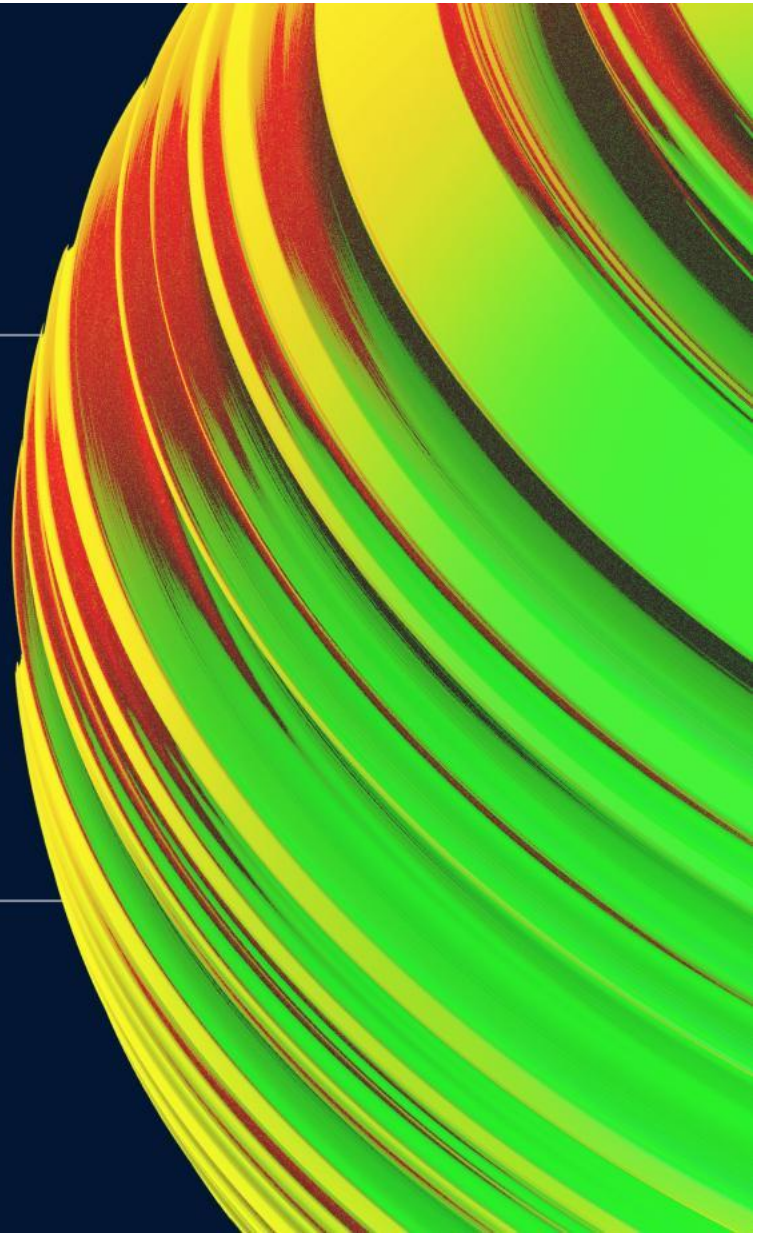
УНИВЕРСИТЕТ ИННОПОЛИС

Введение в машинное обучение. KNN. Метрики качества. Матрица ошибок

Воробьёва Мария

- maria.vorobyova.ser@gmail.com
- @SparrowMaria



План лекции

Машинное обучение. Зачем и что это такое?

Примеры применения машинного обучения

Постановка задачи машинного обучения

Алгоритм knn

Оценка качества моделей

Переобучение моделей

Машинное обучение

INNOPOLIS UNIVERSITY

Искусственный интеллект (Artificial Intelligence) — 1955

Машинное обучение (Machine Learning) — 1959

Интеллектуальный анализ данных (Data Mining) — 1989

Knowledge Discovery in Databases — 1989

Науки о данных (Data Science) — 1997

Предсказательная аналитика (Predictive Analytics) — 2007

Большие данные (Big Data) — 2008

Аналитика больших данных (Big Data Analytics)

Машинное обучение

Машинное обучение — это наука, изучающая способы извлечения закономерностей из ограниченного количества примеров.

Пример применения ML

INNOPOLIS UNIVERSITY

Допустим, мы хотим открыть новый магазин:

- *Объект - локация с параметрами*
- *Цель - “какая будет прибыль через год”*

Пример применения ML

INNOPOLIS UNIVERSITY

Допустим, мы хотим открыть новый магазин:

- *Объект - локация с параметрами*
- *Цель - “какая будет прибыль через год”*

То есть нам нужна функция:

прибыль за год = f(параметры локации, где хотим открыть магазин)

Пример применения ML

innopolis university

Допустим, мы хотим открыть новый магазин:

- *Объект - локация с параметрами*
- *Цель - “какая будет прибыль через год”*

То есть нам нужна функция:

прибыль за год = f(параметры локации, где хотим открыть магазин)

Как можно получить такую функцию?

Пример применения ML

INNOPOLIS UNIVERSITY

Допустим, мы хотим открыть новый магазин:

- *Объект - локация с параметрами*
- *Цель - “какая будет прибыль через год”*

То есть нам нужна функция:

прибыль за год = f(параметры локации, где хотим открыть магазин)

Как можно получить такую функцию?

- ***экспертно***
- ***обучаясь на примерах***

Пример применения ML

обучаясь на примерах

- то есть 1 магазин - это 1 пример

- По каждому магазину необходимо собрать ту информацию, которая нам поможет для решения первоначальной задачи:

Открывать успешные магазины с наибольшей прибылью

Пример применения ML

По каждому магазину мы можем собрать только ту информацию, которая доступна до момента открытия магазина

Нам доступны следующие характеристики магазина:

- площадь магазина
- на каком этаже ТЦ находится магазин
- в каком районе города
- что за город, соц дем характеристики города (плотность населения, безработица, %мужчин, % пенсионеров и так далее)
- а вот среднее количество людей, посещающих магазин, за 1 день неизвестно - использовать нельзя

Пример применения ML

Цель построить функцию

прибыль за год = f(параметры локации, где хотим открыть магазин)

Значит по каждому магазину, для которого собрали все доступные характеристики на момент открытия, мы также соберем прибыль за 1 года

Пример применения ML

ID магазина	1	2	3	4	5	6	7	...	n	доход от магазина
1										1000000
2										200000
3										300000
4										500000
5										600000
6										1000000
7										200000
8										300000
9										500000
10										600000
...										...
k										600000

Обучающая выборка состоит из

k магазинов с n характеристиками

и с 1 признаком - прибыль магазина через год

- *Каждый магазин - 1 строка в выборке*
- *n характеристик - независимые признаки, features, предикторы, факторы*
- *доход магазина - целевая переменная, target, метка, label*

Пример применения ML

Провели обучение (например, выбрали независимые признаки и для выбранных независимых признаков подобрали коэффициенты) :

прибыль в 1 год *= 500000 + 2*площадь_магазина +0.3*размер_населения города - 3*площадь_магазина_ближайшего_конкурента*

Теперь для новой локации мы сможем посчитать прибыль за 1 год!

