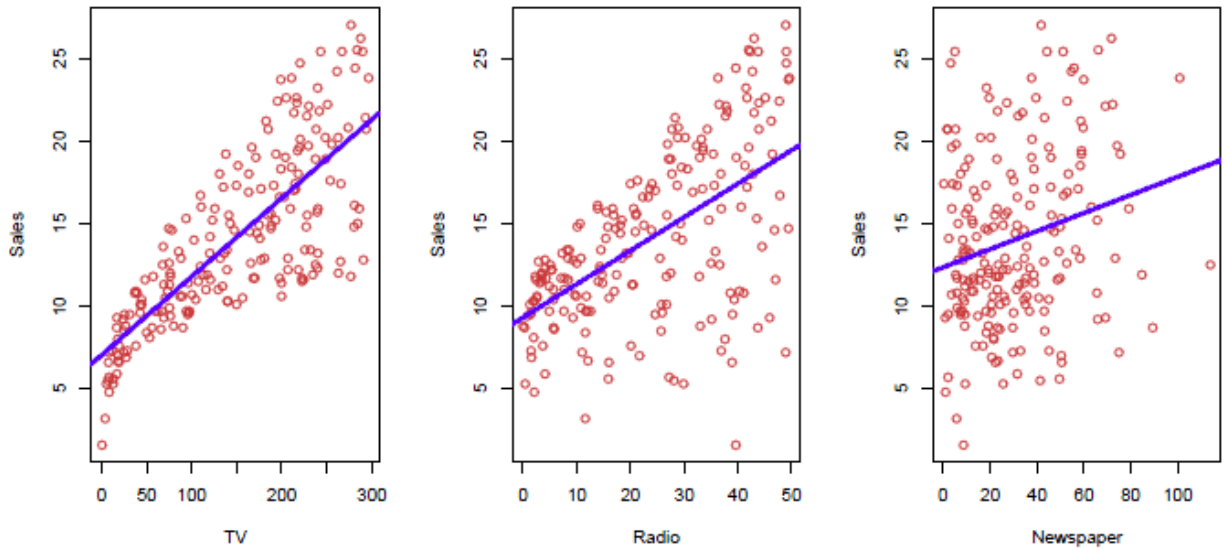
Лекция 5:
задача прогнозирования, проклятие размерности, переобучение

Демонстрационный пример: что такое прогнозирование?



- Показаны *зависимости* продаж от каналов рекламы: ТВ, радио и газет– синяя прямая линейной регрессии отдельно для каждого параметра.
- Можно ли спрогнозировать продажи на основе этого всего в совокупности?
- Возможно мы можем сделать лучший прогноз, используя модель

$$\text{Sales} \approx f(\text{TV}, \text{Radio}, \text{Newspaper})$$

Демонстрационный пример: обозначения

- Здесь Sales – это отклик или целевая переменная, которую мы хотим спрогнозировать. Обычно обозначаем отклик как *Y*.
- TV - это признак или вход; обозначим его как X_1. Признак Radio как X_2, и так далее.
- Тогда весь входной вектор можно обозначить как

$$X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

- Можно описать всю модель как $Y = f(X) + \epsilon$

- где ϵ отражает ошибки измерения и другие отклонения.

Задача «обучения с учителем»

- Множество «размеченных» примеров (прецедентов):
  - обучающая выборка или тренировочный набор:

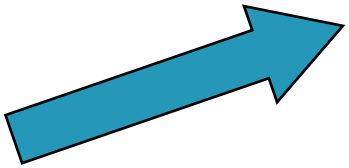
$$Z = \{(x_i, y_i)\}_{i=1}^{n} \in X \times Y$$

  - X: «сигнал», «объект», «ситуация»
  - Y: «отклик», «прогнозируемая величина»
- Неформальная постановка задачи:

$$f_Z : X \to Y$$

- Два этапа: обучение и прогнозирование

Обучение

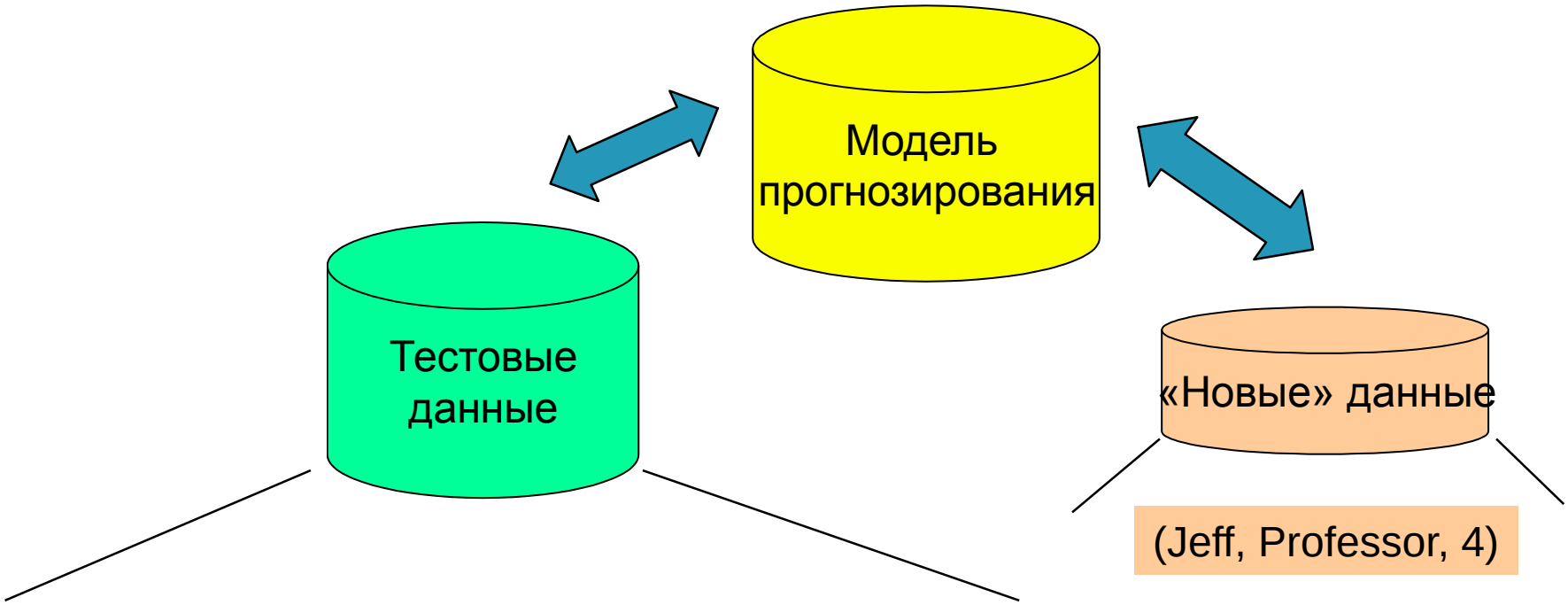


Метод машинного обучения

NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

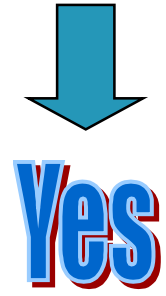
IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’

Оценка и прогнозирование



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

Tenured?



Типы задач прогнозирования

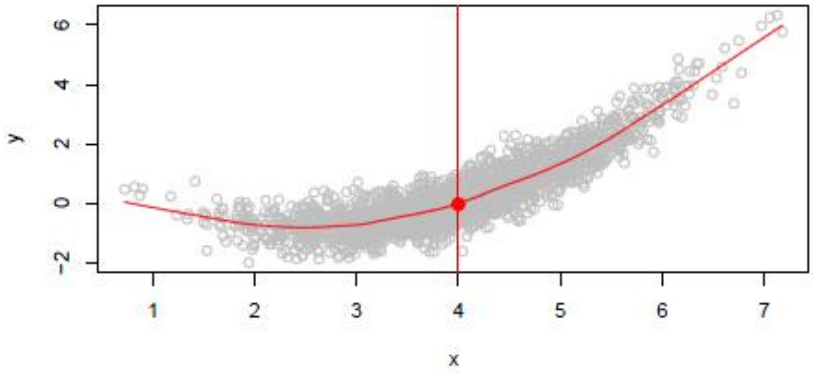
- Определяются типом допустимых значений «отклика» y_i и той оценкой качества, которая используется для выбора модели
- Бинарная классификация (разделение):
  - y_i - бинарная, f – бинарная функция
- Регрессия:
  - y_i - вещественное число, f – вещественная функция
- Классификация:
  - y_i - дискретная величина (метка класса), на Y нет порядка, f – дискретная функция
- Много-темная (multi-label) классификация:
  - y_i – множество неупорядоченных дискретных величин (меток классов), f – бинарная вектор-функция (i-й разряд – «да»/«нет» для i-го класса)
- Ранжирование:
  - y_i – множество упорядоченных дискретных величин (меток классов), f – вещественная вектор-функция (i-й разряд – ранг для i-го класса)

Применение методов прогнозирования в задачах ИАД

- Прогнозирование ради прогнозирования:
  - Автоматическая классификация и прогнозирование (обучились и применяем модель для решения прикладной задачи)
  - Выявление и описание основных зависимостей, т.е. как значения характеристик примера влияют на отклик (важны нтерпретируемость и визуализируемость модели)
- Предобработка данных:
  - «Условная» дискретизация (разбиение значений свойств примеров на интервалы с учетом отклика)
  - Обработка пропущенных значений (импутация)
- Поиск исключений и артефактов:
  - Что не соответствует прогнозу, то аномалия
  - Поиск и построение моделей «редких» (или малых) классов
- Области применения:
  - Везде, где необходим прогноз или классификация

Что означает, что f(X) дает «хороший» прогноз?

- Имея функцию *f* можно найти *Y* для новой точки *X* = *x*.
- Мы можем оценить, какая компонента X_j важна в объяснении *Y*
- В зависимости от сложности функции *f*, мы можем понять как каждый компонент вектора *X* влияет на *Y* .



- Существует ли идеальная *f(X)*? Какое «хорошее» значение *f(X)* для выбранного *X* (например, *X* = *4)*? Может быть много значений *Y* для *X*. Хорошее значение таково, что *f(4)* = *E(Y|X = 4)*
- *E(Y|X = 4) – ожидаемое значение* (среднее) из *Y* для заданного.
- *f(x)* = *E(Y|X = x)* называется функцией регрессии.

Функция регрессии f(x)

- Аналогично определяется для вектора *X*, например:

$$f(x) = f(x_1, x_2, x_3) = E(Y|X_1 = x_1, X_2 = x_2, X_3 = x_3)$$

- Является оптимальным предиктором *Y* относительно среднеквадратичной ошибки прогнозирования: *f(x) = E(Y|X = x)* это функция которая минимизирует

$$E[(Y - g(X))^2|X = x]$$

по всем функциям *g* во всех точках *X = x*.

- $\epsilon = Y - f(x)$ - это *несокращаемая* ошибка, обычно существует распределение возможных значений.
- Для любой оценки $\hat{f}(x)$ функции *f(x)*, мы имеем:

$$E[(Y - \hat{f}(X))^2|X = x] = \underbrace{[f(x) - \hat{f}(x)]^2}_{Reducible} + \underbrace{\mathrm{Var}(\epsilon)}_{Irreducible}$$

Другие функции потерь

- Функция потерь $L = Y \times Y \to R$ характеризует отличие правильного ответа от спрогнозированного
- Примеры:
  - Классификация и регрессия:

$$L(y, y') = [y \neq y'], L(y, y') = |y - y'|,$$

$$L(y, y') = (y - y')^2, L(y, y') = \left[|y - y'| > \delta\right]$$

  - Много-темная классификация:

$$HL = |y \nabla y'|, a \nabla b = (a \cup b) \setminus (a \cap b),\ a \subseteq Y\ , b \subseteq Y$$

  - Ранжирование:

$$RL = \frac{|\{(l,s) \in y \times \bar{y} : y_l \leq y_s\}|}{|y||\bar{y}|}$$

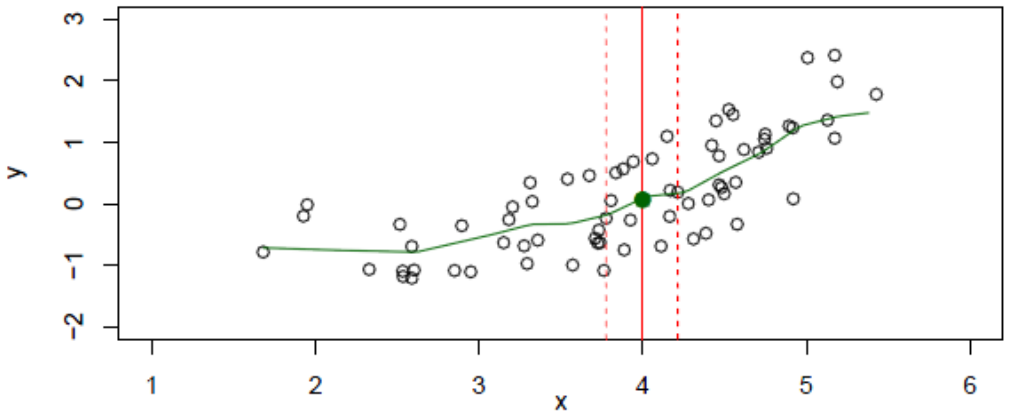
- Оценка качества прогноза:
  - Усреднение потерь по множеству примеров

Как оценить *f*?

