2022

Лекция 1. Введение

МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ

ПАПУЛИН С.Ю. (papulin.study@yandex.ru)

Дополнительные материалы доступны по ссылке: https://github.com/mlmethods

Содержание

1. Структура и содержание курса 2
1.1. Темы лекций и семинаров 2
1.2. Текущие и промежуточный контроли 3
2. Введение в машинное обучение 4
3. Основные обозначения 5
4. Классы методов машинного обучения 6
4.1. Обучение с учителем 6
4.2. Обучение без учителя 7
4.3. Обучение с частичным привлечением учителя (semi-supervised) 7
5. Основные задачи машинного обучения 8
5.1. Типы переменных 8
5.2. Набор данных 8
5.3. Регрессия 8
5.4. Переобучение 9
5.5. Классификация 9
5.6. Кластеризация 11
5.7. Уменьшение размерности 11
5.8. Рекомендательные системы 12
5.9. Ансамбли методов 12
6. Параметрические и непараметрические методы 12
7. Предсказание и вывод 13
8. Точность и интерпретируемости 14
9. Основные этапы построения модели 15
Список литературы 16

1. Структура и содержание курса

1.1. Темы лекций и семинаров

~17 лекций + ~17 семинаров

Регрессия

- Метод ближайших соседей (kNN)
- Линейная регрессия (Linear Regression)
- Байесовская линейная регрессия (Bayesian Linear Regression)
- Метод опорных векторов для регрессии (Support Vector Regression)
- Деревья решений (Classification and Regression Tree (CART) (бинарное дерево), ID3)
- Нейронные сети

Классификация

- Метод ближайших соседей (kNN)
- Логистическая регрессия (Logistic Regression)
- Метод опорных векторов (SVM)
- Наивный байесовский классификатор (Naive Bayesian Classifier)
- Деревья решений (CART, ID3, C4.5)
- Нейронные сети

Оптимизация

- Метод наименьших квадратов
- Градиентный спуск
- Стохастический градиентный спуск
- Условная оптимизация (constrained optimization):
    - Метод множителей Лагранжа (Lagrange Multipliers

Ансамбли методов

- Бэггинг (Bootstrap Aggregating – bagging)
- Бустинг (boosting)
- Стэкинг (stacking)

Выбор модели

- Корректировка ошибки обучения
- Отложенная выборка
- Кросс-валидация

Кластеризация

- Метод k-средних
- Иерархическая кластеризация
- Метод на основе плотности (DBSCAN)
- Смесь гауссовских моделей (Gausian Mixture Models)

Уменьшение размерности

- L1 регуляризация
- Метод главных компонент

Рекомендательные системы

- Контентные
- Коллаборативная фильтрация
- Факторизация матрицы рейтингов

Распределенные алгоритмы

- Статистики
- Расчет косинусного сходства
- Градиентный спуск
- Стохастический градиентный спуск
- Факторизация матрицы посредством ALS

1.2. Текущие и промежуточный контроли

Модуль 1

ДЗ1:

- Часть 1 (10 баллов)
- Часть 2 (10 баллов)

$$\text{РК1} = 35/20 \cdot (K1.1 \cdot \text{ДЗ1. Часть 1} + K1.2 \cdot \text{ДЗ1. Часть 2})$$

$$\text{РК1} = K1.1 \cdot \text{ДЗ1. Часть 1} + K1.2 \cdot \text{ДЗ1. Часть 2} + \text{Вопросы (15 баллов)}$$

K – коэффициент (1; 0.85; 0.7)

Модуль 2

ДЗ2:

- Часть 1 (10 баллов)
- Часть 2 (10 баллов)

$$\text{РК2} = 35/20 \cdot (K2.1 \cdot \text{ДЗ2. Часть 1} + K2.2 \cdot \text{ДЗ2. Часть 2})$$

$$\text{РК2} = K2.1 \cdot \text{ДЗ2. Часть 1} + K2.2 \cdot \text{ДЗ2. Часть 2} + \text{Вопросы (15 баллов)}$$

