Отбор признаков (feature selection) и объектов

Лекция №11

Лектор: Артём Шевляков

Простые методы отбора признаков

Почему количество признаков иногда нужно уменьшать?

1. Экономия времени и памяти.
2. Когда много признаков труднее найти закономерность.
3. Между нецелевыми признаками могут существовать зависимости – многие модели предсказания в этой ситуации работают плохо (например: линейная регрессия, наивный Байес).
4. Если признаков очень много – то возникает «проклятие размерности». Метрические и линейные методы становятся неэффективными.

Какие признаки – кандидаты на удаление?

Допустим, у нас есть таблица с нецелевыми признаками X_1, X_2,… X_n и целевым признаком Y.

Какие из признаков X_1, X_2,… X_n можно удалить?

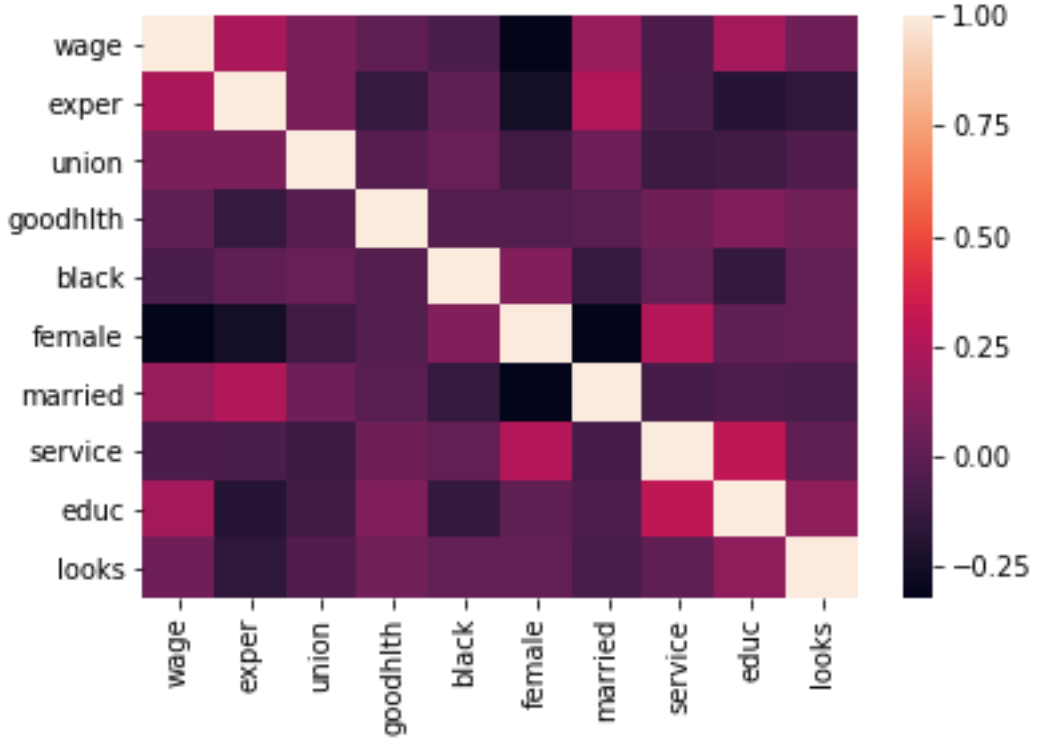
Ваши предложения?

Какие признаки – кандидаты на удаление?

Следующие признаки рекомендуется удалять:

1. Признаки с большим числом пропусков и косяков данных.
2. Числовые признаки с ОЧЕНЬ малым отклонением (в частности все константные признаки).
3. Если между признаками X_1, X_2 очень высокая корреляция, то один из них можно удалить. Для этого составляют матрицу корреляции (см. следующий слайд).

Матрица корреляции признаков



https://github.com/7bits/ml-course-7bits/blob/master/2017-fall/practises/02-pandas-numpy/02-03_pandas_tutorial.ipynb

Какие признаки – кандидаты на удаление?

Следующие признаки рекомендуется удалять:

4. Можно удалить признак X, если его корреляция с Y близка к 0 (хотя тут надо аккуратнее).
5. Для признаков X_i вычислить их информативность (энтропию, неопределенность Джини…) и удалить признаки с наихудшими показателями.
6. Запустить модель предсказания, которая (помимо своей основной работы) умеет определять значимость каждого из признаков. К таким моделям относится, например, линейные модели (в том числе линейная регрессия, логист. регрессия, их регуляризации и лассо).
Значимость каждого признака у линейной модели – это…

Какие признаки – кандидаты на удаление?