И только после этого принимать решение открывать новый магазин в этой локации или нет

Примеры машинного обучения

INNOPOLIS UNIVERSITY

Объект - кредитная заявка

Классы - bad (просрочка 90+ в 1-ый год) или good

Примеры признаков:

- пол, семейное положение, наличие телефона и …
- место проживания, профессия, работодатель
- образование, должность
- возраст, зарплата, стаж работы, сумма прошлых кредитов, размер просрочки

Особенности задачи:

дисбаланс классов

Примеры машинного обучения

INNOPOLIS UNIVERSITY

Объект - абонент в определенный момент времени

Классы - уйдет или не уйдет в следующем месяце

Примеры признаков:

- тарифный план, регион проживания
- длительность разговоров, смс, частота оплаты
- корпоративный клиент, включение услуг

Особенности задачи:

признаки необходимо вычислять по сырым данным

большие данные

Примеры машинного обучения

INNOPOLIS UNIVERSITY

Объект - пациент в определенный момент времени

Классы - диагноз

Примеры признаков:

- пол, возраст, головная боль, слабость
- пульс, артериальное давление, содержание гемоглобина, другие показатели

Особенности задачи:

нужен интерпретируемый алгоритм

обычно много пропусков

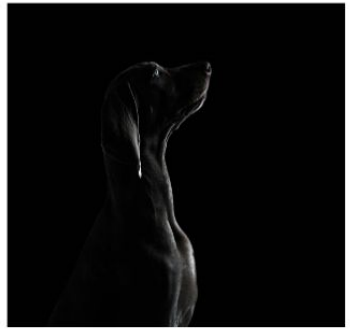
Примеры машинного обучения

INNOPOLIS UNIVERSITY

Объект *- изображение или видеопоследовательность*

Классы*: Собаки/Кошки (или решение объехать/не объехать)*

Методы*: Используются глубокие нейросетевые архитектуры (Deep Learning)*



Классификация задач машинного обучения

Постановка задачи машинного обучения

INNOPOLIS UNIVERSITY

Пусть

X - множество описаний объектов

Y - множество допустимых ответов

Существует неизвестная целевая зависимость y^*

отображение $y^* : X \to Y$

значения y^* известны только на объектах конечной обучающей выборки X^n

$$X^n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$$

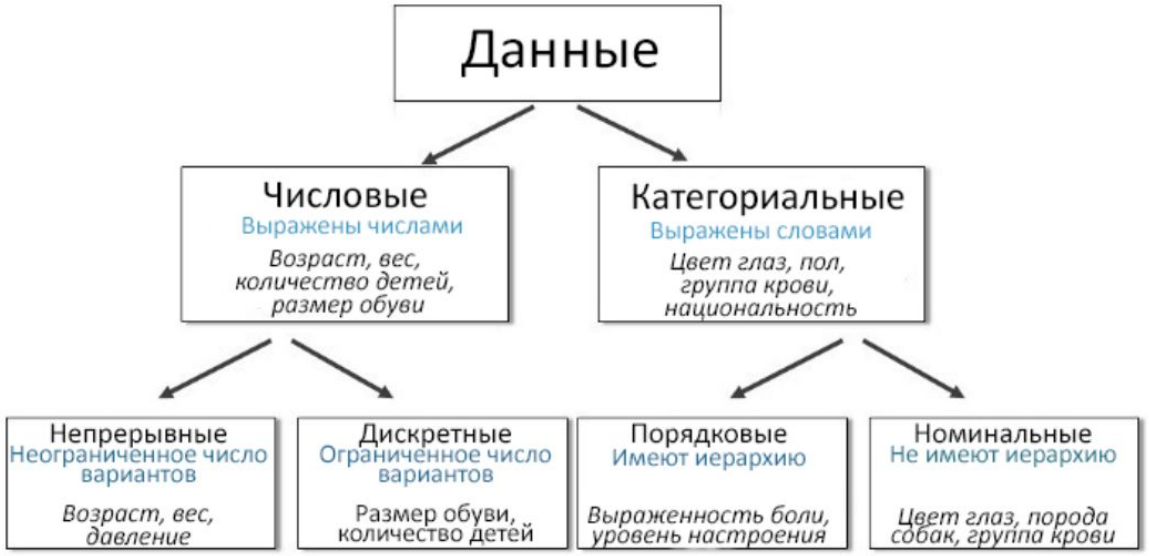
Требуется построить алгоритм a**, который приближал бы неизвестную целевую зависимость как на элементах выборки** X^n**, так и на всем множестве** X

Постановка задачи машинного обучения

Данные X и Y *- репрезентативная выборка, то есть она должна быть iid (independent and identically distributed)*

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Target (passed)
John	22	5	4	Brown	English	5	TRUE
Aahna	17	4	5	Brown	Hindi	4	TRUE
Emily	25	5	5	Blue	Chinese	5	TRUE
Michael	27	3	4	Green	French	5	TRUE
Some student	23	3	3	NA	Esperanto	2	FALSE

Независимые признаки и целевая переменная *могут быть разного типа*



Постановка задачи машинного обучения

INNOPOLIS UNIVERSITY

Гиперпараметры модели *— задаются перед обучением моделей*

Параметры модели *— то, что находим во время обучения*

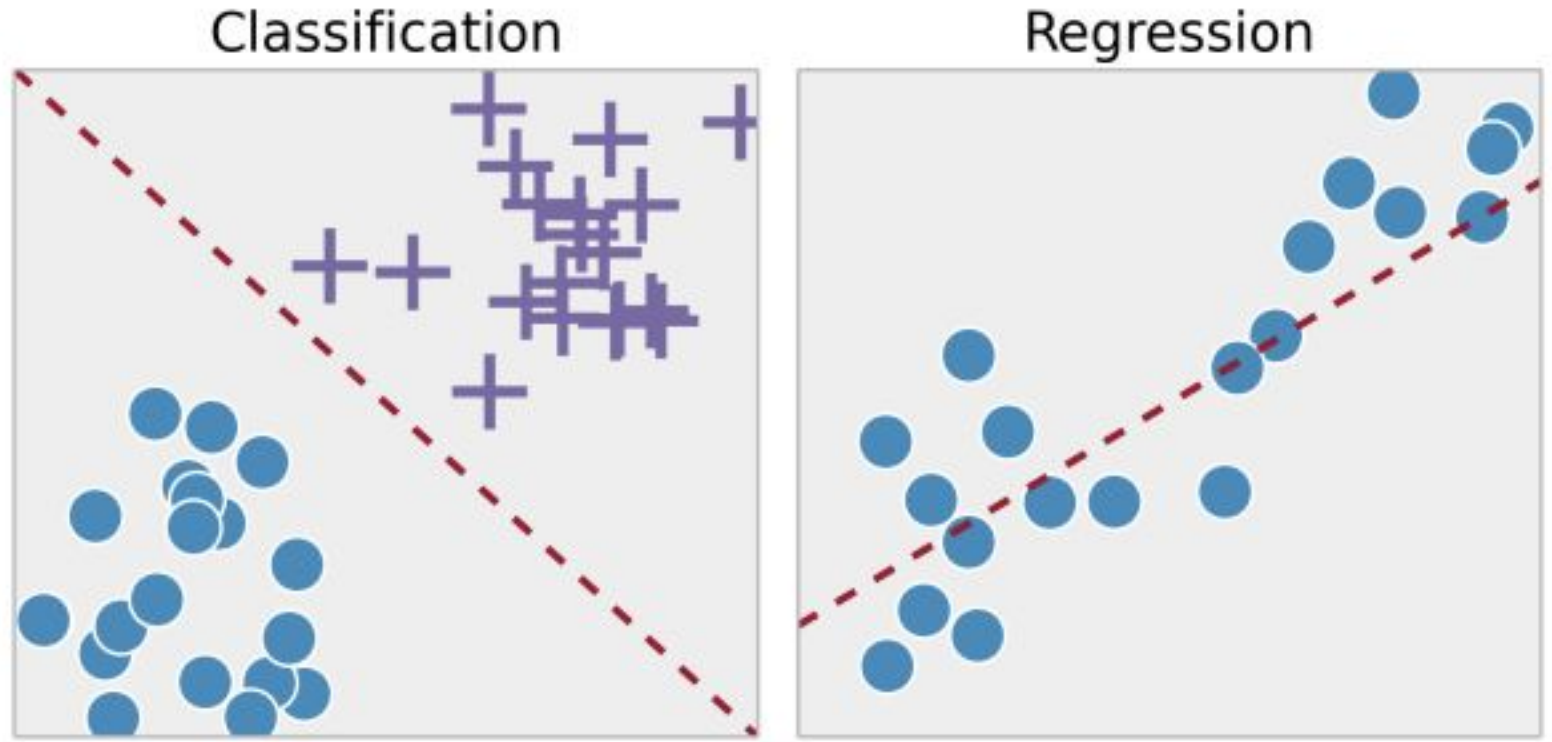
Loss function *— функция, которая характеризует потери при неправильном принятии решений на основе наблюдений, то есть эта функция оценивает на сколько модель ошибается на данных*