Экзамен

30 баллов

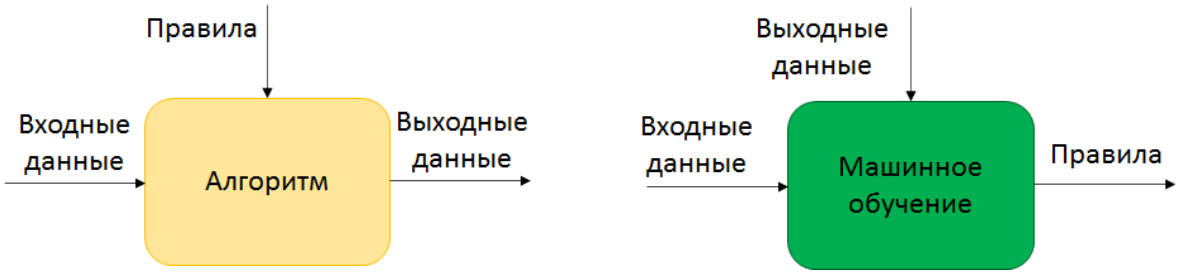
2. Введение в машинное обучение

Задача машинного обучения заключается в обучении программы выполнять определенные задачи с учетом некоторого набора правил. Важным вопросом является как получить эти правила.

Самым простым вариантом является задание жестко закодированных правил для получения откликов на входные данные. В этом случае используется, например, экспертная оценка.

Проблема в том, что в этом случае программа не может обучаться. И как следствие не адаптируется к изменениям. Программисту или оператору необходимо самостоятельно вносить правки в код или правила.

При машинном обучении программа должна сама выявлять правила в явном или неявном виде, определяя взаимосвязь между входными и выходными данными. Таким образом, алгоритмы самостоятельно выискивают шаблоны в данных и применяют обобщение. Выявленные правила в дальнейшем используются для предсказания или принятия решения на основе входных данных.



а) б)

Рисунок – Принципы обучения систем

Машинное обучение можно рассматривать как раздел искусственного интеллекта, наравне с системами основанных на знаниях (например, экспертными системами) или робототехникой.

Машинное обучения охватывает:

- Математический анализ
- Линейную алгебру
- Теорию оптимизации
- Теорию вероятностей и статистику

При этом машинное обучение не является чисто математической дисциплиной со строгими правилами и законами. Её скорее можно отнести к инженерной, так как применение тех или иных подходов и методов зависит от конкретной поставленной задачи и исходных данных.

3. Основные обозначения

Область определения (domain set):

$$X$$

x – вектор признаков, независимые переменные, предикторы, объясняющие переменные

$$x \in X$$

$$x \in \mathbb{R}^p$$

Область значений целевой функции (label set):

$$Y$$

y – целевое значение (target), метка (label), истинное значение (true value), действительное значение (actual value), наблюдаемое значение (observed value), зависимая переменная

$$y \in Y$$

Регрессия: $y \in \mathbb{R}$

Классификация: $y \in \{0,1\}$

Вектор признаков для i наблюдения:

$$x_i = \left(x_{i1}, x_{i2}, \dots, x_{ip}\right)^T$$

Вектор значений признака j для всего набора данных:

$$\mathrm{x}_j = \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{pmatrix}$$

Матрица признаков всех наблюдений:

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}$$

Вектор целевых значений:

$$\mathrm{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Вектор предсказаний:

$$\hat{y} = \begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{pmatrix}$$

Действительная функция (неизвестна):

$$f: X \to Y$$

$$y_i = f(x_i)$$

Основная задача – оценка неизвестной функции f

Функция предсказания (гипотеза) – модель:

$$h: X \to Y$$

$$y_i = h(x_i) + \epsilon$$

Оценка функции f через оценку функции предсказания h:

$$\hat{h}: X \to Y$$

$$\hat{y}_i = \hat{h}(x_i)$$

Основная задача – оценка неизвестной функции f

Обучающее множество (training set):

$$S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$$

$$(x_i, y_i) \in X \times Y$$

4. Классы методов машинного обучения

4.1. Обучение с учителем

Для каждого наблюдения $i = 1 \dots n$, существует значение признаков x_i (предикторов) и связанное с ним целевое значение y_i (отклик).