- Обычно мы имеем немного точек с одинаковым *X*.
- Таким образом мы не можем вычислить *E(Y|X = x)!*
- Определим

$$\hat{f}(x) = \mathrm{Ave}(Y|X \in \mathcal{N}(x))$$

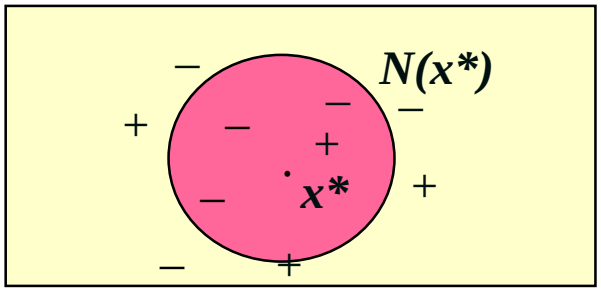
где *N(x)* – некоторая окрестность точки *x*.

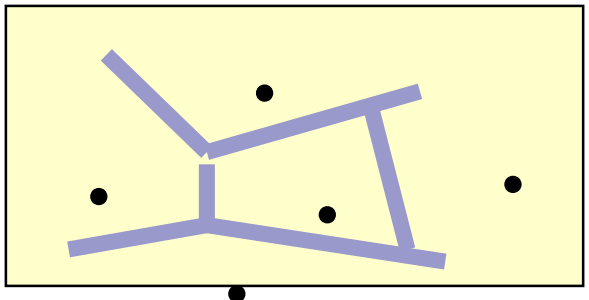


- Усреднение по ближайшим соседям может быть достаточно хорошо для малых *p* (число признаков) и больших *N* (число наблюдений).
- *Методы ближайших соседей* могут плохо работать при больших *p*.

Метод K ближайших соседей

- Общая схема работы:
  - Каждый пример – точка в пространстве, все примеры хранятся
  - Вводится метрика расстояния с учетом нормирования координат
  - Ищется K (от 1 до …) ближайших соседей
  - Прогноз вычисляется как функция от откликов найденных соседей по одному из алгоритмов: $y^* = F_{x_i \in N(x^*)}(y_i)$
- Метод KNN:
  - Для задачи регрессии отклик считается как среднее по откликам всех соседей: $y^* = \frac{1}{K} \sum_{x_i \in N(x^*)} y_i$
  - Для классификации выбирается самый частый класс: $y^* = \arg\max_{c \in C, x_i \in N(x^*)} [|y_i = c|]$





Метод «взвешенных» K ближайших соседей

- Метод KWNN:
  - На базе KNN, но помимо распределения «отклика» учитываются и расстояния до соседей в окрестности
  - Учет происходит за счет «взвешенного» голосования для классификации:

$$y^* = \arg\max_{c \in C, x_i \in N(x^*)} \left[\frac{w_i \left| y_i = c \right|}{\sum_{x_j \in N(x^*)} w_i} \right]$$

  - И «взвешенного» среднего для регрессии

$$y^* = \frac{\sum_{x_i \in N(x^*)} w_i y_i}{\sum_{x_i \in N(x^*)} w_i}$$

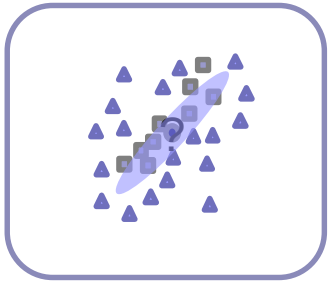
  - весовой коэффициент обратно пропорционален квадрату расстояния или пропорционален корреляции с откликом

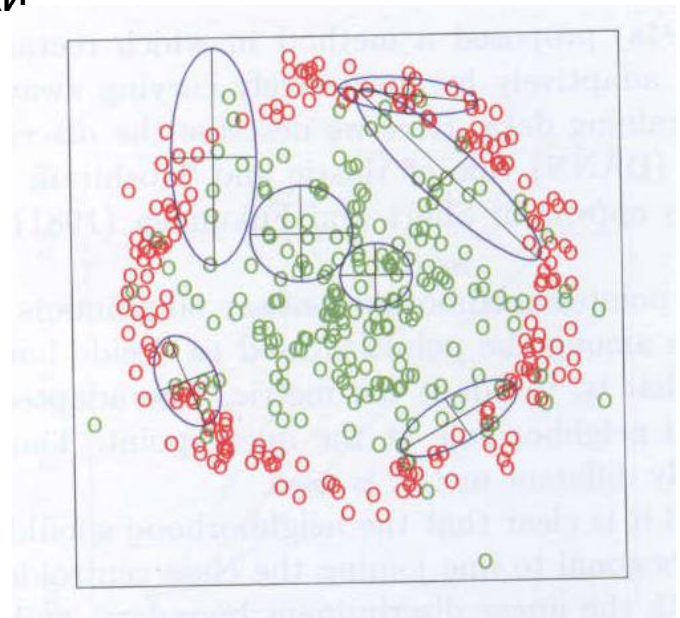
Метод K ближайших соседей с адаптивным расстоянием

- Метод DANN (в майнере не поддерживаетя):
  - На базе KNN, но используется локальный дискриминантный анализ для адаптации метрики расстояния с учетом структуры распределения соседей в окрестности:

$$d^{(l)}(x^*, x_i) = (x^* - x_i)^T \sum^{(l)} (x^* - x_i)$$

- Параметры алгоритма:
  - K_M – число соседей для оценки метрики (нужно побольше)
  - K – число соседей для прогноза (лучше поменьше)
  - ε – «смягчающий» параметр





Метод K ближайших соседей с адаптивным расстоянием

- Процедура:
  1. Инициализация метрики единичной матрицей ∑ = I
  2. Поиск K ближайших соседей вокруг x* в метрике ∑.
  3. Расчет W - взвешенной суммы внутриклассовых ковариационных матриц:

$$W = \sum_{c \in C} \sum_{x_k \in N(x^*), y_k = c} \pi_k (x_k - \bar{x}_c)(x_k - \bar{x}_c)^T$$

  4. … и B - взвешенной суммы межклассовых ковариационных матриц:

$$B = \sum_{K} \pi_k (x_k - \bar{x})(x_k - \bar{x})^T$$

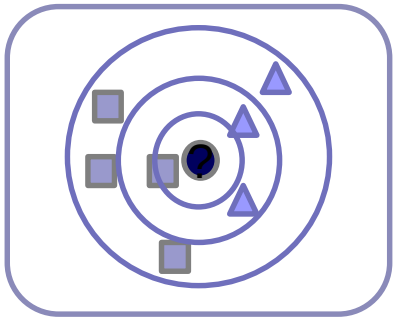
  5. Новая метрика:

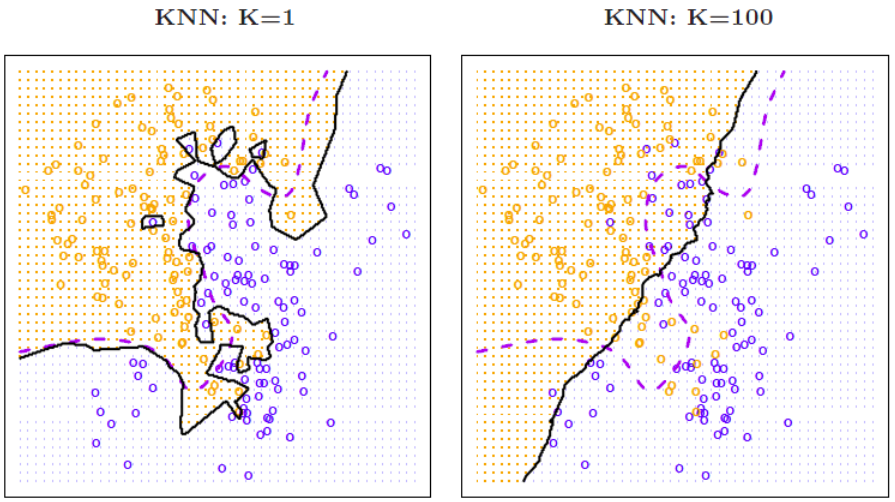
$$\sum^{(l+1)} = W^{-1/2}[W^{-1/2} B W^{-1/2} + \varepsilon I]W^{-1/2}$$

  6. Повторить шаги 2-5 заданное число раз
  7. Применить стандартный KNN

Выбор параметра K

- Важность K:
- *k* = 1: Результат = квадрат
- *k* = 5: Результат = треугольник
- *k* = 7: Снова квадрат





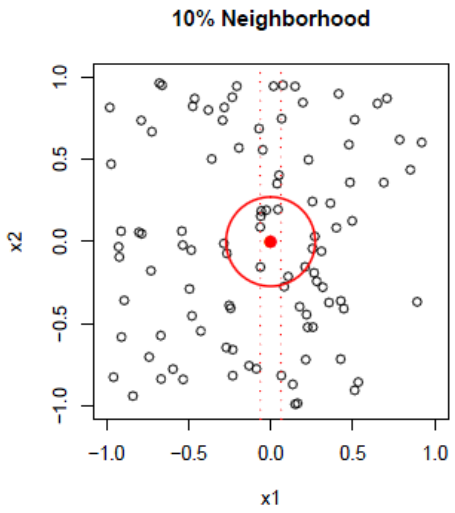
- Выбор *k*:
  - Если *k* мал, то чувствительность к шуму, и негладкие границы классов
  - Если *k* велико, то окрестность может сильно «задеть» соседний класс, зато гладкие границы При классификации надо использовать нечетный *k*, чтобы не было «ничьей»
  - Выбирается кросс-валиадцией или на валидационном наборе
  - Стандартная эвристика k=sqrt(n)

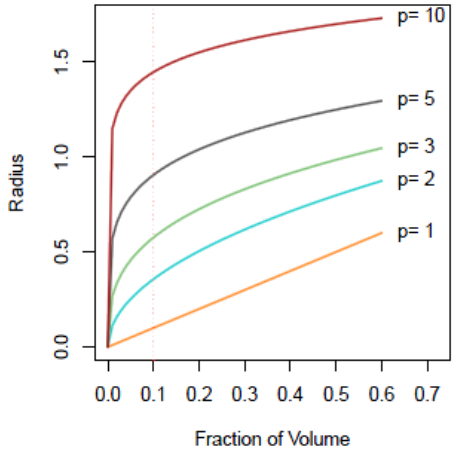
Свойства методов KNN

- Основные свойства:
  - «Ленивый классификатор» - не надо ничего обучать
  - Качество классификации зависит, в основном, от структуры данных, от параметров в меньшей степени
  - Обязательно нужна хорошая метрика и нормированные атрибуты
- Достоинства:
  - Простой и легко реализуемый
  - Один из самых точных
  - Легко адаптируется под сложные типы «откликов», включая ранжирование, многотемность и т.д.
  - Можно интегрировать экспертные знания, задавая веса у примеров, или параметры у метрики
- Недостатки:
  - «черный ящик» - результат не интерпретируемый совсем
  - Достаточно вычислительно трудоемкий, проблема использования индексов для сложных структур X
  - **«Проклятие размерности»**

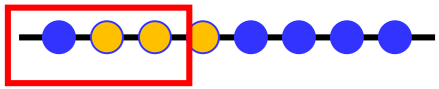
Проклятие размерности

- Ближайшие соседи как правило расположены далеко при больших размерностях.
  - Нам нужно получить значительную часть из N значений y_i , чтобы снизить дисперсию - например, 10%.
  - 10% соседей для случая больших размерностей не может быть локализована, так что мы уже можем сделать оценку $E(Y|X = x)$ на основе локального усреднения.