Метрики качества *— это числовые показатели или статистики, используемые для оценки качества или производительности системы, модели или алгоритма. В контексте машинного обучения и анализа данных метрики часто используются для измерения точности, эффективности или других аспектов работы модели или алгоритма.*

Классификация или регрессия?

Классификация - целевая категориальная. Цель: найти такую поверхность, которая разделяет классы

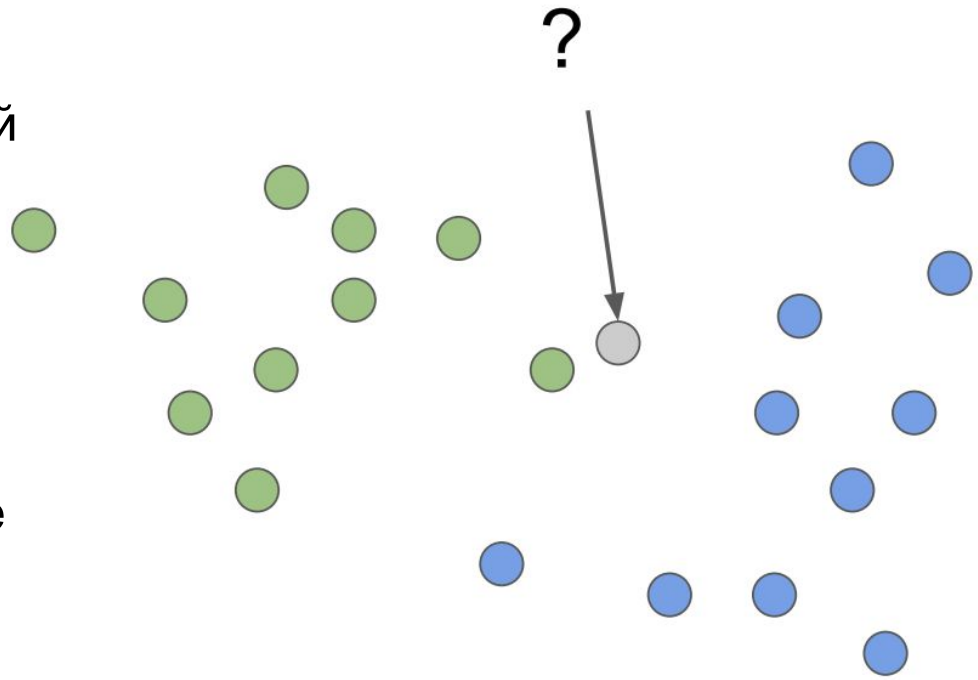
Регрессия - целевая числовая. Цель регрессии — найти такую поверхность, которая проходит через точки обучающей выборки, минимизируя отклонения между предсказанными значениями и реальными числовыми значениями целевой переменной.



Алгоритм kNN - k Nearest Neighbours

Для **классификации** каждого из объектов тестовой выборки необходимо последовательно выполнить следующие операции:

- Вычислить расстояние до каждого из объектов обучающей выборки
- Отобрать объекты обучающей выборки, расстояние до которых минимально
- Класс классифицируемого объекта — это класс, наиболее часто встречающийся среди ближайших соседей



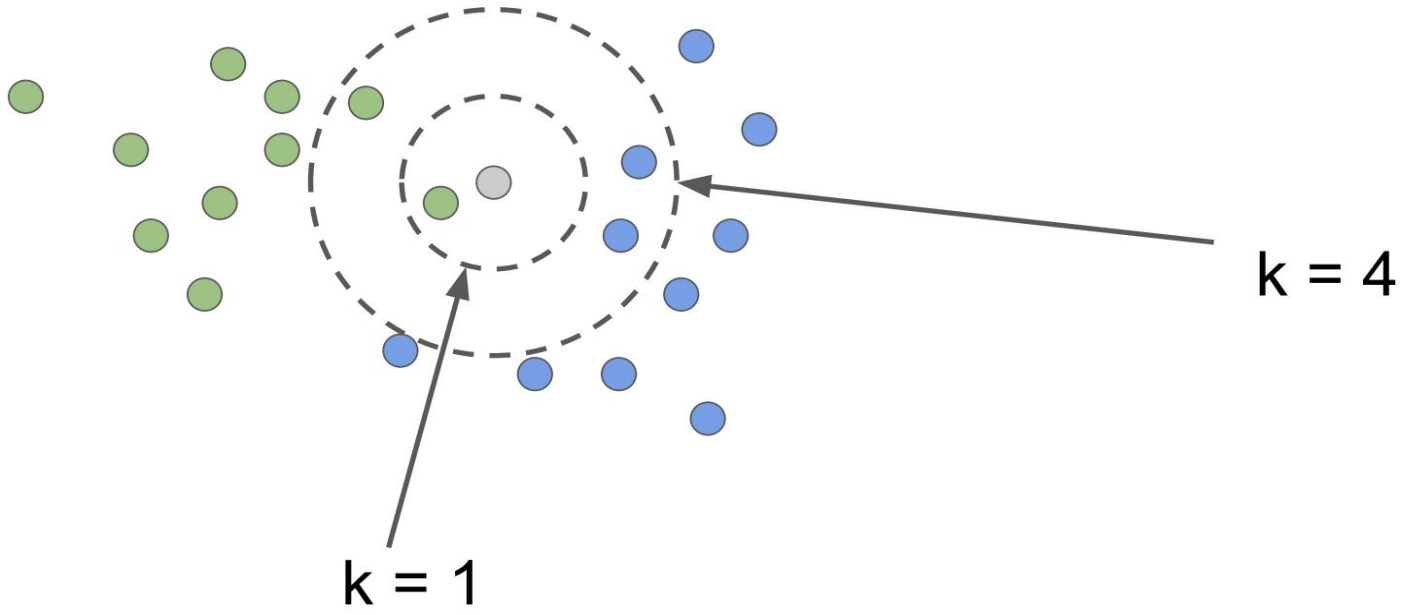
Для **задачи регрессии** возвращается не метка, а число – среднее (или медианное) значение целевого признака среди соседей.

алгоритм kNN - k Nearest Neighbours

Число k - гиперпараметр модели

Расстояние:

- Евклидовое расстояние
- Minkowski (расстояние Минковского)
- косинусное расстояние
- и так далее



$$d(x,y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

$$\text{Minkowski Distance} = \left(\sum_{i=1}^{n}\left|x_i - y_i\right|^p\right)^{1/p}$$

$$\text{Manhattan Distance} = d(x,y) = \left(\sum_{i=1}^{m}\left|x_i - y_i\right|\right)$$

Алгоритм kNN. Как можно улучшить алгоритм?