Необходимо построить модель, которая определяет взаимосвязь между признаками и целевыми значением.

Две основные задачи:

- Предсказание (prediction): для предсказания откликов y на новые наблюдения x
- Вывод (inference): для лучшего понимания взаимосвязи между откликом y и признаками x

Примеры:

- Регрессия
- Классификация
- Ранжирование
- Прогнозирование

4.2. Обучение без учителя

Используется для определения взаимосвязи между переменными или между наблюдениями, например, кластеризация. Цель кластерного анализа заключается в установлении на основе значений $x_1, \cdots, x_n$ попадает ли наблюдение в относительно отдельную группу.

В данном случае для каждого наблюдения $i = 1 \ldots n$ мы имеем вектор измерений x_i, при этом ассоциируемый с ним отклик y_i отсутствует. Условно это можно представить, как формирование значений y_i только по имеющимся значениям x_i.

Примеры:

- Кластеризация
- Уменьшение размерности
- Определение выбросов

4.3. Обучение с частичным привлечением учителя (semi-supervised)

Допустим у нас есть множество из n наблюдений. Для m наблюдений, где $m < n$, у нас есть значения предикторов и отклика. Для оставшихся $n - m$ наблюдений есть только значения предикторов, но нет значений отклика. Как правило, $m \ll n$

Таким образом, методы обучения с частичным привлечениям учителя – это те, которые для построения моделей используют всю имеющуюся информация по наблюдениям с откликом и без.

Данный подход может быть применим к задачам обучения с учителем и без.

5. Основные задачи машинного обучения

5.1. Типы переменных

Переменные можно разделить на количественные и качественные (категориальные).

Количественные переменные принимают числовые значения (непрерывные и дискретные значения). Например, возраст, рост, доход, стоимость жилья, акций и пр.

Качественные переменные принимают значения одного из K различных классов или категорий. Примеры, пол человека, марка машины, диагноз, категория товара. Можно выделить порядковые, номинальные и бинарные переменные.

5.2. Набор данных

Как правило, мы ограничены в количестве исходных данных. Поэтому встает вопрос о рациональном их использовании для обучения. Стандартной техникой является разделение исходного множества наблюдений на обучающее и тестовое подмножества. Обучающая часть применяется для обучения модели предсказания, в то время как тестовая используется для оценки качества предсказания модели на данных, которые модель не видела в процессе обучения.

5.3. Регрессия

Регрессия относится к задаче с количественным откликом

Примеры:

- Предсказание продаж
- Прогноз цены на недвижимость
- Предсказание потребления бензина по характеристикам автомобиля
- Предсказание количества просмотров ресурса

Оценка качества

Для оценки качества моделей предсказания необходимо измерить то, как хорошо предсказания соответствуют наблюдаемым данным.

Для этого необходимо количественно оценить степень близости предсказанного значения отклика для заданного наблюдения и действительного значения отклика этого наблюдения.

Среднеквадратическая ошибка (Mean Squared Error – MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{h}(x_i))^2$$

где $\hat{h}(x_i)$ – предсказание для i-го наблюдения.

MSE будет небольшим, если предсказанное целевое значение очень близко к действительному значению, и будет большим, если для некоторых наблюдений предсказанные и действительные значения существенно отличаются.

Другие метрики

- MAE (средняя абсолютная ошибка)
- R^2 (коэффициент детерминации)

5.4. Переобучение

Обучающие данные используются для обучения модели. Ошибка обучения – это ошибка (пример, MSE), вычисляемая на обучающих данных. Однако наибольший интерес представляет не то, как модель предсказывает на обучающем подмножестве, а точность предсказания на новых данных, с которыми модель ранее не работала, т. е. на тестовых данных.