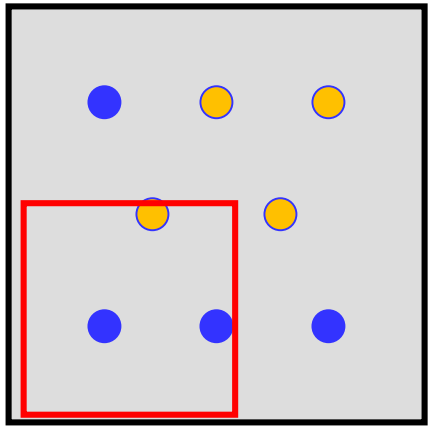




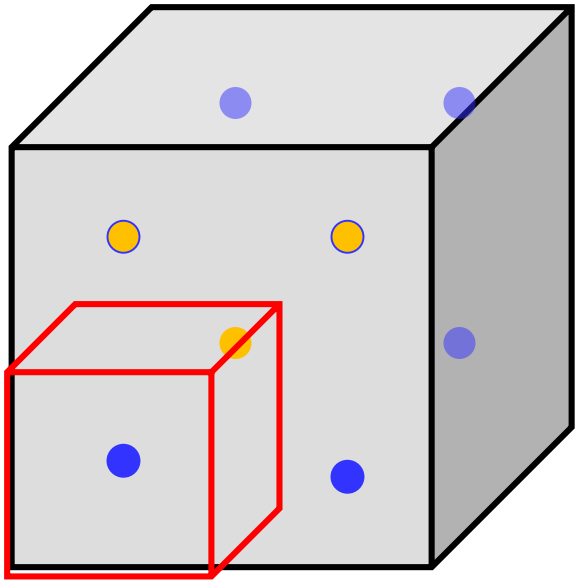
Модельный пример, демонстрирующий проклятие размерности



1–D



2–D



3–D

- $r=K/N$
- $E_p(r)=r^{1/p}$
- $E_{10}(0.01)=0.63$
- $E_{10}(0.1)=0.8$

Параметрические модели

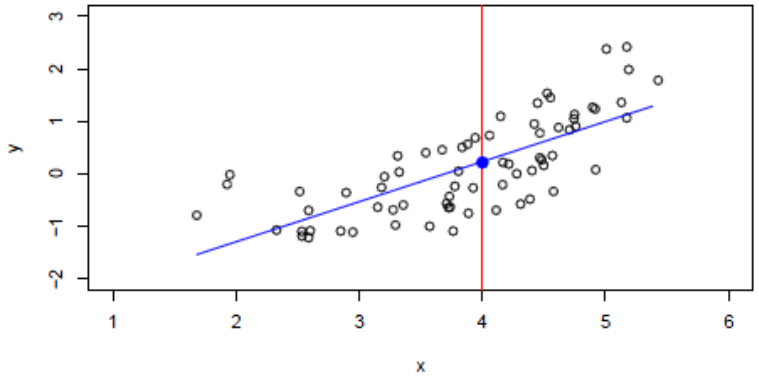
Линейная модель представляет собой важный пример параметрической модели:

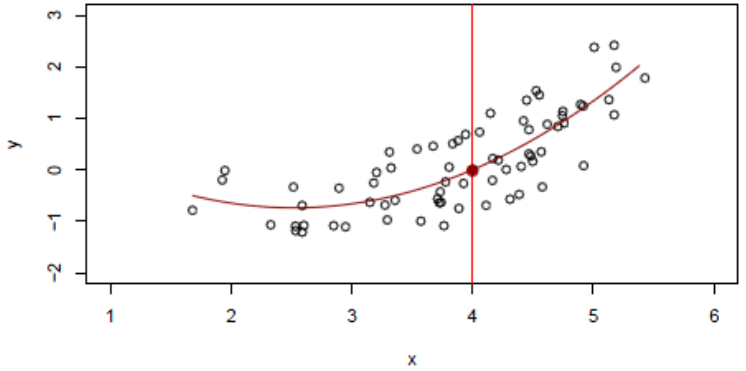
$$f_L(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots \beta_p X_p.$$

- Линейная модель определяется *p + 1* параметрами

$$\beta_0, \beta_1, \ldots, \beta_p.$$

- Мы оцениваем параметры на основе подгонки модели на обучающем наборе данных.
- Хотя такие модели *почти никогда* не показывают очень хорошую точность, но служат хорошей и интерпретируемой аппроксимацией неизвестной истинной функции *f(X)*.





Линейная регрессия

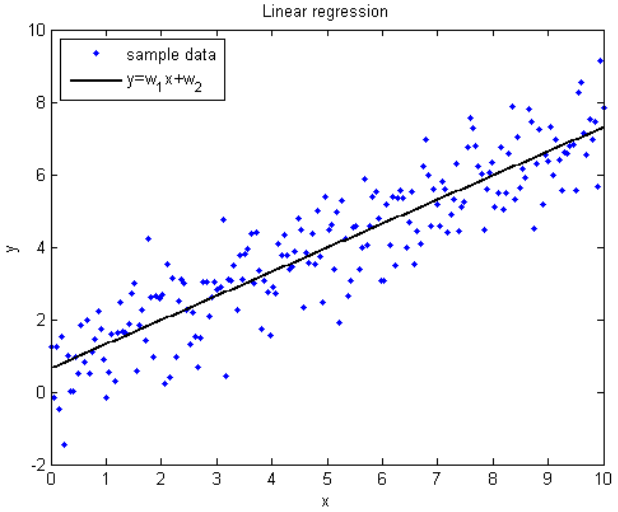
- Задача регрессии:

$$y(x_1,...,x_p) = E(Y \mid X_1 = x_1,..., X_p = x_p)$$

- Уравнение линейной регрессии:

$$f(X) = b_0 + \sum_{j=1}^{p} X_j b_j + \varepsilon$$

  - ε = N(0,σ²) - шум
  - Y –отклик
  - X=(X_1,…, X_p) - регрессоры (предикторы)
  - b – параметры модели



- Линеаризируемые регрессии:
  - Степенная ,Экспоненциальная
  - Гиперболическая, и другие

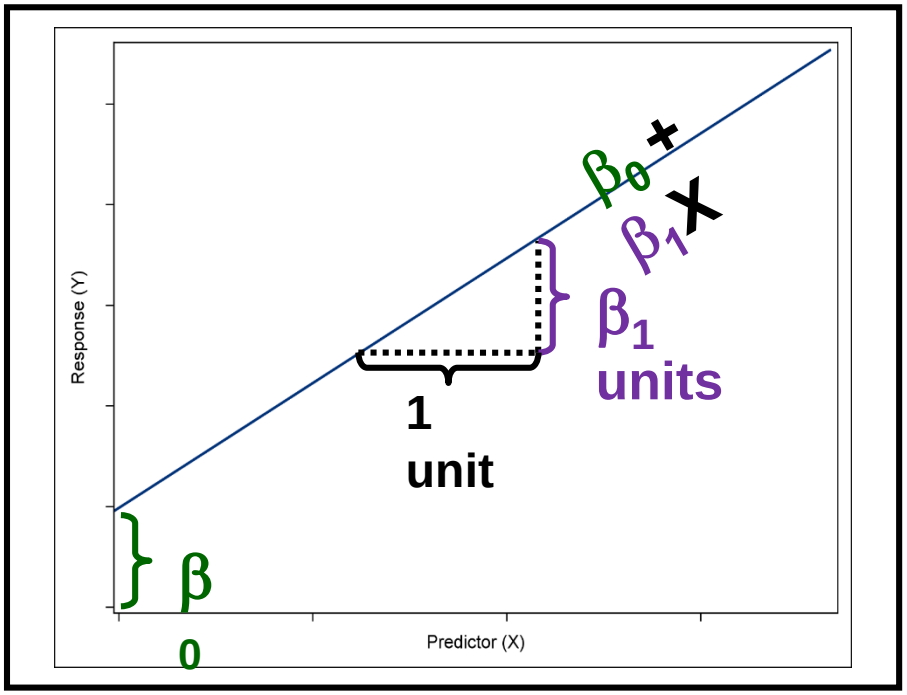
$$y = a x_1^{b_1} x_2^{b_2} ... x_p^{b_p} \varepsilon,$$

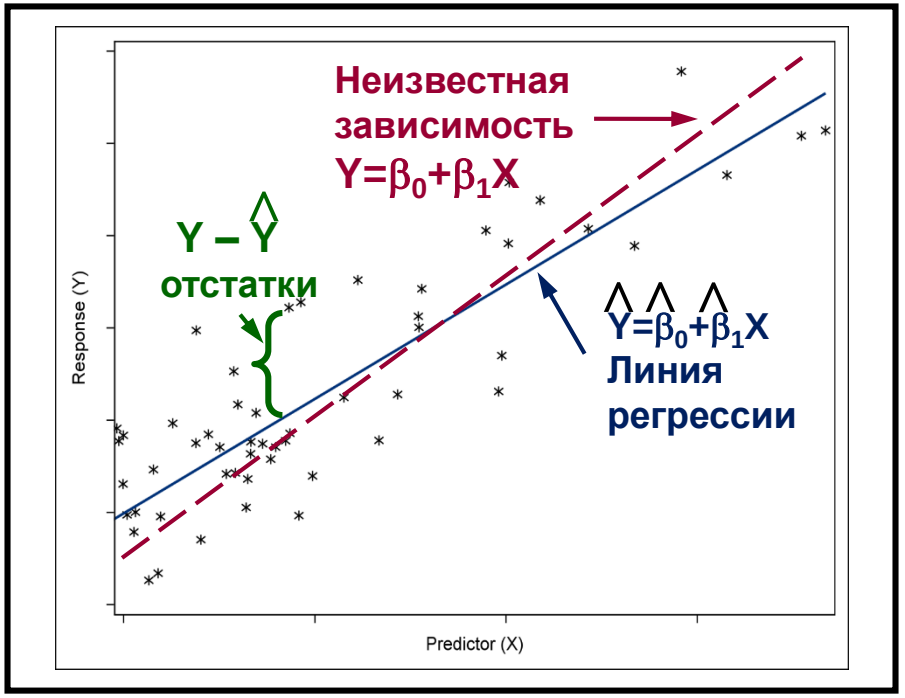
$$y = e^{a+b_1x_1+b_2x_2+...+b_px_p+\varepsilon},$$

$$y = (a + b_1x_1 + b_2x_2 + ...b_px_p + \varepsilon)^{-1}$$

- Цель регрессионного анализа:
  - Определение наличия связи между переменными и характера этой связи (подбор уравнения)
  - Предсказание значения зависимой переменной с помощью независимой(-ых)
  - Определение вклада отдельных независимых переменных в вариацию зависимой

Простая линейная регрессия

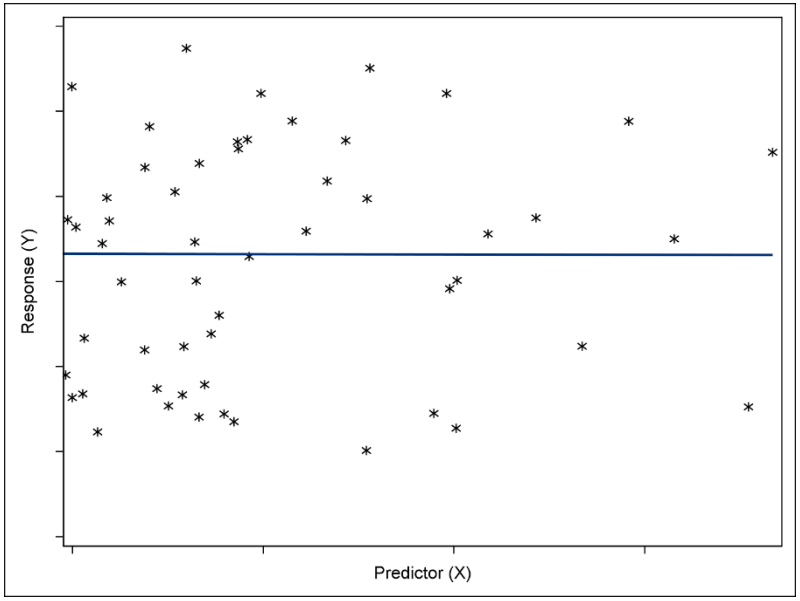


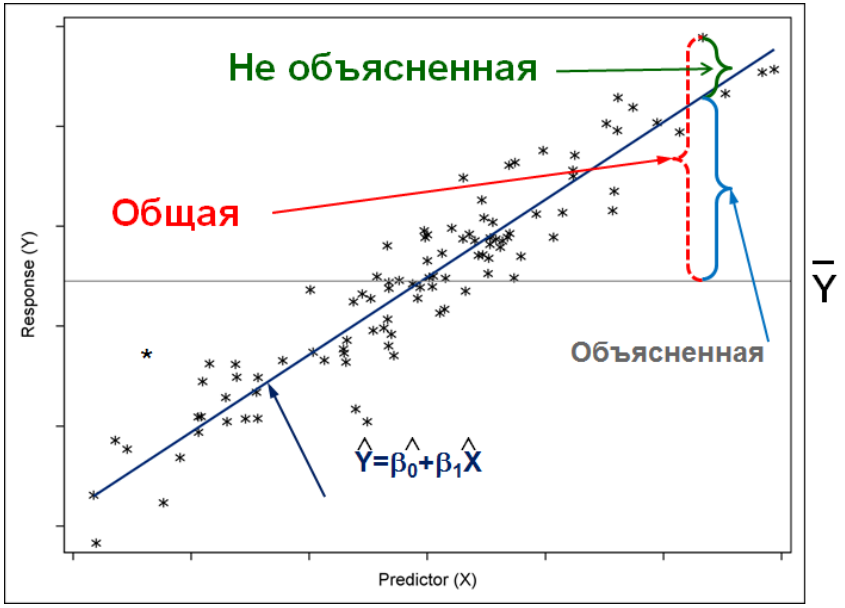


Предположения:

- Независимость наблюдений
- Выбранное уравнение регрессии (например, линейное) соответсвует истинной зависимости в данных
- Нормальность ошибки (с константной дисперсией по всем наблюдениям)

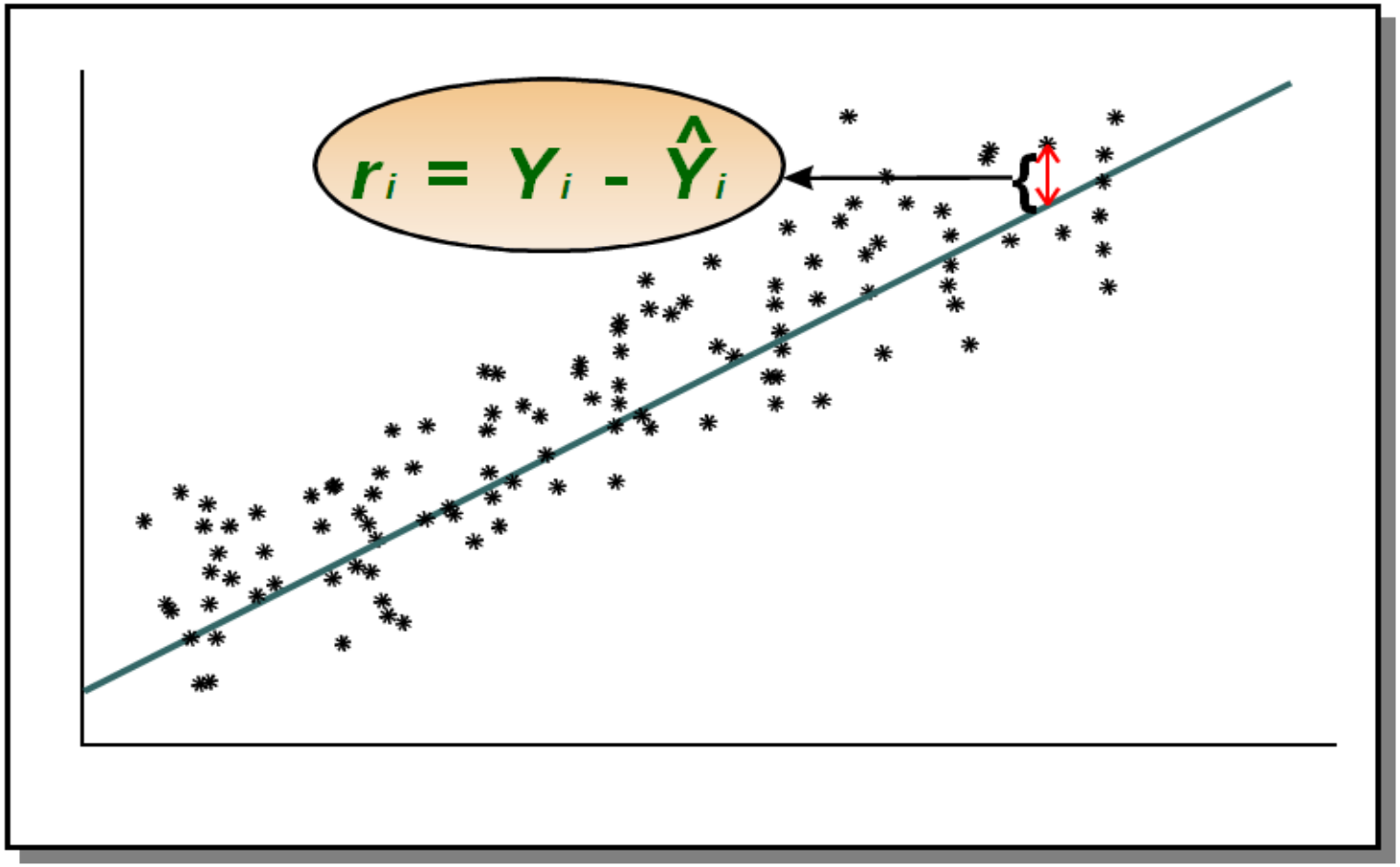
Базовая модель (Нулевая гипотеза)





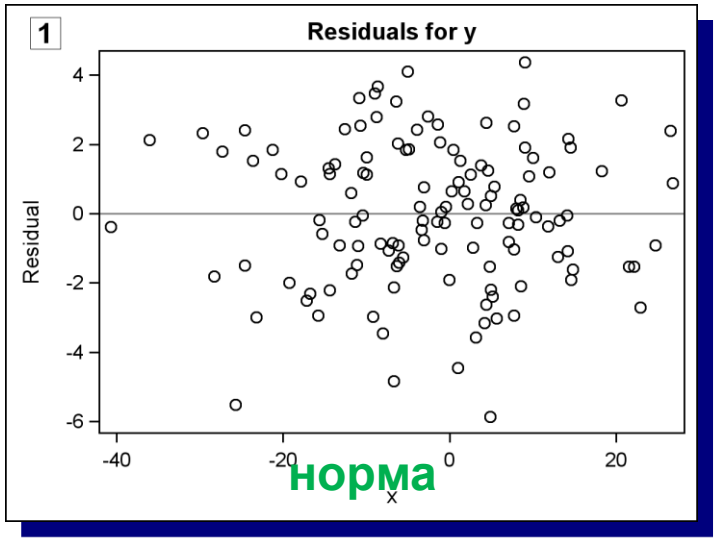
- Нулевая гипотеза:
  - Регрессионная модель приближает наблюдаемые данные не лучше базовой модели – константы ($\beta_1=0$)
- Альтернативная гипотеза:
  - Регрессионная модель лучше приближает наблюдаемые данные чем базовая модель – константа ($\beta_1\neq0$)

Проверка предположений модели с помощью графиков остатков

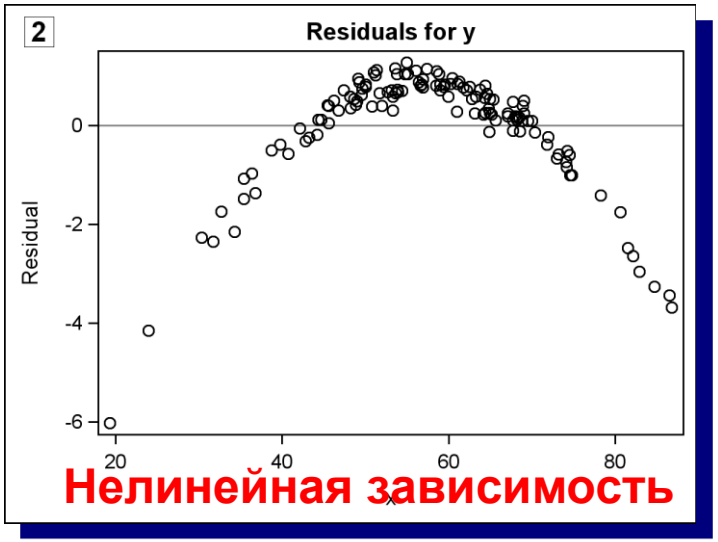


Графики: как остатки зависят от прогноза, от отклика, от предикторов

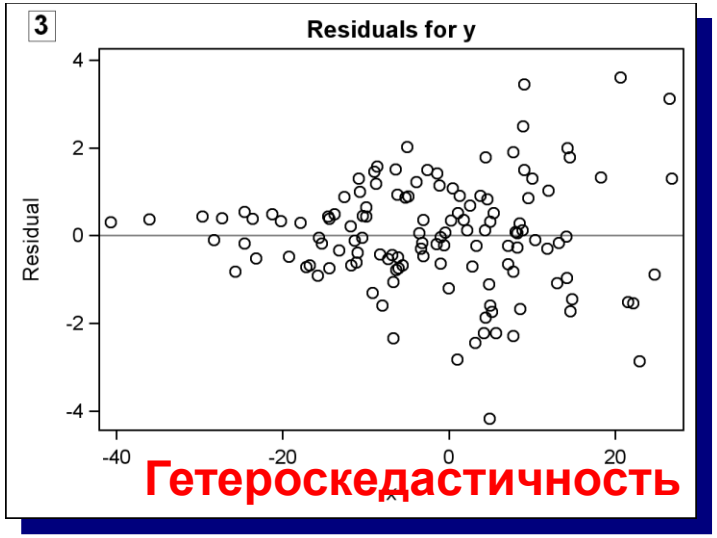
Графики остатков



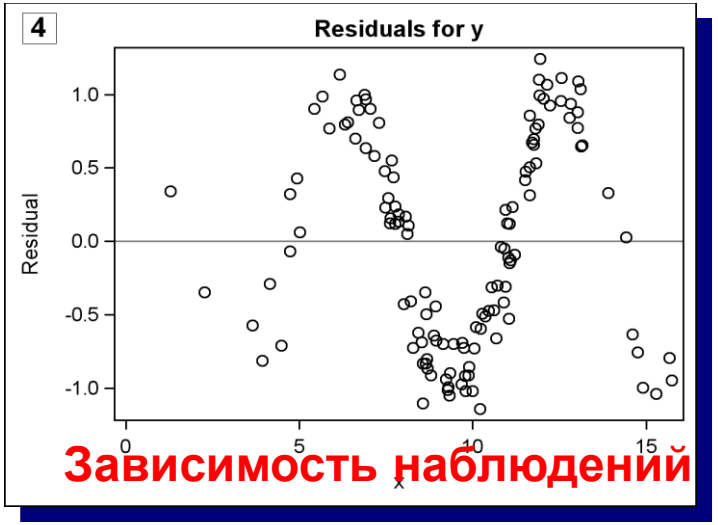
норма



Нелинейная зависимость



Гетероскедастичность



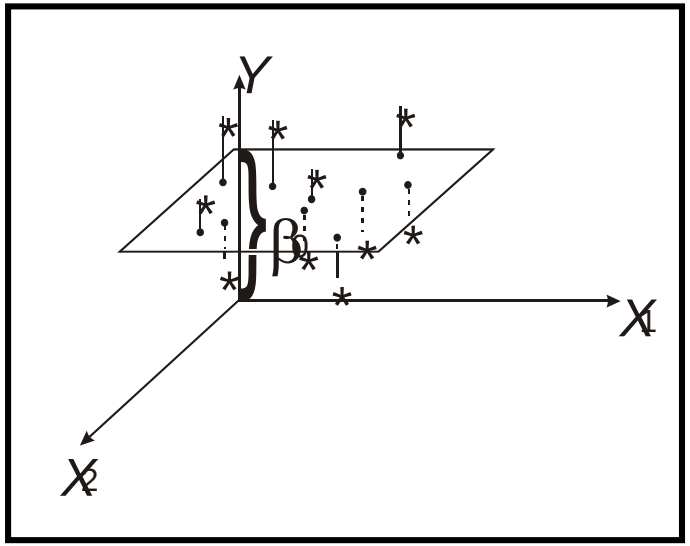
Зависимость наблюдений

Множественная линейная регрессия

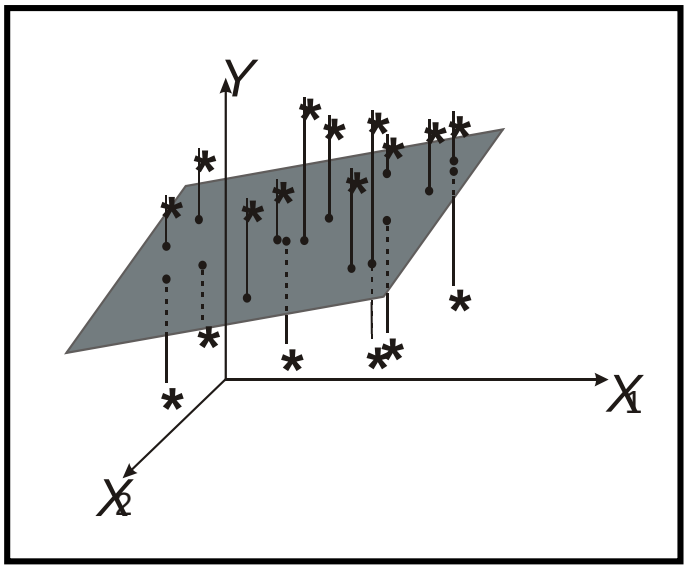
- Пример линейной модели с двумя переменными

$$Y=\beta_0+\beta_1X_1+\beta_2X_2+\varepsilon, \text{ где}$$

Y – отклик, X_1 и X_2 предикторы, ε - ошибка, β_0, β_1, и β_2-параметры (неизвестные)



