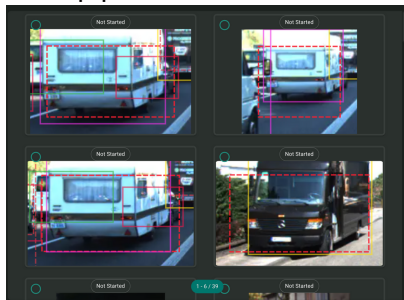
Если расходы на компанию, предоставляющую полный комплекс услуг по маркировке данных, непомерно высоки, можно воспользоваться чистым программным обеспечением для маркировки.

- ▶ Label Studio является дружественной платформой с открытым исходным кодом для этого.



- ▶ Aquarium помогает широко изучить данные и определить подходящую стратегию маркировки для классов, которые могут быть менее распространенными или эффективными.
- ▶ Snorkel ai предлагает платформу, которая автоматически маркирует

- ▶ Label Studio является дружелюбной платформой с открытым исходным кодом для этого.
- ▶ Aquarium помогает широко изучить данные и определить подходящую стратегию маркировки для классов, которые могут быть менее распространенными или эффективными.



- ▶ Snorkel.ai предлагает платформу, которая автоматически маркирует точки данных на основе эвристики и обратной связи с человеком.

- ▶ Label Studio является дружественной платформой с открытым исходным кодом для этого.
- ▶ Aquarium помогает широко изучить данные и определить подходящую стратегию маркировки для классов, которые могут быть менее распространенными или эффективными.
- ▶ Snorkel.ai предлагает платформу, которая автоматически маркирует точки данных на основе эвристики и обратной связи с человеком.

- ▶ В итоге, если вы можете позволить себе не маркировать данные, не маркируйте;
- ▶ Наймите компанию с полным спектром услуг, которая позаботится об этом.
- ▶ В противном случае попробуйте использовать существующее программное обеспечение.

- ▶ В итоге, если вы можете позволить себе не маркировать данные, не маркируйте;
- ▶ Наймите компанию с полным спектром услуг, которая позаботится об этом.
- ▶ В противном случае попробуйте использовать существующее программное обеспечение.

- ▶ В итоге, если вы можете позволить себе не маркировать данные, не маркируйте;
- ▶ Наймите компанию с полным спектром услуг, которая позаботится об этом.
- ▶ В противном случае попробуйте использовать существующее программное обеспечение.

1. Жизненный цикл

2. Обработка данных

3. Разработка модели

4. Разработка программы



Разработка модели состоит из следующих этапов:

1. **Обучение модели** — процесс применения алгоритма на данных для обучения ML-модели. Он также включает в себя разработку признаков и настройку гиперпараметров для обучения модели.
2. Оценка модели.
3. Тестирование модели.
4. Упаковка модели.

Разработка модели состоит из следующих этапов:

1. Обучение модели.
2. **Оценка модели** — проверка обученной модели на соответствие исходным кодифицированным целям перед тем, как предоставить модель ML в разработку продукта до конечного пользователя.
3. Тестирование модели.
4. Упаковка модели.

Разработка модели состоит из следующих этапов:

1. Обучение модели.
2. Оценка модели.
3. **Тестирование модели** — выполнение валидации модели с использованием тестового набора данных.
4. Упаковка модели.

Разработка модели состоит из следующих этапов:

1. Обучение модели.
2. Оценка модели.
3. Тестирование модели.
4. **Упаковка модели** — процесс экспорта окончательной модели ML в определенный формат (например, PMML, PFA или ONNX), который описывает модель, для того, чтобы ее могло использовать бизнес-приложение.

Разработка модели состоит из следующих этапов:

1. Обучение модели.
2. Оценка модели.
3. Тестирование модели.
4. Упаковка модели.

Разработка модели состоит из следующих этапов:

1. Обучение модели.
2. Оценка модели.
3. Тестирование модели.
4. Упаковка модели.

При многократном запуске модели как вы будете отслеживать влияние гиперпараметра?

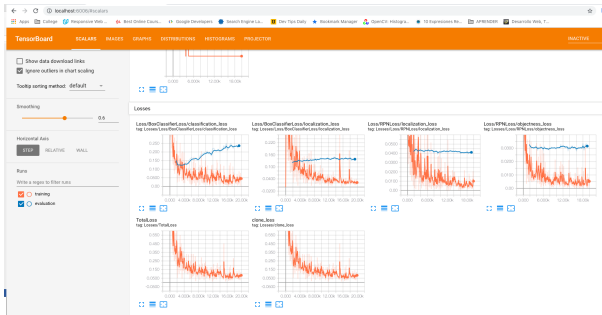
- ▶ По мере проведения многочисленных экспериментов для уточнения модели легко потерять информацию о коде, гиперпараметрах и артефактах.
- ▶ Итерации модели могут привести к большой сложности и беспорядку.
- ▶ Например, вы можете отслеживать влияние скорости обучения на метрику производительности вашей модели.

- ▶ По мере проведения многочисленных экспериментов для уточнения модели легко потерять информацию о коде, гиперпараметрах и артефактах.
- ▶ Итерации модели могут привести к большой сложности и беспорядку.
- ▶ Например, вы можете отслеживать влияние скорости обучения на метрику производительности вашей модели.