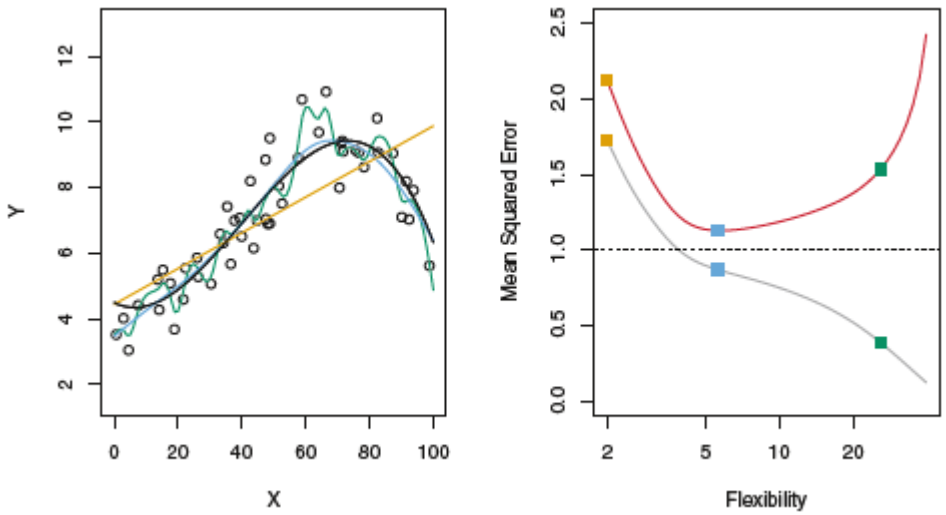


Рисунок – Гибкость модели и переобучение [1]

Как правило, ошибка обучения уменьшается с увеличением гибкости модели. В то время как тестовая ошибка может и нет.

В случае, когда модель дает малую обучающую ошибку (ошибку на обучающих данных) и большую тестовую ошибку (ошибку на тестовых данных), это свидетельствует о переобучении модели. Переобучение часто можно наблюдать при использовании гибких моделей (например, с большим количество параметров), обученных на небольшом наборе данных.

5.5. Классификация

Мы говорим про задачу классификации, когда предсказание является категориальным значением.

Предполагается, что обучающие данные состоят из наблюдений вида

$$\{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)\},$$

где y_i – категориальное значение i-го наблюдения.

Примеры:

- Медицинская диагностика
- Кредитный скоринг
- Распознавание образов

Оценка качества

Доля ошибок классификации:

$$ErrorRate = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

Доля правильных классификаций:

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n} I(y_i = \hat{y}_i),$$

где y_i и $\hat{y}_i$ – действительная и предсказанная метка класса для i-го наблюдения;

$$I(y_i = \hat{y}_i) = \begin{cases} 1, \text{если } y_i = \hat{y}_i \\ 0 \text{ иначе} \end{cases}$$

Хороший классификатор – это тот, у которого доля ошибок (error rate) на тестовых данных наименьшая или доля правильных классификаций (accuracy) наибольшая.

Другие метрики

- Recall/Precision (точность/полнота)
- F1
- ROC
- AUC

Пример

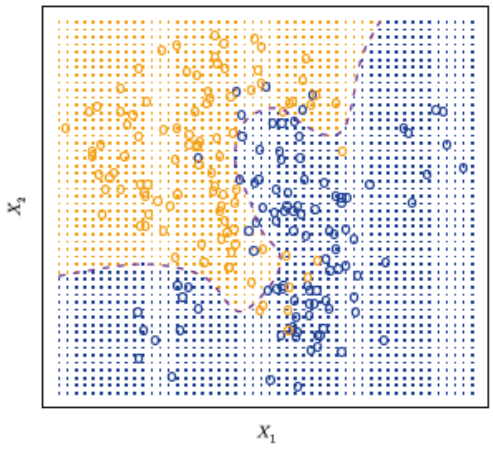


Рисунок – Пример бинарной классификации с двумя признаками [1]

5.6. Кластеризация

Целью кластеризации является разделение наблюдений на несколько групп так, чтобы наблюдения из одной группы были похожи, а из разных – отличались друг от друга.