Нет зависимости

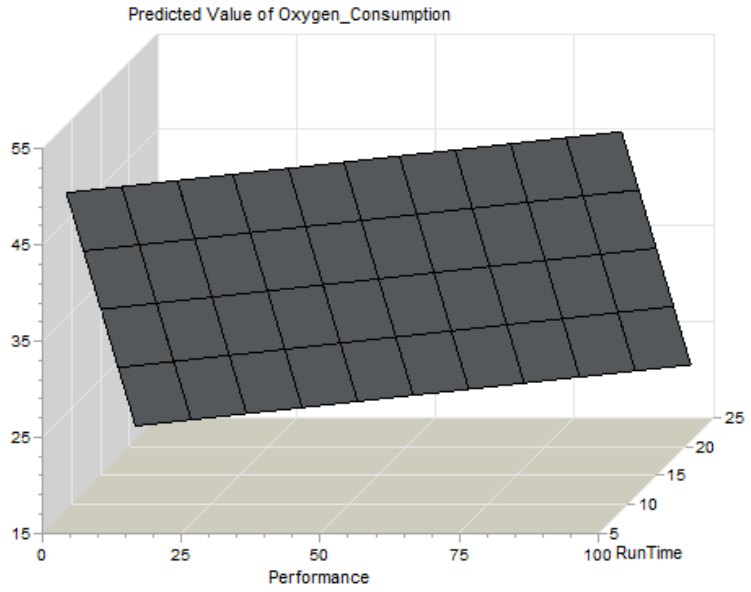


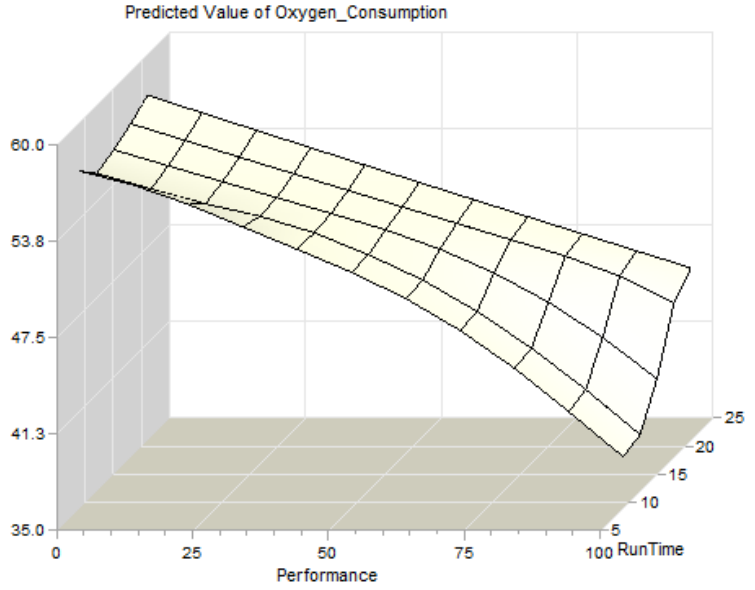
Есть зависимость

Множественная линейная регрессия

- В общем случае ищем зависимость как линейную комбинацию *k* предикторов X_1 - X_k:

$$Y=\beta_0+\beta_1X_1+...+\beta_kX_k+\varepsilon$$





$$Y=\beta_0+\beta_1X_1+\beta_2X_2+\varepsilon$$

Линейная модель с линейными эффектами

$$Y=\beta_0+\beta_1X_1+\beta_2X_1^2+\beta_3X_2+\beta_4X_2^2+\varepsilon$$

Линейная модель с нелинейными эффектами

Метод наименьших квадратов и проблема мультиколлинеарности

- Оценка ошибки = сумма регрессионных остатков (квадратичная функция потерь):

$$RSS(B) = \sum_{i=1}^{N} (y_i - f(\bar{x}_i))^2 = \sum_{i=1}^{N} (y_i - b_0 - \sum_{j=1}^{p} x_{ij} b_j)^2$$

- В матричной форме:

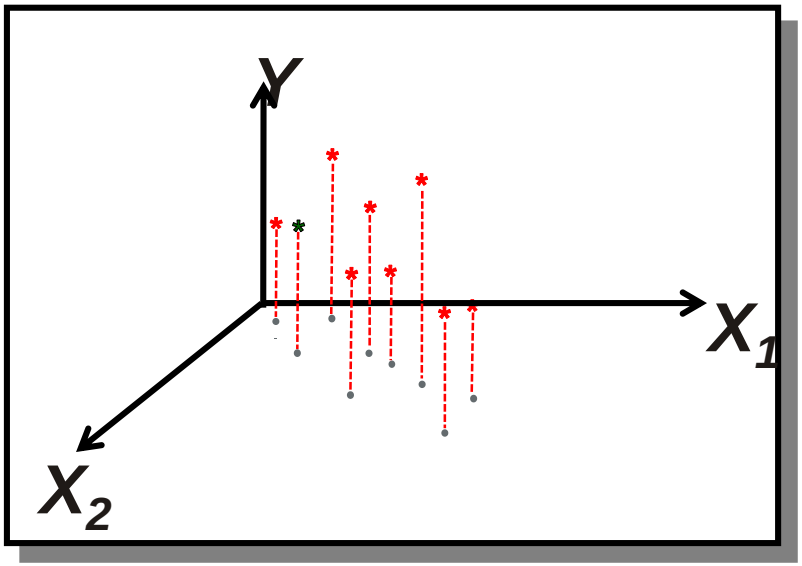
$$RSS(B) = (\bar{y} - XB)^T (\bar{y} - XB)$$

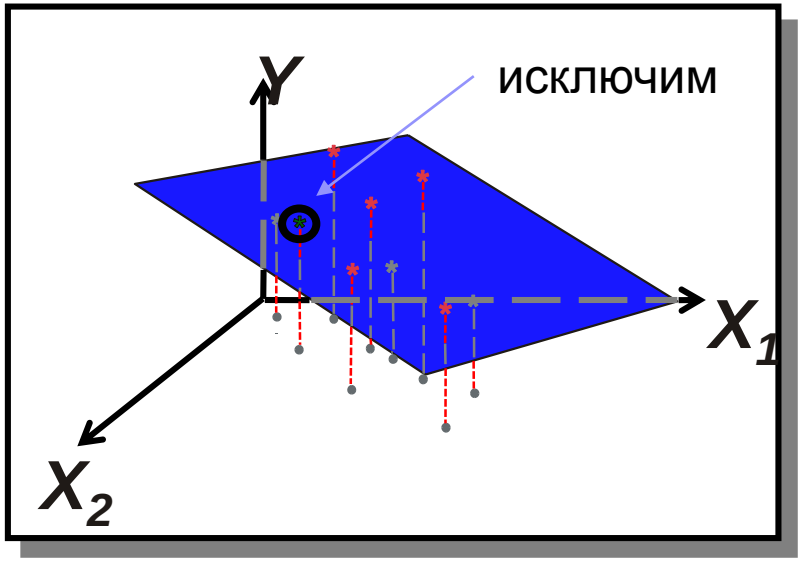
- Единственное оптимальное решение (если матрица данных не сингулярная)

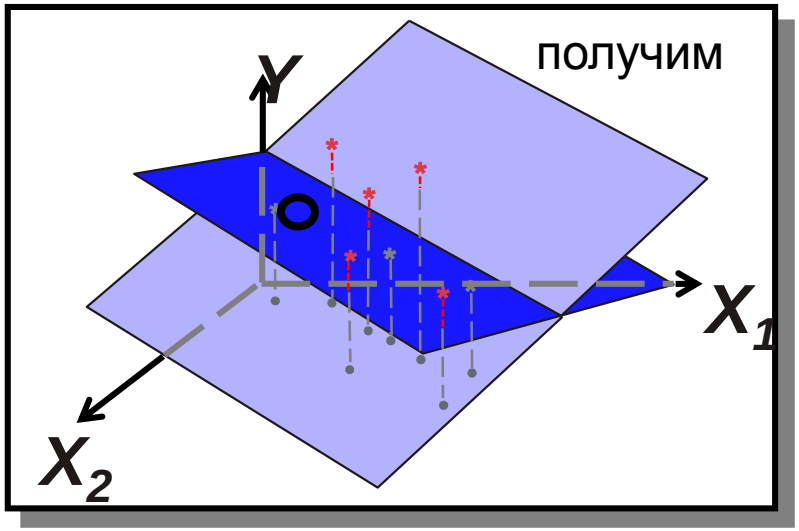
$$B = (X^T X)^{-1} X^T \bar{y}$$

- Недостатки:
  - Сингулярная матрица данных из-за коррелированных факторов
  - Большое число регрессоров – плохая точность и интерпретируемость
- Основные подходы:
  - Поиск и удаление зависимых и незначимых факторов
  - Использование «смещенных» регуляризированных моделей
  - переход к новым независимым факторам, например, с помощью метода главных компонент

Иллюстрация мультиколлинеарности







- Портятся статистики с оценкой значимости переменных
- Увеличивается вариативность оценки параметров и как следствие ошибка
- Есть тенденция к неограниченному росту коэф.

Множественная линейная регрессия

Предположения множественной линейной регрессии:

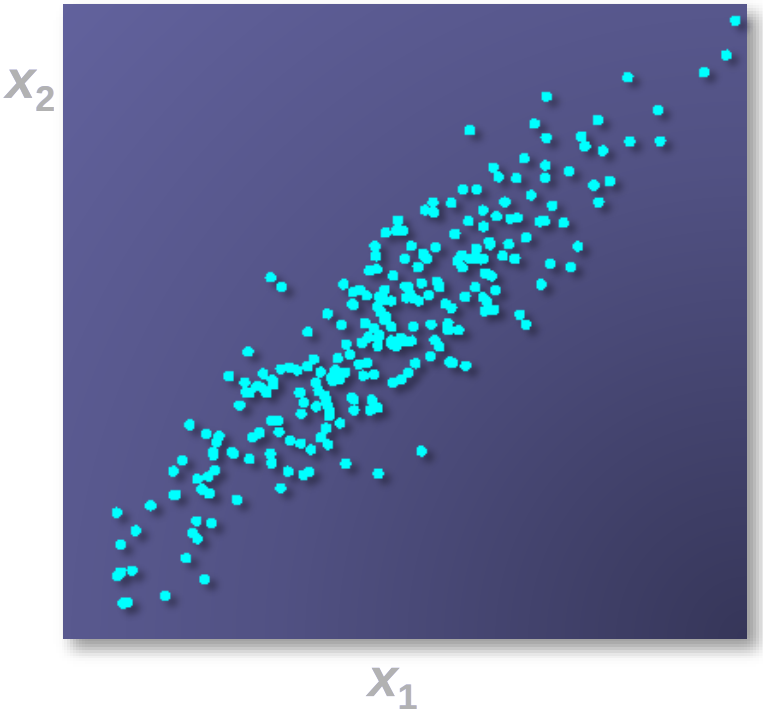
- Зависимость условного мат. ожидания отклика от предикторов - линейная
- Ошибка ε из $N(0, \sigma^2)$ с константной дисперсией.
- Ошибки независимы

Применяется для:

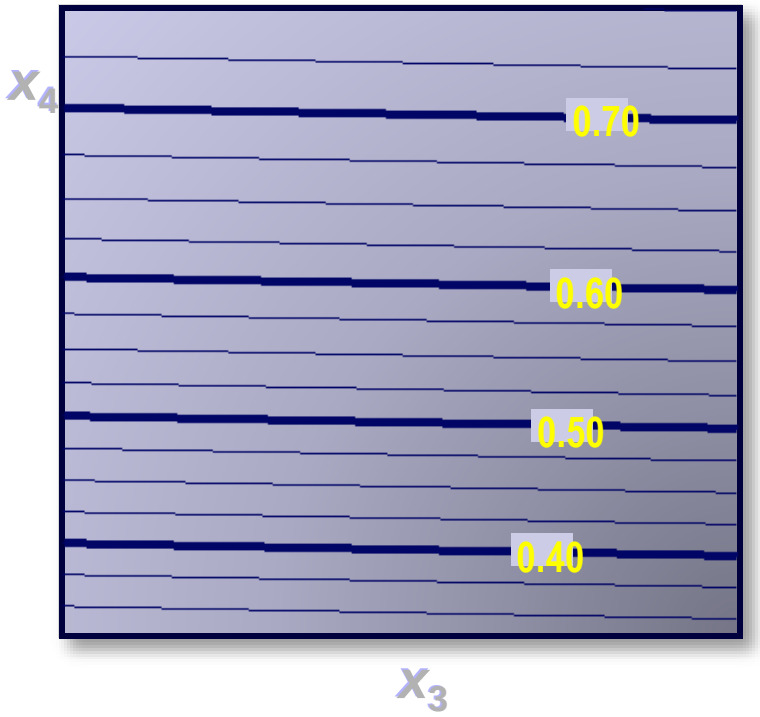
- Прогнозирования – важна не интерпретируемсть модели, значимость коэф. и т.д., а точность на тестовом наборе
- Разведочный анализ – важны значения и знаки коэф., уровни значимости и доверительные интервалы, цель – выявить интерпретируемые зависимости в данных

Проблемы входных переменных и для KNN и для МНК

Зависимость



Не релевантность отклику



Выхода два: либо преобразование либо исключение

...