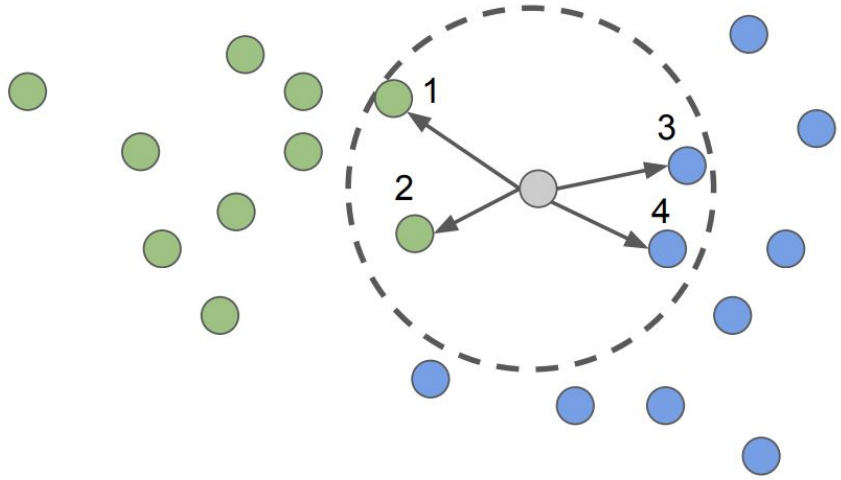
Будем учитывать расстояние до каждого объекта
в явном виде

$$w(\mathbf{x}_{(i)}) = w(d(\mathbf{x}, \mathbf{x}_{(i)}))$$

$$p_{\text{green}} = \frac{w(\mathbf{x}_1) + w(\mathbf{x}_2)}{w(\mathbf{x}_1) + w(\mathbf{x}_2) + w(\mathbf{x}_3) + w(\mathbf{x}_4)}$$

$$p_{\text{blue}} = \frac{w(\mathbf{x}_3) + w(\mathbf{x}_4)}{w(\mathbf{x}_1) + w(\mathbf{x}_2) + w(\mathbf{x}_3) + w(\mathbf{x}_4)}$$

k = 4



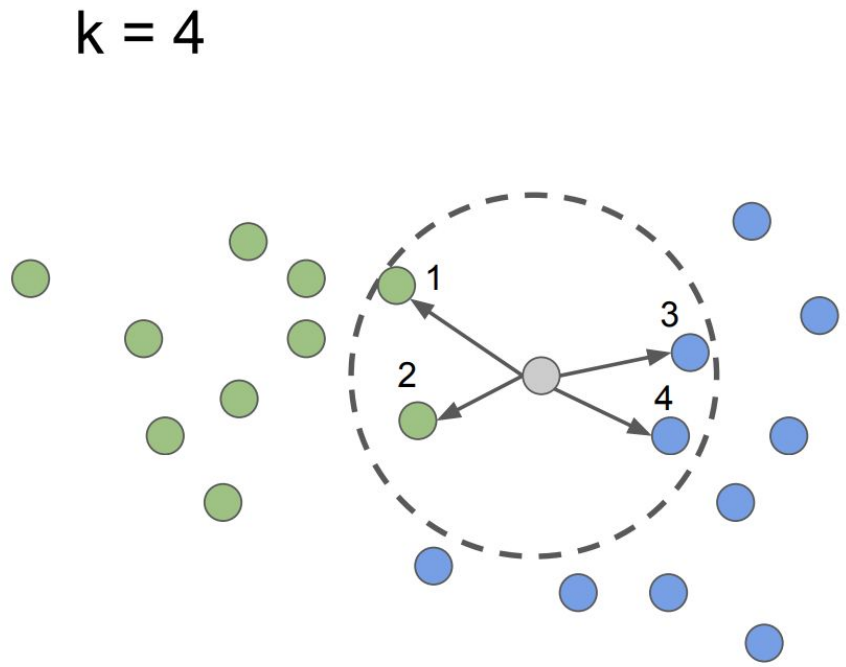
Алгоритм kNN. Как можно улучшить алгоритм?

INNOPOLIS UNIVERSITY

Гиперпараметр:

- функция расстояния
- число k

Рекомендуется выборку X нормировать:



Минусы:

- классический алгоритм плохо работает на больших данных

INNOPOLIS UNIVERSITY

А как понять, что мы построили хорошую модель?

Как измерить качество моделей?

Метрики качества зависят от типа целевой переменной:
если целевая числовая: вещественная

ID магазина	1	2	3	4	5	6	7	...	n	доход от магазина	прогноз		
1										1000000	999900	100	10000
2										200000	200100	100	10000
3										300000	299870	130	16900
4										500000	499800	200	40000
5										600000	600200	200	40000
6										1000000	1000150	150	22500
7										200000	199900	100	10000
8										300000	299900	100	10000
9										500000	499800	200	40000
10										600000	600200	200	40000
...										...			
k										600000	600300	300	90000
												161.8	29945.5

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - Y_i^p)^2,$$

$$RMSE = \sqrt{MSE}$$

$$MAE = \frac{1}{N}abs(Y_i - Y_i^p)$$

где

Y_i^p - прогнозное значение ,

Y_i - фактическое значение,

N - количество объектов

Как измерить качество моделей?

Метрики качества зависят от типа целевой переменной:
если целевая числовая: вещественная

ID магазина	1	2	3	4	5	6	7	...	n	доход от магазина	прогноз		
1										1000000	999900	100	10000
2										200000	200100	100	10000
3										300000	299870	130	16900
4										500000	499800	200	40000
5										600000	600200	200	40000
6										1000000	1000150	150	22500
7										200000	199900	100	10000
8										300000	299900	100	10000
9										500000	499800	200	40000
10										600000	600200	200	40000
...										...			
k										600000	600300	300	90000
												161.8	29945.5

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(Y_i - Y_i^p)^2}{\sum_{i=1}^{N}(Y_i - Y_{avg})^2}$$

где

Y_i^p - прогнозное значение ,

Y_i - фактическое значение,

N - количество объектов

Коэффициент детерминации измеряет долю дисперсии, объяснённую моделью, в общей дисперсии целевой переменной.
Фактически, данная мера качества — это нормированная среднеквадратичная ошибка. Если она близка к единице, то модель хорошо объясняет данные, если же она близка к нулю, то прогнозы сопоставимы по качеству с константным предсказанием.

Как измерить качество моделей?

$$R^2_{adj} = 1 - \frac{s^2}{s_y^2} = 1 - \frac{RSS/(n-k)}{TSS/(n-1)} = 1 - (1 - R^2)\frac{(n-1)}{(n-k)} \leq R^2,$$

где

$RSS = \sum_{n}^{t=1} e_t^2 = \sum_{n}^{t=1} (y_t - \hat{y}_t)^2$ — сумма квадратов остатков регрессии,

$TSS = \sum_{n}^{t=1} (y_t - \bar{y})^2 = n\hat{\sigma}_y^2$ — общая дисперсия,

n — количество наблюдений в наборе данных,

k — количество параметров модели.

Не учитывает качество модели:

- Высокий R2 не всегда означает, что модель хорошая. Модель может объяснять значительную часть дисперсии, но при этом иметь значительные ошибки предсказания.

Зависимость от числа предикторов:

- R2 всегда увеличивается с добавлением новых предикторов, даже если они незначительно улучшают модель. Это может привести к избыточной подгонке модели (overfitting).

Не измеряет предсказательную силу:

- R2 показывает, насколько хорошо модель объясняет дисперсию в обучающих данных, но не говорит о её предсказательной способности на новых данных. Для оценки предсказательной силы используются другие метрики, такие как RMSE, MAE и т.д.

Как измерить качество моделей?