Методы:

- Метод k-средних (k-means)
- Иерархическая кластеризация
    - Агломеративные методы (agglomerative)
    - Дивизионные методы (divisive)
- DBSCAN (Основанная на плотности пространственная кластеризация для приложений с шумами)
- BIRCH (Сбалансированное итеративное сокращение и кластеризация с помощью иерархий) – большие данные
- Спектральная кластеризация (Spectral clustering)

Примеры:

- Сегментация клиентов
- Тематическое моделирование
- Выявление аномалий

5.7. Уменьшение размерности

Целью данной задачи является преобразование данных размерности N в данные размерности M, такой что $M \ll N$, с сохранением информативности. Пример: сжатие изображений

В машинном обучении может быть использовано для уменьшения количества признаков входных данных модели при сохранении качества.

- Метод главных компонент (Principal Component Analysis – PCA)
- Анализ независимых компонент (Independent component analysis – ICA)
- Линейный дискриминантный анализ (Linear Discriminant Analysis – LDA) (**supervised**)
- Isometric Mapping – Isomap

5.8. Рекомендательные системы

Предсказывают альтернативы, которые могут заинтересовать пользователя.

Методы:

- Контентные методы
- Коллаборативная фильтрация по схожести пользователей (user-based collaborative filtering)
- Коллаборативная фильтрация по схожести объектов (item-based collaborative filtering)
- Факторизация матрицы рейтингов (matrix factorization)

5.9. Ансамбли методов

Ансамблевые методы используются для улучшения обобщающей способности предсказания в случаях, когда применение одной модели дает неудовлетворительный результат в виду её чрезмерной жесткости или гибкости.

Выделяют два базовых подхода: с параллельным (независимым) и последовательным обучением.

Известны следующие техники:

- Бэггинг (Bootstrap Aggregating – bagging)
- Бустинг (boosting)
- Стэкинг (stacking)

6. Параметрические и непараметрические методы

Наша цель – оценить неизвестную функцию f, используя обучающие данные, то есть найти такую $\hat{f}$, что для каждого наблюдения x

$$y \approx \hat{f}(x)$$

Параметрические (parametric)

Параметрические модели сводят задачу по оценке f до оценки набора параметров. В этом случае проще, например, оценить параметры линейной модели, чем подобрать одну подходящую функцию из всех возможных вариантов функций.

Выбор модели предсказания («формы» функции оценки f):

$$h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p$$

Обучение модели (оценка параметров):

$$y \approx \hat{h}(x) = \hat{\theta}_0 + \hat{\theta}_1 x_1 + \hat{\theta}_2 x_2 + \cdots + \hat{\theta}_p x_p$$

В качестве потенциального недостатка такого подхода можно выделить то, что модель, которую мы выбираем, обычно не соответствует действительной неизвестной форме функции f.

Для решения обозначенной проблемы можно выбрать более гибкие модели, которые могут учесть много различных возможных функциональных форм для f. Однако в целом обучение более гибкой параметрической модели требует оценки большего количества параметров. В свою очередь более сложная модель может привести к переобучению.

Пример, линейная регрессия, логистическая регрессия, метод опорных векторов.

Непараметрические (non-parametric)

Непараметрические методы не делают явных предположений о форме f. Вместо этого они пытаются найти оценку f, которая будет максимально близкой к имеющимся наблюдениям, но в «сглаженном» (или в обобщенном) виде.

Данный подход имеет важное преимущество перед параметрическими методами: нет необходимости строить предположения о форме f, так как происходит подстройка формы к имеющимся данным в процессе обучения.

Недостаток подхода заключается в том, что в отличие от параметрических методов, которые сводят оценку к небольшому (относительно) набору параметров, данный подход требует большего (значительно больше, чем для параметрических) количества наблюдений для получения точных оценок для f.

Пример, k-ближайших соседей, деревья решений, многослойные нейронные сети

7. Предсказание и вывод

Существует две основные причины, по которым может понадобиться оценка неизвестной функции f:

- Предсказание (prediction)
- Вывод (inference)

Предсказание

Нам необходимо получить модель предсказания некоторого значения целевой переменной по известным значениям признаков (предикторов):

$$\hat{y} = \hat{f}(\boldsymbol{x}),$$

где $\hat{f}$ – оценка неизвестной функции f; $\hat{y}$ – предсказание для $\boldsymbol{x}$.