Сокращение размерности в SAS EM

Дано: входные переменные {x1,…,xn} и выходная (числовая или бинарная) y

Задача: оставить только значимые и независимые xi

Работает в два этапа:

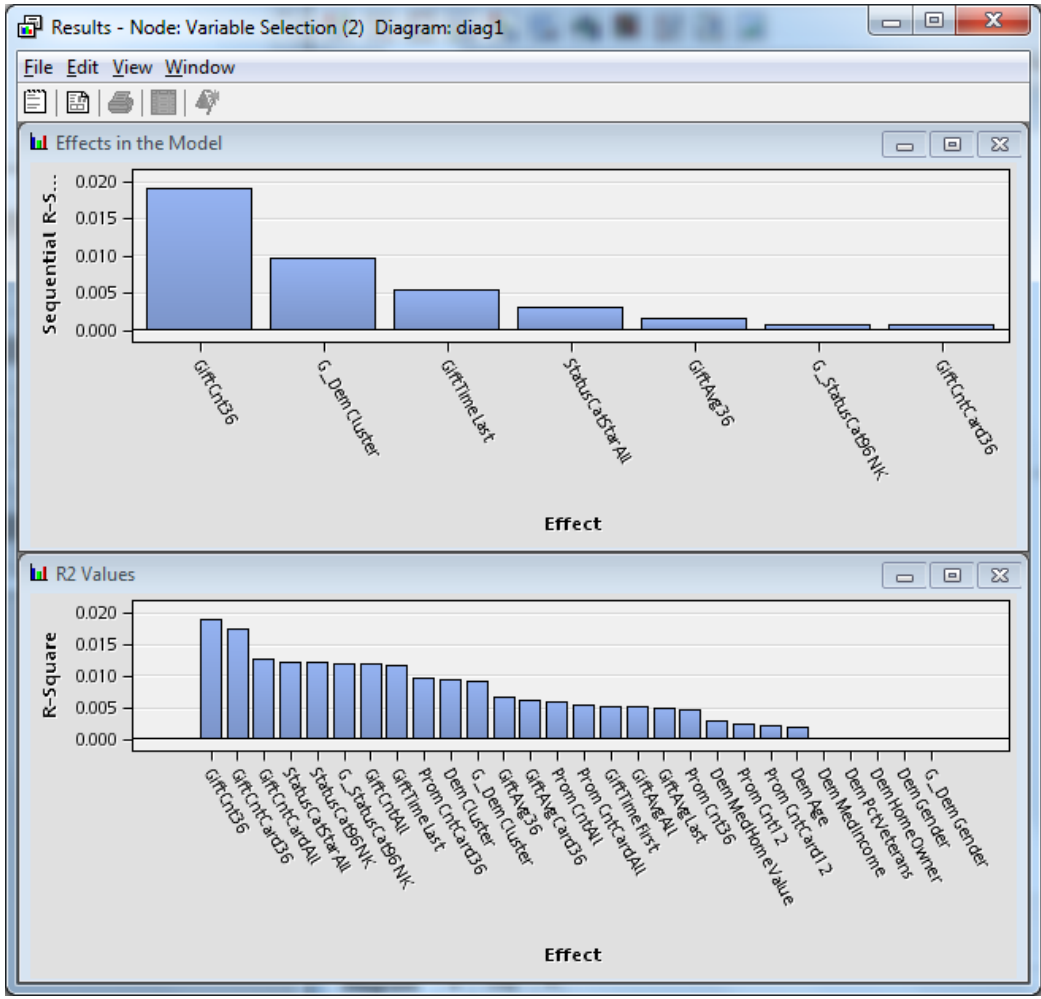
1. Уделяет все x_i, где $R^2(x_i)<T1$ *удаление незначимых*
2. Forward stepwise регрессия $f(x_{i1},\ldots x_{ik})$ пока

$R^2(f(x_{i1},\ldots xi_{ik}))-R^2(f(x_{i1},\ldots x_{ik-1}))>T2$

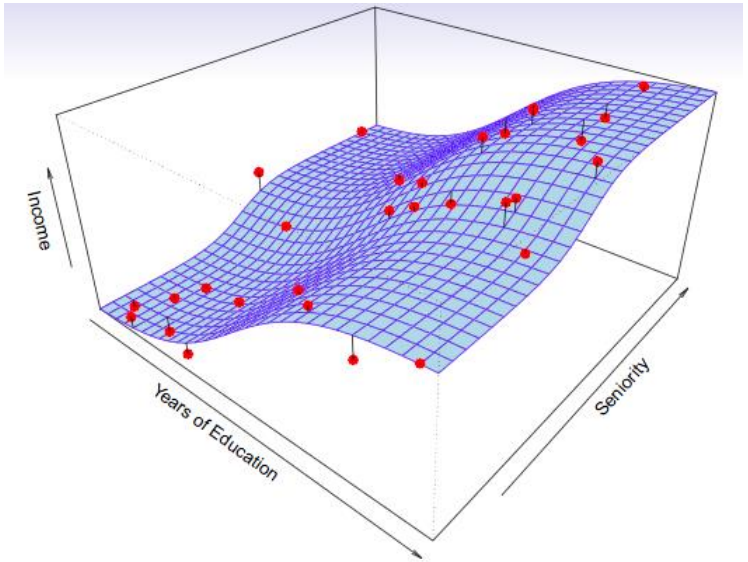
удаление зависимых

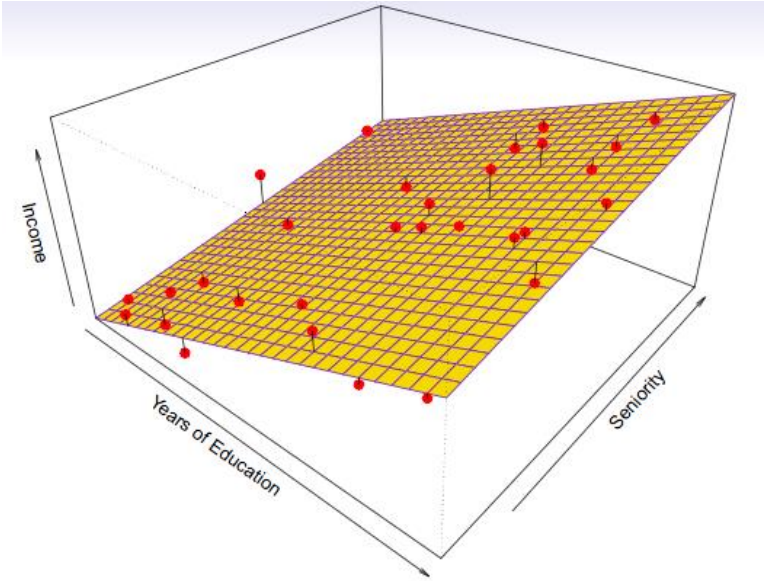
Преобразования переменных:

- Дискретизация непрерывных
- Группировка категориальных



Проблема недообучения и переобучения





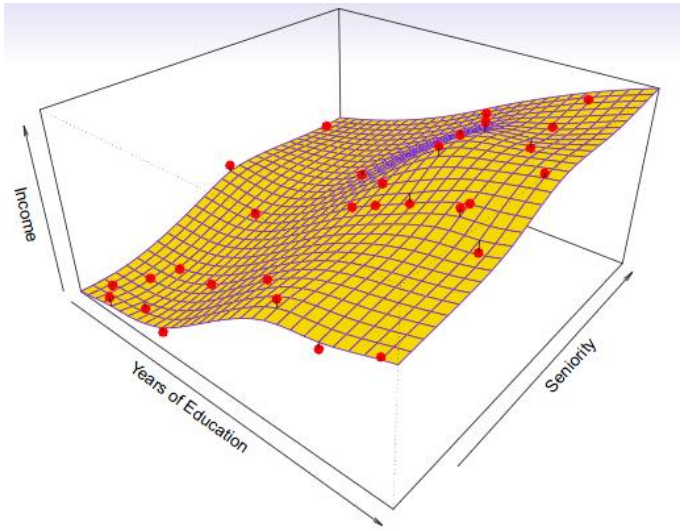
Модельный пример.

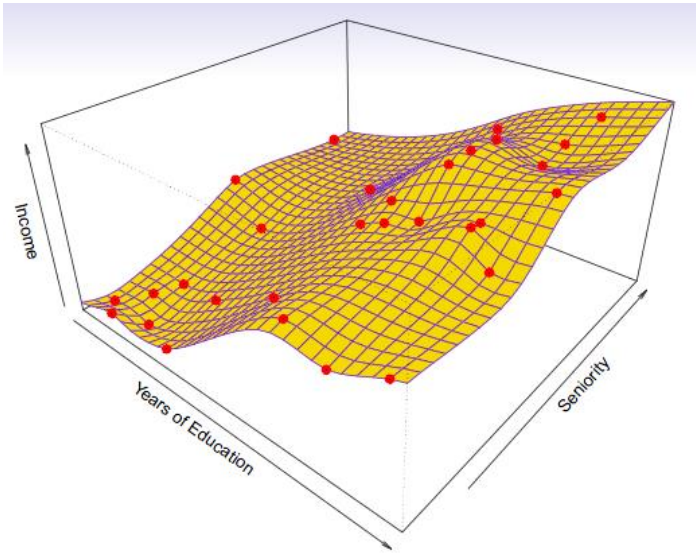
- Красные точки - наблюдения, синяя поверхность – истинная зависимость $\texttt{income} = f(\texttt{education}, \texttt{seniority}) + \epsilon$
- Желтая поверхность линейная модель

$$\hat{f}_L(\texttt{education}, \texttt{seniority}) = \hat{\beta}_0 + \hat{\beta}_1 \times \texttt{education} + \hat{\beta}_2 \times \texttt{seniority}$$

- Плохая точность приближения

Проблема недообучения и переобучения





Модельный пример.

- Более сложные модели (сплайны или полиномиальные регрессии или нейронные сети или еще что-то
- Справа модель не допускает ошибок на обучающем наборе.
- Это хорошо? Нет!