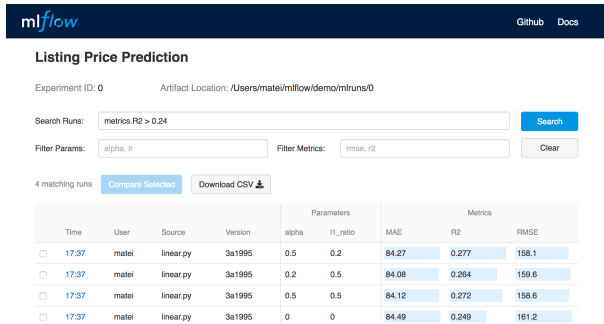
- ▶ По мере проведения многочисленных экспериментов для уточнения модели легко потерять информацию о коде, гиперпараметрах и артефактах.
- ▶ Итерации модели могут привести к большой сложности и беспорядку.
- ▶ Например, вы можете отслеживать влияние скорости обучения на метрику производительности вашей модели.

Специальные платформы управления экспериментами являются решением этих проблем. Давайте рассмотрим несколько наиболее распространенных из них:

- **TensorBoard**: платформа поставляется с TensorFlow. Плохо подходит для отслеживания и сравнения нескольких экспериментов, но легка в освоении.



- ▶ TensorBoard.
- ▶ **MLFlow**: полная платформа для жизненного цикла ML. В основе платформы лежит отличное управление экспериментами и прогонами моделей.



The screenshot shows the MLFlow web interface for an experiment titled "Listing Price Prediction". At the top, there are links for "Github" and "Docs". Below the title, the "Experiment ID" is 0 and the "Artifact Location" is /Users/matei/mlflow/demo/mlruns/0. A search bar contains the query "metrics.R2 > 0.24" with a "Search" button. Below the search bar, there are filters for "Filter Params" (alpha, lr) and "Filter Metrics" (rmse, r2), with a "Clear" button. A section indicates "4 matching runs" with buttons for "Compare Selected" and "Download CSV". A table displays the results of these runs, with columns for Time, User, Source, Version, Parameters (alpha, l1_ratio), and Metrics (MAE, R2, RMSE).

	Time	User	Source	Version	Parameters		Metrics		
					alpha	l1_ratio	MAE	R2	RMSE
<input type="checkbox"/>	17:37	matei	linear.py	3a1995	0.5	0.2	84.27	0.277	158.1
<input type="checkbox"/>	17:37	matei	linear.py	3a1995	0.2	0.5	84.08	0.264	159.6
<input type="checkbox"/>	17:37	matei	linear.py	3a1995	0.5	0.5	84.12	0.272	158.6
<input type="checkbox"/>	17:37	matei	linear.py	3a1995	0	0	84.49	0.249	161.2

- ▶ Платные платформы (Comet.ml, Weights and Biases, Neptune).
- ▶ другие...

- ▶ TensorBoard.
- ▶ MLFlow.
- ▶ **Платные платформы (Comet.ml, Weights and Biases, Neptune):**
основательные платформы для управления экспериментами, с такими инструментами, как диффиксы кода, составление отчетов, визуализация данных и функции регистрации моделей.
- ▶ другие...

- ▶ TensorBoard.
- ▶ MLFlow.
- ▶ Платные платформы (Comet.ml, Weights and Biases, Neptune).
- ▶ другие...

1. Жизненный цикл

2. Обработка данных

3. Разработка модели

4. Разработка программы

Заключительным этапом является интеграция ранее разработанной ML-модели в существующее программное обеспечение. Этот этап включает в себя следующие операции:

1. **Обслуживание модели** — процесс обращения к артефакту модели ML в производственной среде.
2. Мониторинг производительности модели.
3. Ведение журнала производительности модели.

Этот этап включает в себя следующие операции:

1. Обслуживание модели.
2. **Мониторинг производительности модели** — процесс наблюдения за производительностью модели ML на основе живых и ранее невидимых данных, таких как предсказания или рекомендации.
3. Ведение журнала производительности модели.

Этот этап включает в себя следующие операции:

1. Обслуживание модели.
2. Мониторинг производительности модели.
3. **Ведение журнала производительности модели** — Каждый запрос на вывод приводит к записи в журнале.

Этот этап включает в себя следующие операции:

1. Обслуживание модели.
2. Мониторинг производительности модели.
3. Ведение журнала производительности модели.

Этот этап включает в себя следующие операции:

1. Обслуживание модели.
2. Мониторинг производительности модели.
3. Ведение журнала производительности модели.

После того как вы разработали модель, необходимо её интегрировать в бизнес-процесс, для этого можно воспользоваться сервисами:

- ▶ Heroku
- ▶ PythonAnywhere
- ▶ Algorithmia

где разместим, например, веб-приложение написанное при помощи фреймворков:

- ▶ Flask
- ▶ FastAPI
- ▶ Django

- ▶ Процесс разработки приложения на базе ML требует большого спектра навыков: от анализа данных до разработки программного обеспечения.
- ▶ Работа с данными требует значительного времени, ведь от их качества зависит эффективность всего приложения в целом.
- ▶ Обработка данных процесс творческий и трудоемкий, но для автоматизации рутины существует множество готовых решений.
- ▶ При разработке модели следует обратить внимание на логирование промежуточных результатов.
- ▶ Разработанная модель \neq готовый продукт.