В данном случае мы рассматриваем $\hat{f}$ как черный ящик, в том смысле, что нам не так важно знать точную форму $\hat{f}$, главное, чтобы она давала приемлемую точность предсказания.

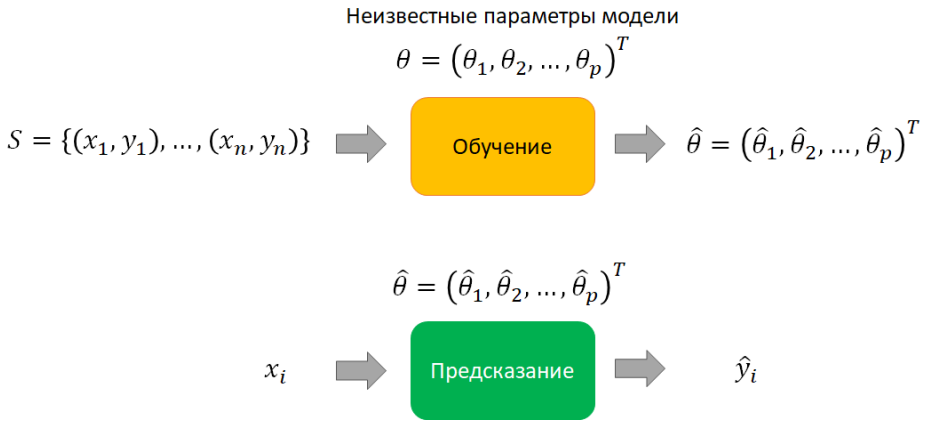


Рисунок – Обучение и предсказания

Вывод

Иногда нас больше интересует понимание того, как отклик y зависит от изменения каждого предиктора $\boldsymbol{x} = (x_1, \cdots, x_p)$, то есть как y меняется как функция от $x_1, \cdots, x_p$. В данном случае мы не может $\hat{f}$ рассматривать как черный ящик, так как нам необходимо знать её точную форму.

Типичные вопросы:

- Какие предикторы/признаки ассоциируются с откликом?
- Какой предиктор оказывает наибольшее влияние?
- Какой характер зависимости между предикторами и откликом (линейная или нет)?

Пример, разные виды рекламы и количество продаж

8. Точность и интерпретируемости

В каких случаях стоит использовать более ограниченную/жесткую модель вместо гибкой?

Если основная цель является вывод, то более ограниченные модели (с меньшим количеством параметров) являются более интерпретируемыми. Например, при использовании линейной

модели хорошо прослеживается взаимосвязь между целевым значением и каждым из предикторов. В случае с более гибкими методами можно прийти к очень сложной модели предсказания, при которой будет затруднительно оценить влияние каждого предиктора (признака) на отклик (предсказание).

Бывает, что более ограниченная модель дает лучшее предсказания из-за склонности гибких моделей к переобучению.