INNOPOLIS UNIVERSITY

MSE (или RMSE)

- сильное влияние оказывают выбросы
- интерпретируемая метрика

MAE

- интерпретируемая метрика
- не учитывает масштаб ошибки, 10 будет ошибкой и для 1000 и 1010 и для ситуации 10 и 20

Absolute Total Difference (разность между суммарным прогнозом и суммарным фактом) или **Bias (Relative Total Difference** - отношение суммарного прогноза к суммарному факту минус 1)

- подсвечивает есть ли в целом сдвиг модели вверх или вниз
- понятна бизнесу
- метрика отвечает только за глобальную точность

Как измерить качество моделей?

$$MAPE(y^{true}, y^{pred}) = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - f(x_i)|}{|y_i|}$$

Плюсы:

- Прост в интерпретации и расчете.
- Учитывает масштаб ошибки, отклонение на 10 ед при факте 2000 менее будет критично, чем при факте 20

Минусы:

- Неустойчив к нулевым значениям в базовых данных (когда фактические значения равны нулю), что может вызвать проблемы при расчете. Есть доработка, например, max(y, eps) или y заменяем на y+eps, но нет алгоритма как правильно выбрать eps
- Не симметричен и может давать разные результаты в зависимости от порядка фактических и прогнозных значений. За перепрогноз штрафует больше, чем за недопрогноз

Month Year	Actual Spend	Forecasted Spend	Absolute Percentage Error
Jan-22	0.2	1	400.00

Month Year	Actual Spend	Forecasted Spend	Absolute Percentage Error
Jan-22	500	600	20.00
Feb-22	600	500	16.67

Как измерить качество моделей?

$$SMAPE(y^{true}, y^{pred}) = \frac{1}{N}\sum_{i=1}^{N}\frac{2\,|y_i - f(x_i)|}{y_i + f(x_i)}$$

SMAPE (Symmetric Mean Absolute Percentage Error):

Плюсы:

Симметричен и не зависит от порядка фактических и прогнозных значений.

Обрабатывает нулевые значения более устойчиво, чем MAPE.

Минусы:

Может быть бесконечным, если фактическое значение равно нулю, и прогноз также равен нулю.

Менее интуитивен в интерпретации

Как измерить качество моделей?

Метрики качества зависят от типа целевой переменной: если целевая категориальная:

- матрица ошибок (Confusion matrix)
- Accuracy
- точность и полнота (Precision и Recall)
- F1-мера
- ROC-кривая(ROC Curve)
- ROC AUC (площадь под ROC Curve)

Как измерить качество моделей?

Необходимо снова сравнить факт с прогнозом, задаем cutoff

для прогноза и от оценки вероятностей переходим к прогнозу 0

id клиента	fact	прогноз модели	бинарный прогноз модели
1	1	0.2	0
2	0	0.3	0
3	1	0.5	1
4	0	0.1	0
5	0	0.2	0
6	1	0.55	1
7	1	0.7	1
8	1	0.8	1
9	0	0.55	1

COUNT of id	прогноз	
fact	0	1
0	3	1
1	1	4

Матрица ошибок

True Positives	False Positives	Число наблюдений классифицированных как **P**
False Negatives	True Negatives	Число наблюдений классифицированных как **N**
Число наблюдений из **P** (TP + FN)	Число наблюдений из **N** (FP + TN)	

Если результат классификации положительный (или 1) и фактическое значение тоже положительное (то есть тоже 1), то **TRUE POSITIVE (TP)**

Если результат классификации положительный (или 1) и фактическое значение отрицательное (то есть -1 или 0), то **FALSE POSITIVE (FP)**

Если результат классификации отрицательный (-1, или 0) и фактическое значение положительное (то есть тоже 1), то **TRUE NEGATIVE (TN)**

Если результат классификации отрицательный (-1 или 0) и фактическое значение тоже отрицательное (-1 или 0) , то **FALSE NEGATIVE (FN)**

Accuracy

innopolis university

True Positives	False Positives	Число наблюдений классифицированных как **P**
False Negatives	True Negatives	Число наблюдений классифицированных как **N**
Число наблюдений из **P** (TP + FN)	Число наблюдений из **N** (FP + TN)	

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy

True Positives	False Positives	Число наблюдений классифицированных как **P**
False Negatives	True Negatives	Число наблюдений классифицированных как **N**
Число наблюдений из **P** (TP + FN)	Число наблюдений из **N** (FP + TN)	

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Пример:

пусть в выборке

1. 10000 выданных кредитов
2. 100 кредитов достигли просрочку 90+ за 1-ый год, то есть доля “1” = 0.01

Теперь возьмем “самый глупый” алгоритм, который всем объектам прогнозирует 0,
то accuracy = 0.99.

Это хороший алгоритм?

Матрица ошибок

True Positives	False Positives	Число наблюдений классифицированных как **P**
False Negatives	True Negatives	Число наблюдений классифицированных как **N**
Число наблюдений из **P** (TP + FN)	Число наблюдений из **N** (FP + TN)	

Precision = TP/(TP+FP)
Чувствительность алгоритма (**sensitivity**) = TP/(TP+FP)

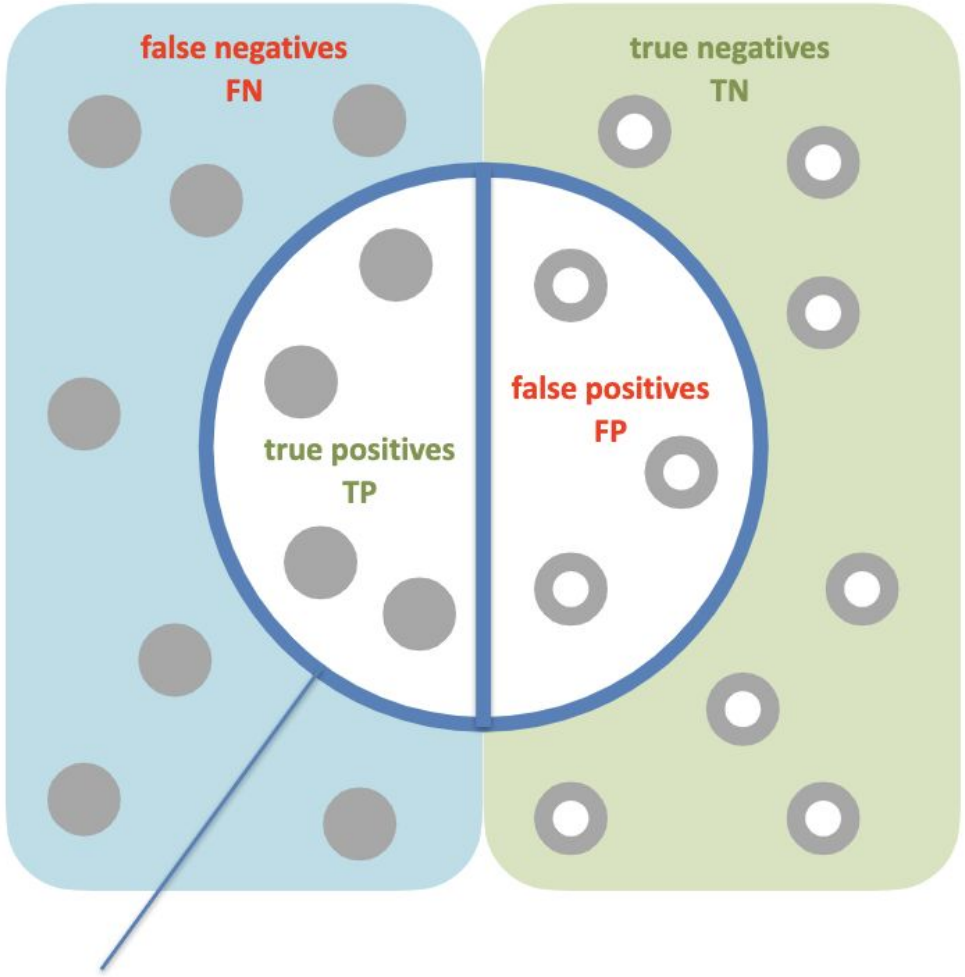
Recall = TP/(TP+FN)