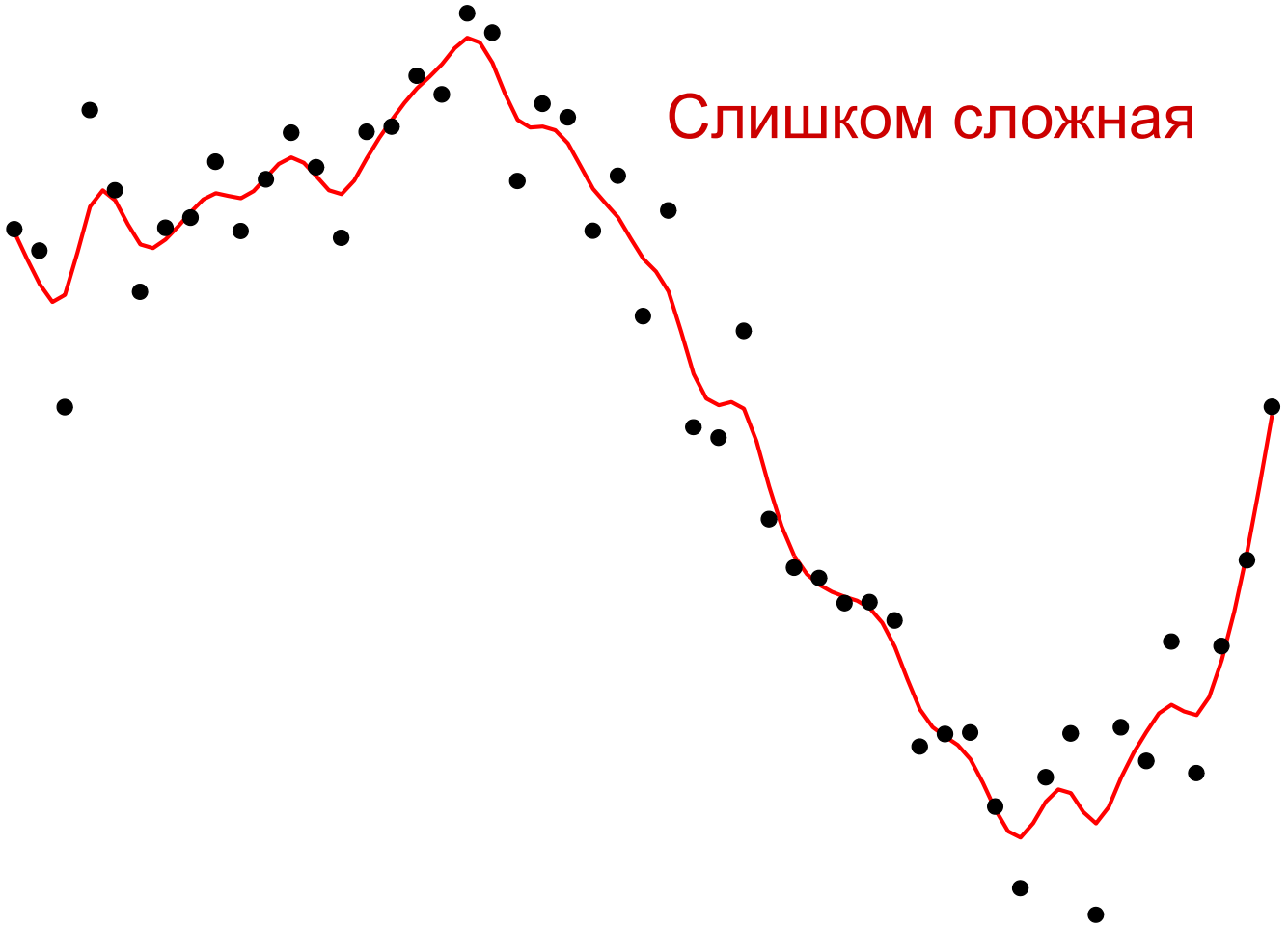
Переобучение

- Основная проблема методов машинного обучения!!!
- По сути:
  - Высокая точность на тренировочном наборе и плохая на тестовом
- Причины:
  - Сложность модели: например, для параметрических моделей много степеней свободы (параметров модели) или слишком сложное уравнение
  - Шум и выбросы в тренировочной выборке
  - Малый объем или неравномерность тренировочной выборки
- Обобщающая способность:
  - способность метода машинного обучения правильно прогнозировать «отклик» для объектов и ситуаций, которых не было в тренировочном наборе
  - метод называется состоятельным, если он с большой вероятностью делает маленькую ошибку на данных, которых не было в обучающей выборке
  - Как оценить?

Сложность модели

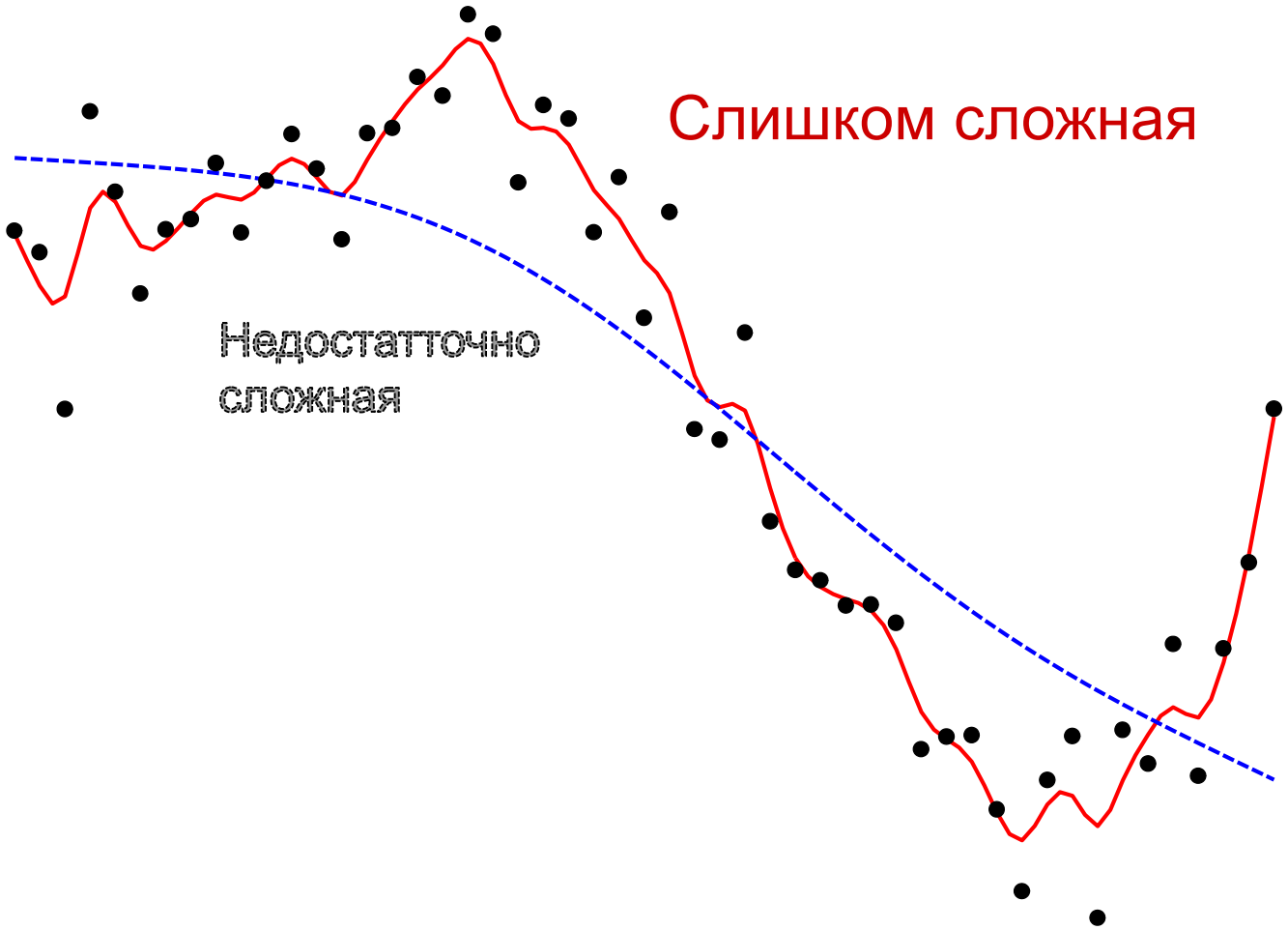
...

Сложность модели



...

Сложность модели



...

Экспериментальная оценка качества модели

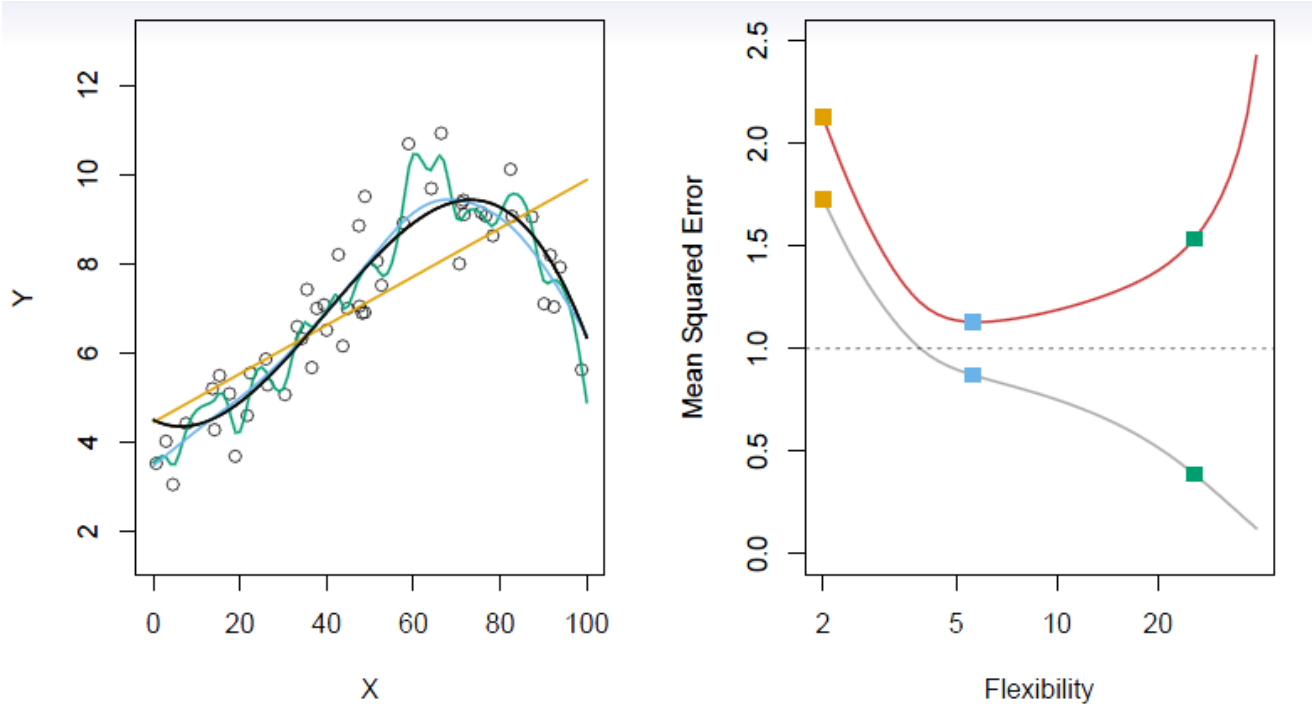
- Предположим, что мы строим модель $\hat{f}(x)$ на обучающем наборе данных $\mathsf{Tr} = \{x_i, y_i\}_1^N$ и хотим, чтобы она была наилучшей.
  - Мы можем вычислить среднеквадратичную ошибку прогнозирования для *Tr*:

$$\mathrm{MSE}_{\mathsf{Tr}} = \mathrm{Ave}_{i \in \mathsf{Tr}} [y_i - \hat{f}(x_i)]^2$$

- Оценка может быть смещена в сторону более очевидных моделей.
  - Вместо этого мы можем, если возможно, вычислить оценку, используя тестовый набор данных $\mathsf{Te} = \{x_i, y_i\}_1^M$:

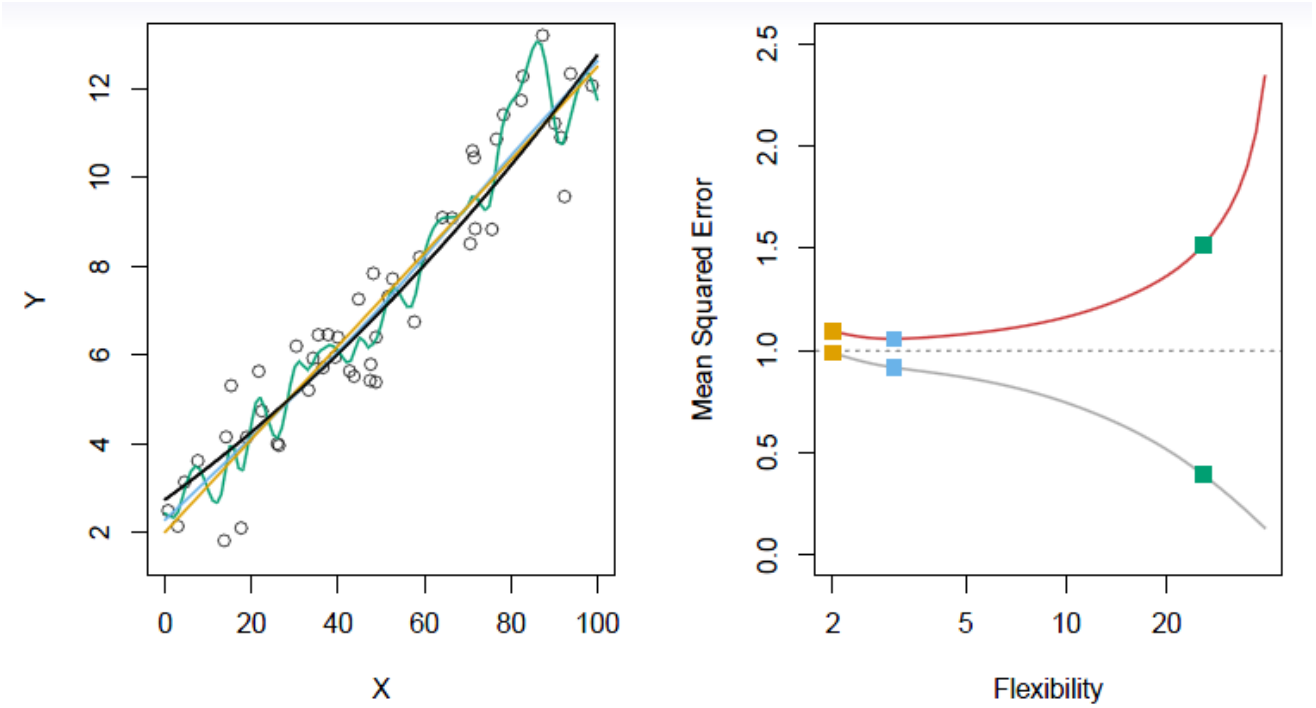
$$\mathrm{MSE}_{\mathsf{Te}} = \mathrm{Ave}_{i \in \mathsf{Te}} [y_i - \hat{f}(x_i)]^2$$

Оценка качества модели (сложная зависимость, много шума)



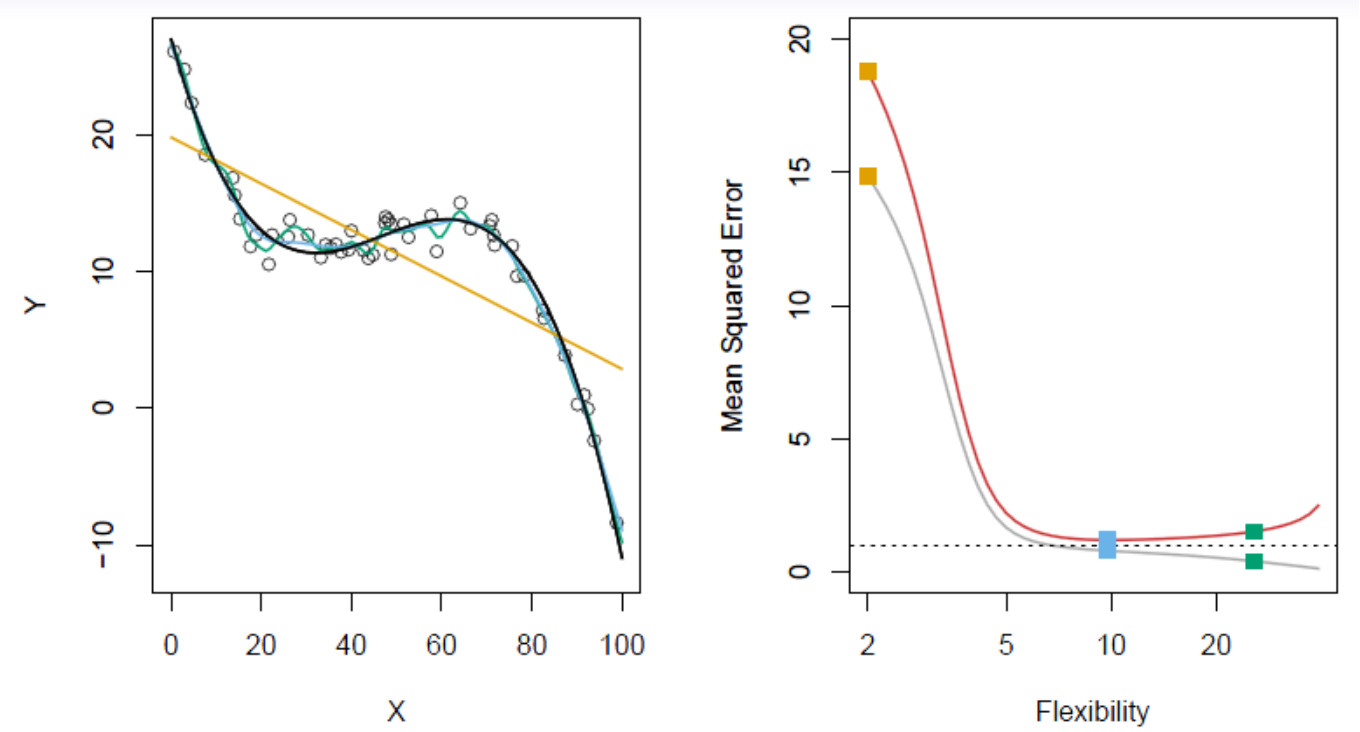
- Кривая, обозначенная черным цветом, - истинные значения.
- Красная кривая на правом рисунке – MSE_{Tr} , серая кривая – MSE_{Tr}.
- Оранжевая, голубая и зеленая кривые соответствуют подгонке моделей различной гибкости.
- Простые модели недообучены, сложные модели переобучены

Оценка качества модели (простая зависимость, много шума)



- Простые модели дают высокую обощающую способность
- Сложные модели переобучены

Оценка качества модели (сложная зависимость, мало шума)



- Простые модели недообучены
- Сложные обладают хорошей обобщающей способностью

Некоторые интуитивно понятные компромиссы

- Точность прогноза vs интерпретируемость.

  - Линейные модели легко интерпретируемы, тогда как более гибкие модели как правило - нет.

- Хорошее качество подгонки vs переобучение или недообучение.

  - Как определить, в какой момент подгонка наиболее точная?

- Простота vs черный ящик.

  - Мы часто предпочитаем более простую модель с участием меньшего количества переменных по сравнению с прогнозированием черным ящиком с участием их всех.

Компромис отклонения смещения

Пусть мы строим модель $\hat{f}(x)$ на некотором обучающем наборе *Tr,* и пусть (x_0, y_0) - некоторый тестовый образец. Если истинная модель $Y = f(X) + \epsilon$ ($f(x) = E(Y|X = x)$), то

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon).$$

Заметим, что

$$\mathrm{Bias}(\hat{f}(x_0))] = E[\hat{f}(x_0)] - f(x_0).$$

Как правило, когда *сложность* $\hat{f}(x)$ увеличивается, дисперсия возрастает, а смещение уменьшается. Таким образом, выбор сложности, основанный на средних ошибках на тестах, представляет собой *компромисс отклонения смещения*.

MSE декомпозиция

$$MSE = E[(\hat{D} - D)^2] = E[\hat{D}^2] + E[D^2] - E[2\hat{D}D] =$$

$$= Var(\hat{D}) + Var(D) + (E[\hat{D}] - E[D])^2$$

Дисперсия оценки

Дисперсия шума (не зависит от модели)

Квадрат смещения

Компромисс: Дисперсией vs Смещение!!!!

Сложнее модель => точнее приближение => меньше смещение +++

Сложнее модель => больше параметров => больше дисперсия ---

… и наоборот …

Поиск баланса между точностью и сложностью = поиск компромисса между смещением и дисперсией

MSE декомпозиция (примеры)

$$D = f(x) + \varepsilon$$

D – наблюдения, *f(.)* – истинная зависимость, ε – шум *N(0,σ)*

- **K-NN:**

$$\hat{D}(x) = \frac{1}{k}\sum_{i \in N_k(x)} D_i, Var(D) = \sigma^2, Var(\hat{D}(x)) = \frac{1}{k^2}\sum_{i \in N_k(x)} Var(D_i) = \frac{\sigma^2}{k},$$

$$\left[E(\hat{D}(x)) - f(x)\right]^2 = \left[\frac{1}{k}\sum_{i \in N_k(x)} E(D_i) - f(x)\right]^2,$$

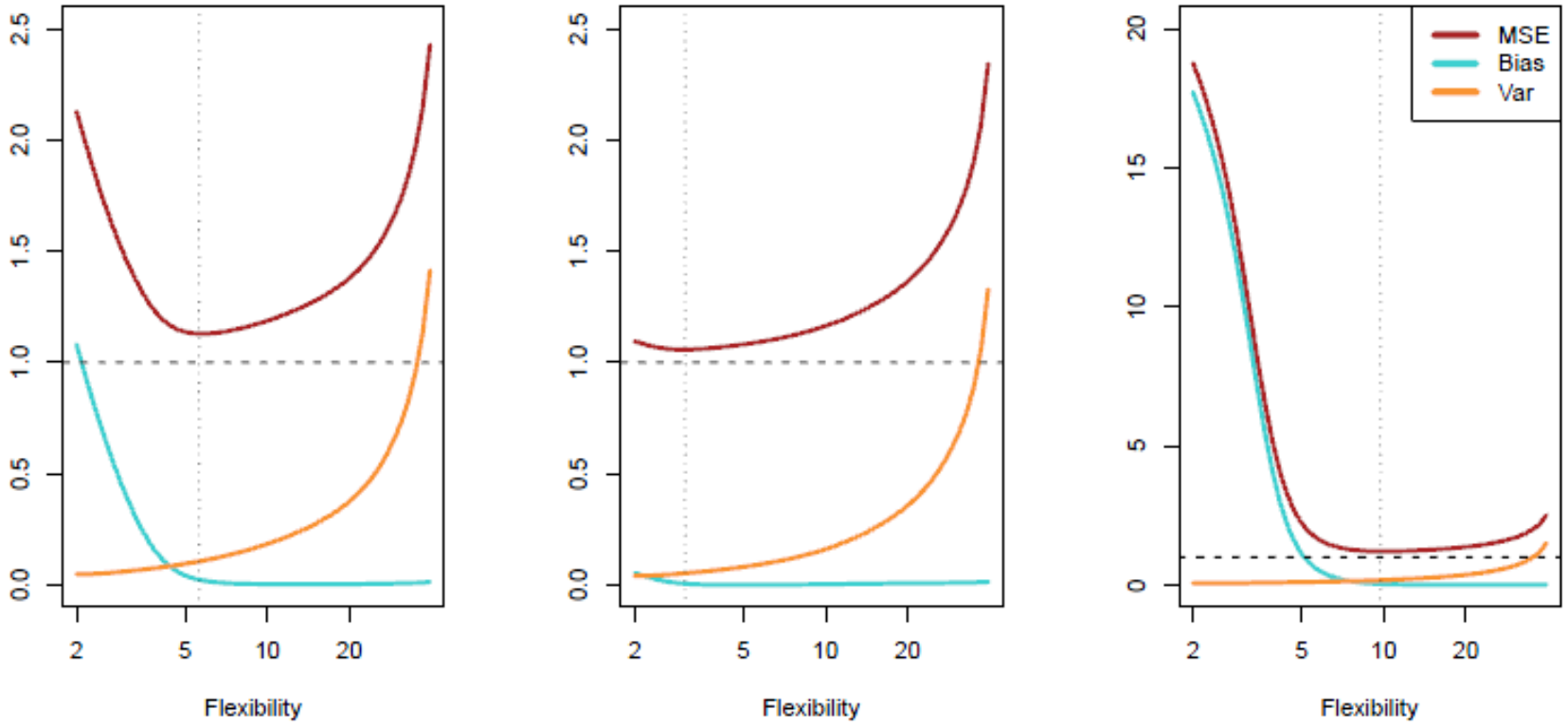
$$MSE = \sigma^2 + \frac{\sigma^2}{k} + \left[\frac{1}{k}\sum_{i \in N_k(x)} f(x_i) - f(x)\right]^2$$

- **Линейная регрессия:**

$$\hat{D}(x) = x^T (X^T X)^{-1} X^T \bar{D}, Var(D) = \sigma^2, Var(\hat{D}(x)) = \frac{p}{N}\sigma^2,$$

$$MSE = \sigma^2 + \frac{p}{N}\sigma^2 + \frac{1}{N}\sum_x \left[E[\hat{D}(x)] - f(x)\right]^2$$

Компромис отклонения смещения для трех примеров



Качество на обучающем и тестовом наборе