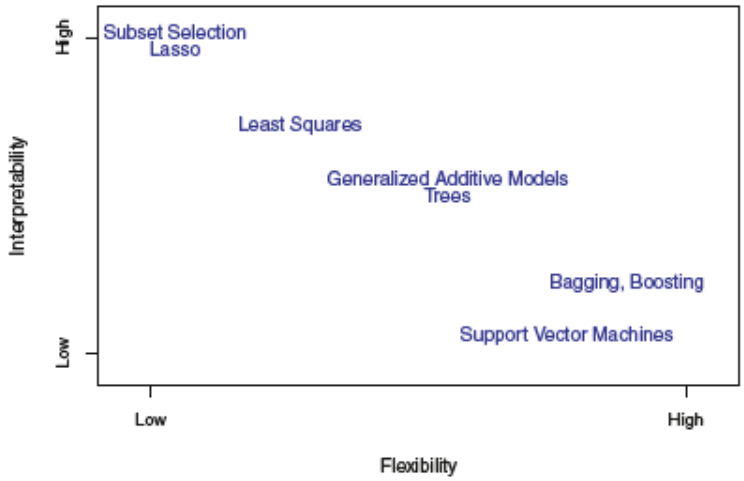
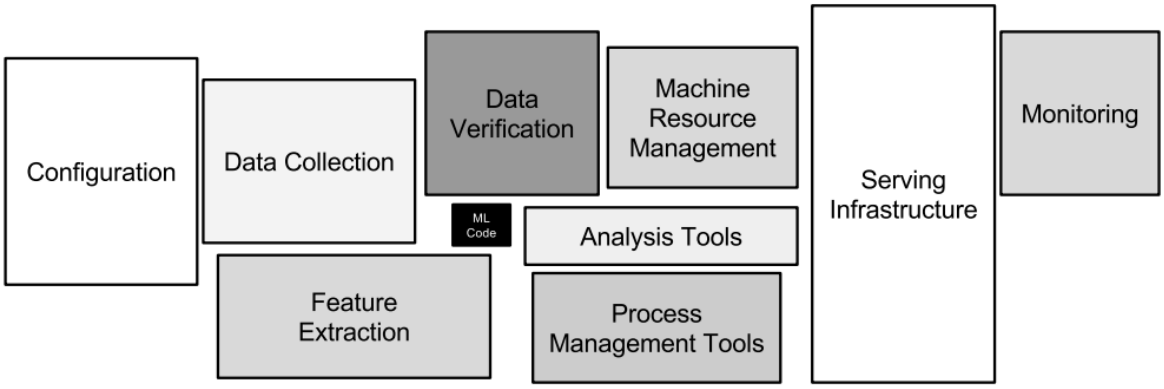
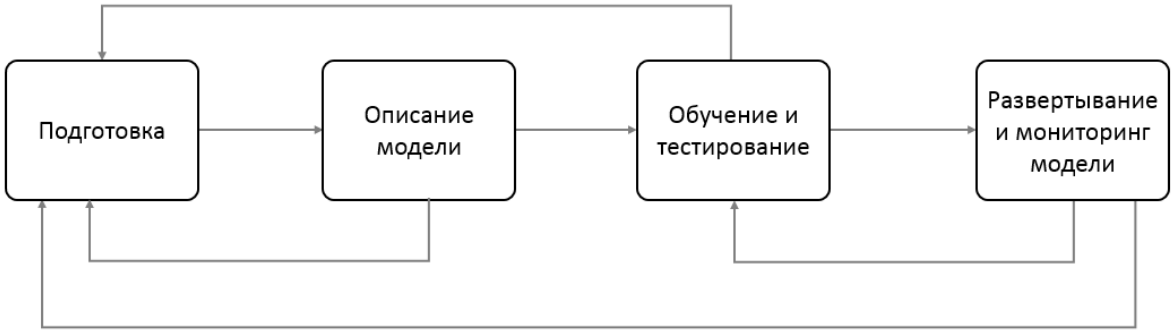


Рисунок – Гибкость и интерпретируемость [1]

9. Основные этапы построения модели



Решаемые задачи в системах машинного обучения [2]



Этапы проектирования модели

#	Этап
1	Подготовка
	• Описание проблемы: постановка цели проекта и определение метрик её достижения • Определение входных и выходных данных • Оценка взаимосвязи между входными и выходными данными • Сбор и разметка данных • Определение способа оценивания качества модели • Методы предобработки и очистки данных • Установка базовой отметки (baseline)
2	Описание модели
	• Выбор моделей, которые могут превзойти базовую отметку • Выбор подходящего методы регуляризации, чтобы избежать переобучения • Выбор функции потерь (loss function) и методов оптимизации
3	Обучение и тестирование модели
	• Запуск обучения выбранных моделей • Выбор значений гиперпараметров (количество итераций, коэффициента регуляризации) • Оценка качества моделей на отложенной выборке • Определение наилучшей модели • Анализ ошибок предсказания
4	Развертывание и мониторинг модели
	• Использование лучшей модели для предсказания • Мониторинг производительности модели • Сравнение выбранной модели с кандидатами из предыдущего этапа • Периодическая калибровка модели

Список литературы

1. Chapter 2. Statistical Learning // An Introduction to Statistical Learning by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshir. pp. 15–57. URL: http://faculty.marshall.usc.edu/gareth-james/ISL/
2. D. Sculley etc. Hidden Technical Debt in Machine Learning Systems