TPR (True Positive Rate) = TP/(TP+FN)

Specificity = TN/(TN+FP) = 1 - FPR = 1-FP/(TN+FP)

https://habr.com/ru/companies/ods/articles/328372/

Precision и recall



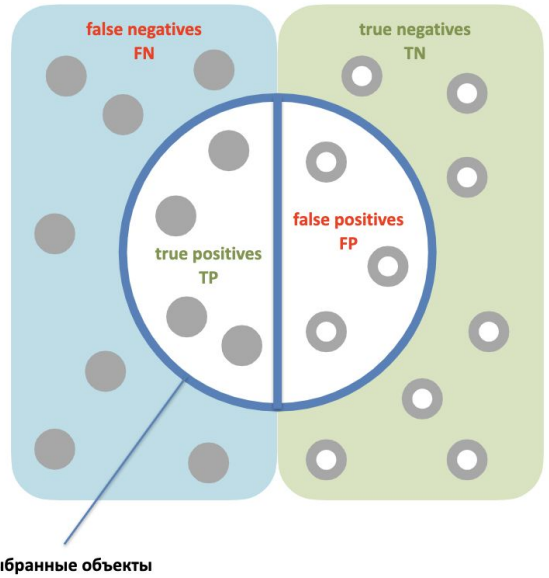
Сколько выбранных объектов корректны?

$$precision = \frac{TP}{TP + FP}$$

Как много корректных объектов выбрано?

$$recall = \frac{TP}{TP + FN}$$

Precision и recall



Сколько выбранных объектов корректны?

$$precision = \frac{TP}{TP + FP}$$

Как много корректных объектов выбрано?

$$recall = \frac{TP}{TP + FN}$$

Пример:

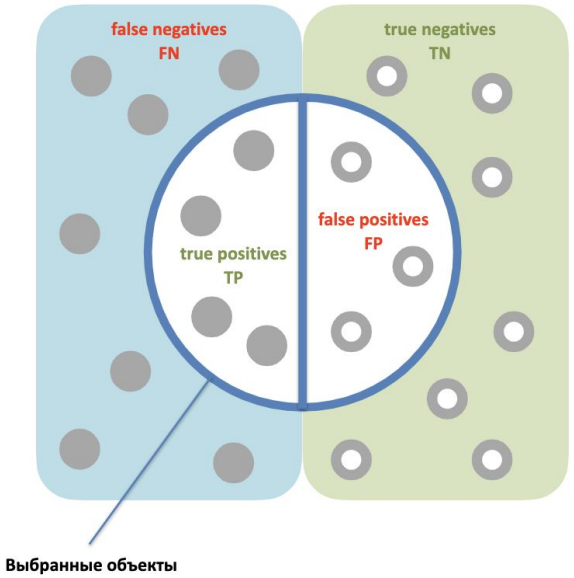
пусть в выборке

1. 10000 выданных кредитов
2. 100 кредитов достигли просрочку 90+ за 1-ый год, то есть доля “1” = 0.01

Теперь возьмем “самый глупый” алгоритм, который всем объектам прогнозирует 0

Precision = 0/0
Recall =0/100 = 0

Precision и recall

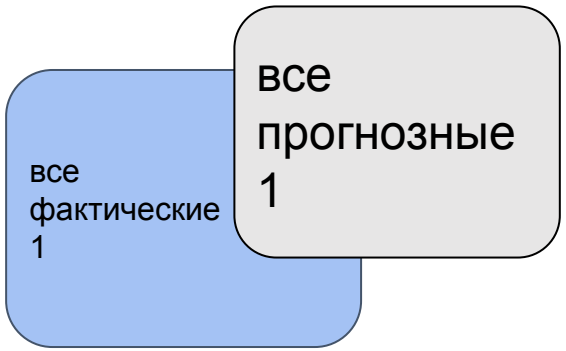


Сколько выбранных объектов корректны?

$$precision = \frac{TP}{TP + FP}$$

Как много корректных объектов выбрано?

$$recall = \frac{TP}{TP + FN}$$



Невозможно уменьшить и precision, и recall

Что же делать?

Все зависит от задачи:

Например, задача **детектировать болезнь клиента**:
Тогда нам важен recall, пропустить клиента с болезнью будет хуже, чем если мы кого-то лишний раз проверим

а вот в задаче **не пропустить безбилетника,** важна точность, то есть precision

Как объединить Precision и Recall

INNOPOLIS UNIVERSITY

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

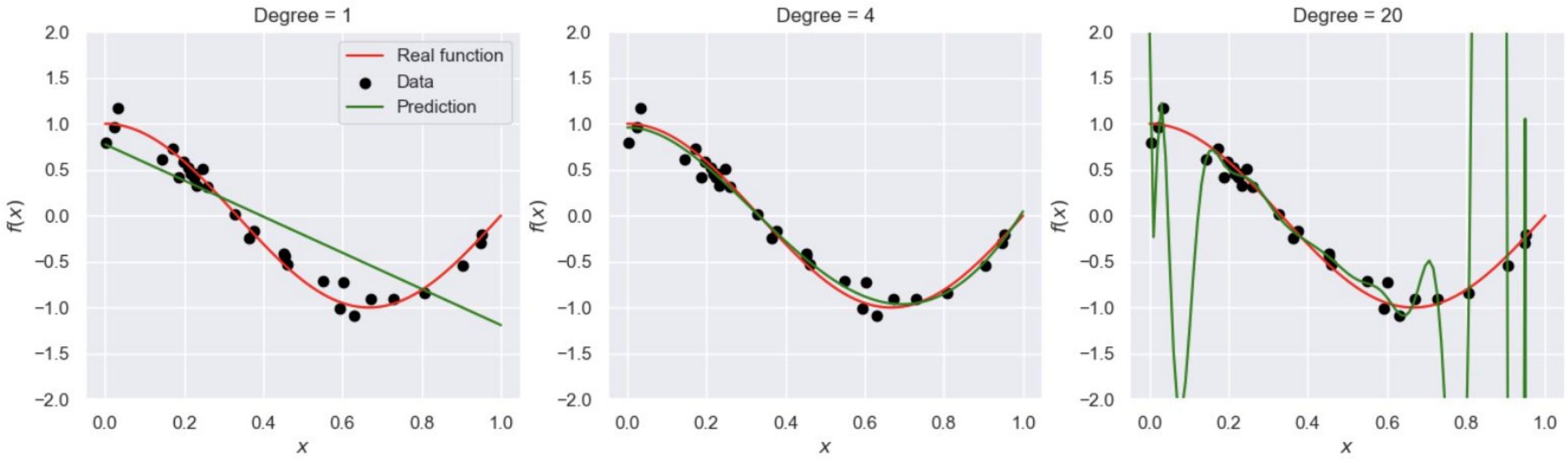
https://neerc.ifmo.ru/wiki/index.php?title=Оценка_качества_в_задачах_классификации

Теперь все ясно!

надо просто строить модель с идеальными метриками качества!

Но тут мы встречаемся с такой проблемой как ПЕРЕОБУЧЕНИЕ

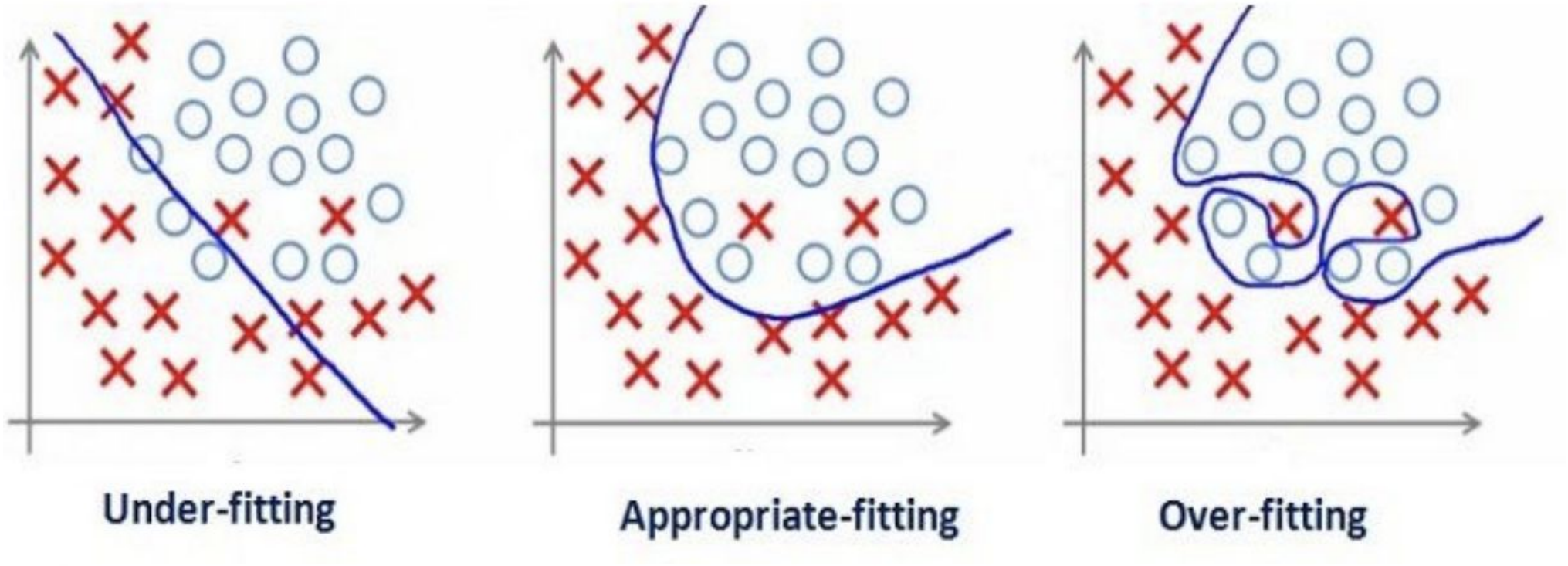
Проблема переобучения



Недообучение (underfitting) - модель слишком проста

Переобучение (overfitting) - модель склишком сложна

Проблема переобучения



Недообучение (underfitting) - модель слишком проста

Переобучение (overfitting) - модель склишком сложна

Проблема переобучения

INNOPOLIS UNIVERSITY

Причина переобучения

- слишком сложная модель
- избыточные параметры в модели, то есть наблюдается мультиколлинеарность
- иногда переменные из “будущего”

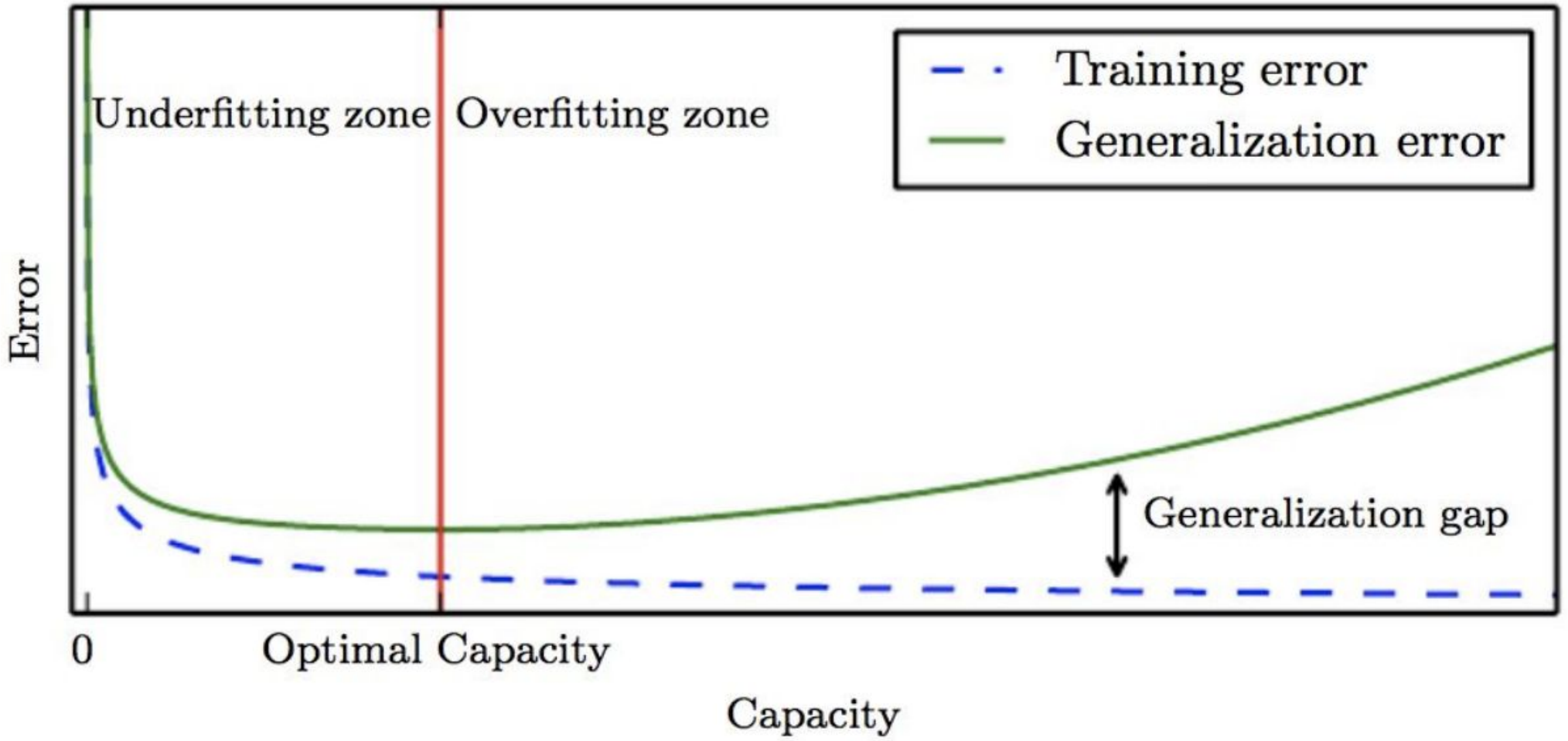
Как обнаружить переобучение

- эмпирически, путем разбиения на выборки и замерения качества моделей на отложенной выборке (тестовой)

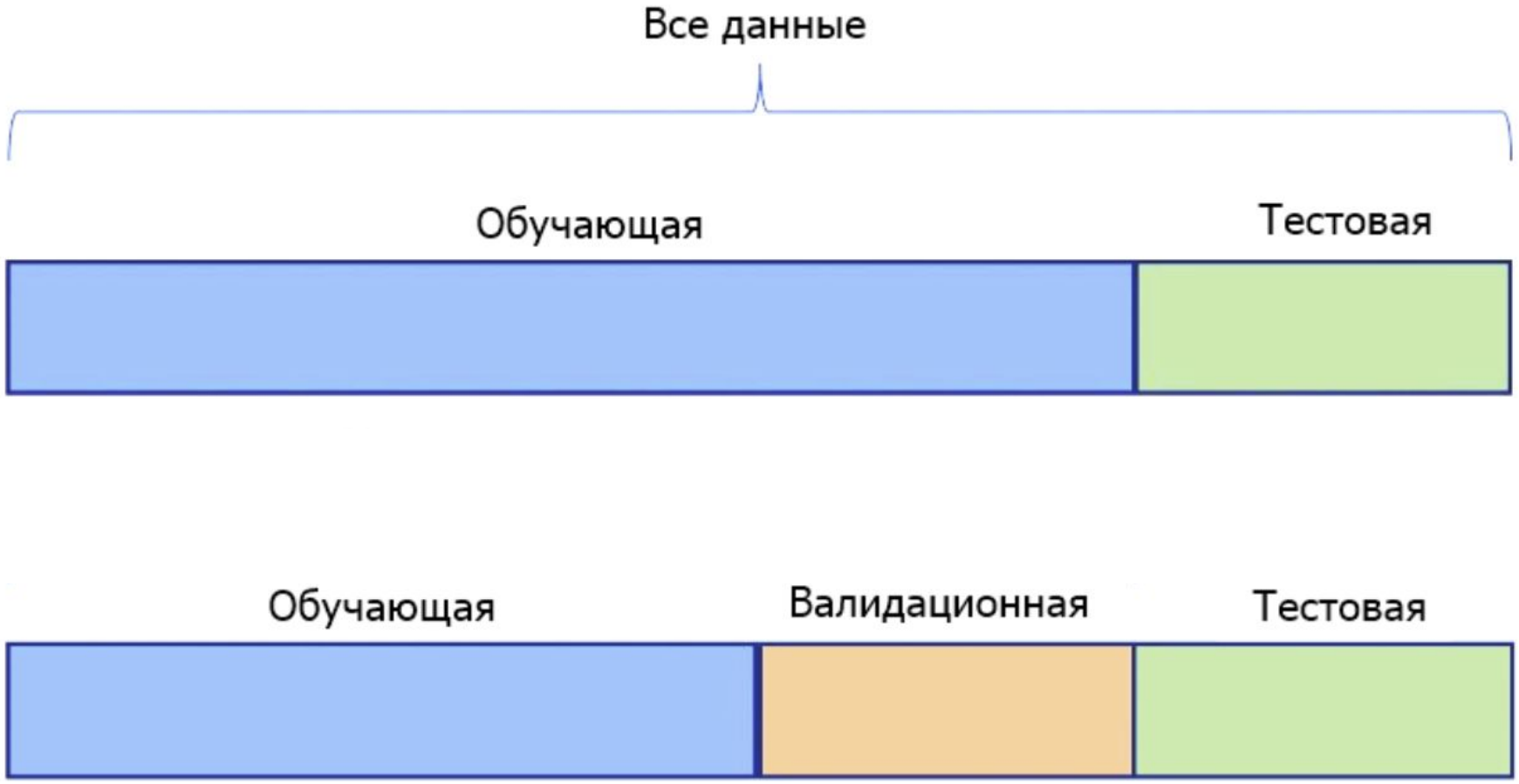
Избавить от переобучения нельзя, его можно МИНИМИЗИРОВАТЬ

- накладывать ограничения на коэффициенты при независимых признаках
- выбрать модель по оценкам обобщающей способности

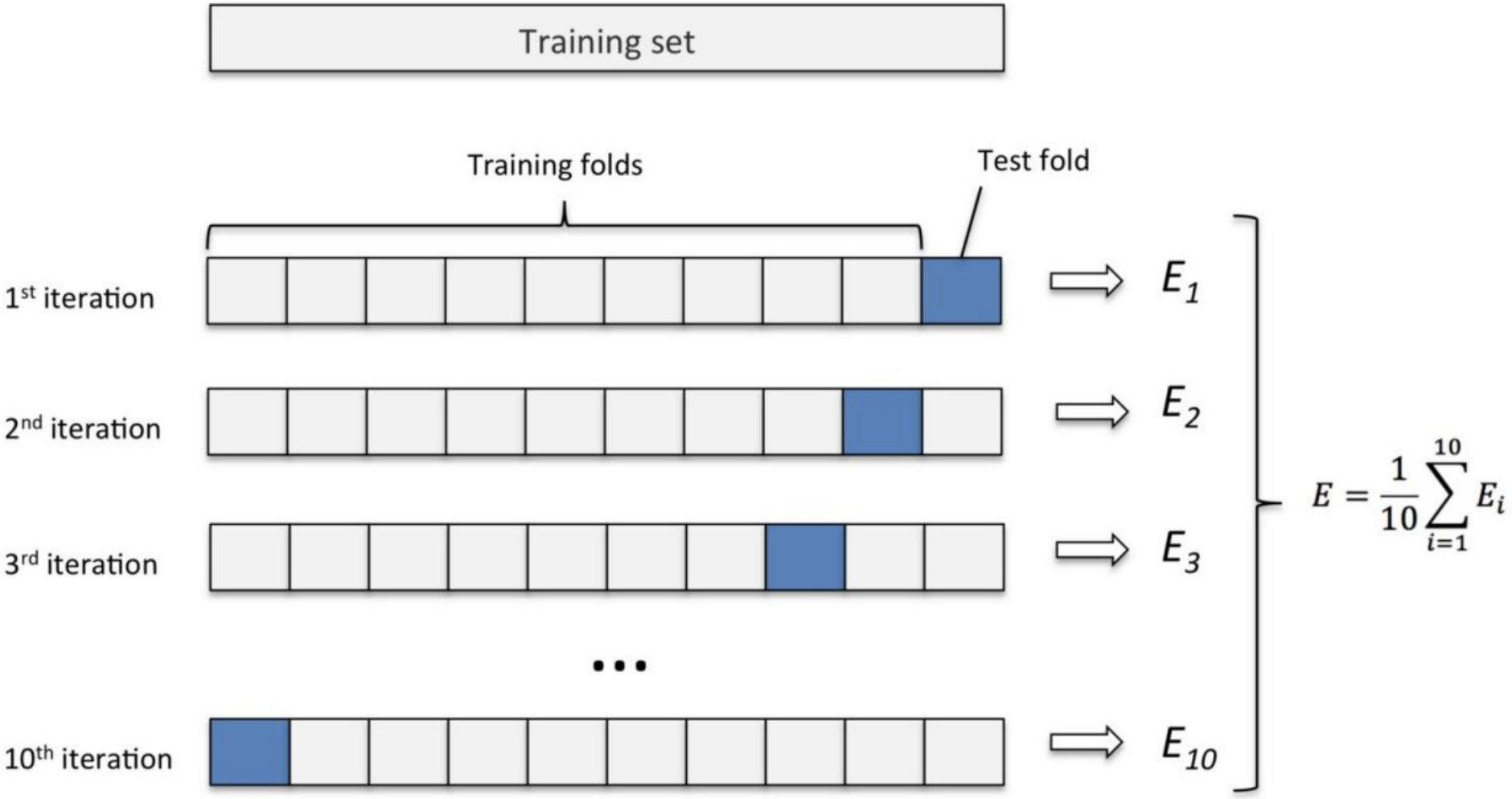
Проблема переобучения



Проблема переобучения



Проблема переобучения



УНИВЕРСИТЕТ
ИННОПОЛИС

ВОПРОСЫ И ОТВЕТЫ