6. Запустить модель предсказания, которая (помимо своей основной работы) умеет определять значимость каждого из признаков. К таким моделям относится, например, линейные модели (в том числе линейная регрессия, логист. регрессия, их регуляризации и лассо).
Значимость каждого признака у линейной модели – это коэффициент при этом признаке.

$$Y=1.5X_1+0.01X_2-2X_3+10$$

Деревья также могут находить наиболее значимые признаки. Это будут какие признаки?...

Какие признаки – кандидаты на удаление?

6. Запустить модель предсказания, которая (помимо своей основной работы) умеет определять значимость каждого из признаков. К таким моделям относится, например, линейные модели (в том числе линейная регрессия, логист. регрессия, их регуляризации и лассо).
Значимость каждого признака у линейной модели – это коэффициент при этом признаке.

$$Y=1.5X_1+0.01X_2-2X_3+10$$

Деревья также могут находить наиболее значимые признаки. Это будут признаки попавшие в дерево!

Random Forest также может найти наиболее значимые признаки.

Сложные методы отбора признаков

Отбор признаков в несколько итераций

Можно перебрать все подмножества признаков, для каждого подмножества построить модель предсказания. Выбрать подмножество с наилучшим качеством предсказания.

Но это очень трудоемко. Если признаков m штук, то нужно построить 2^m моделей.

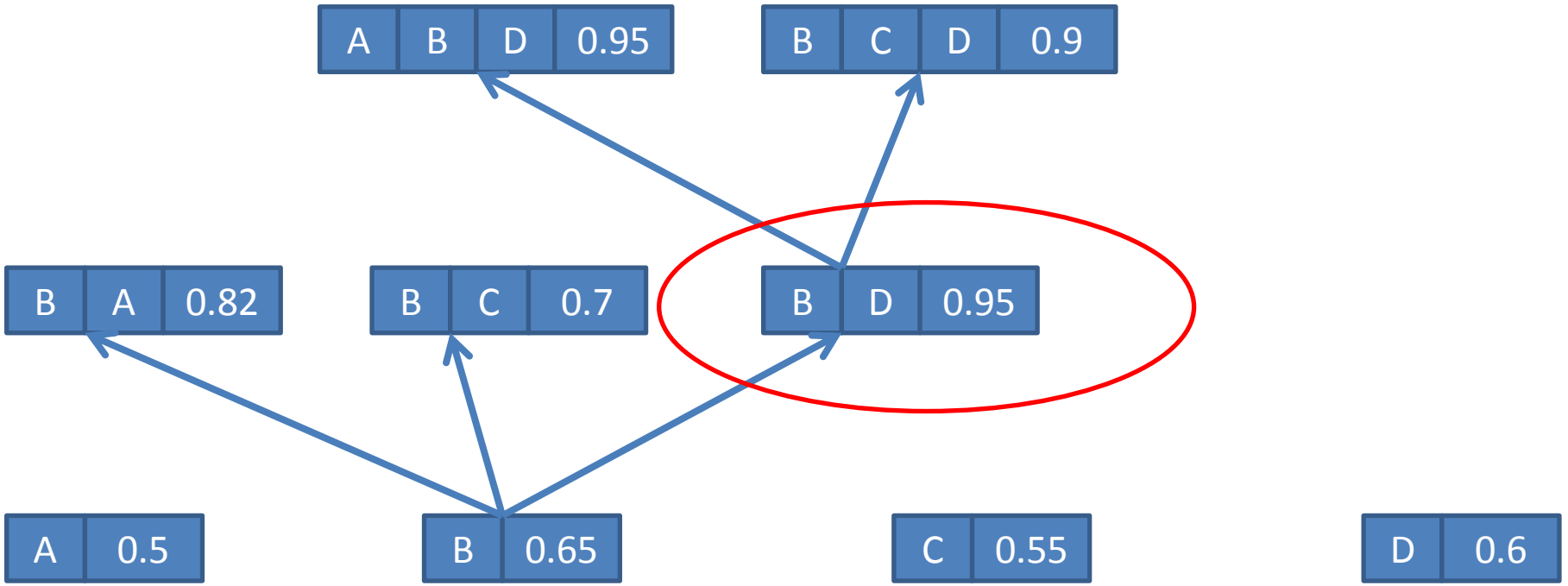
Чтобы не перебирать такое большое число моделей, используют различные методы и эвристики:

1) жадные алгоритмы;
2) генетические алгоритмы поиска;
3) …

Жадный алгоритм №1

Можно так: Фиксируем небольшое число N, перебираем все комбинации по N признаков, выбираем лучшую комбинацию, потом перебираем комбинации из N+1 признаков так, что предыдущая лучшая комбинация признаков зафиксирована, а перебирается только новый признак. Таким образом можно перебирать, пока не упремся в максимально допустимое число признаков или пока качество модели не перестанет значимо расти.

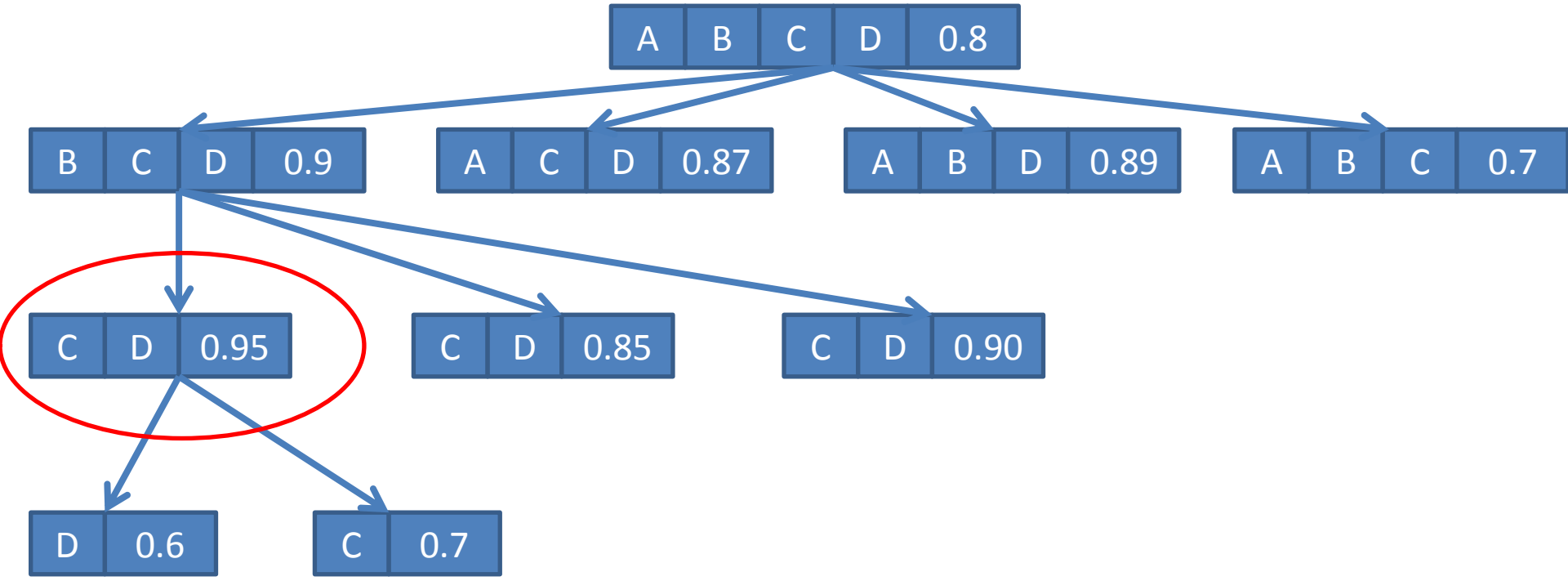
Жадный алгоритм №1



Жадный алгоритм №2

Последний алгоритм можно развернуть: начинать с полного пространства признаков и выкидывать признаки по одному, пока это не портит качество модели или пока не достигнуто желаемое число признаков.

Жадный алгоритм №2



Генетический алгоритм при отборе признаков

Имеем всего m нецелевых признаков. Любой результат отбора признаков можно представить в виде вектора длины m и состоящего из 0 и 1.

Например, вектор (1,1,0,0,1) соответствует отбору первого, второго и пятого признака.

Для каждого такого вектора *v* можно вычислить точность (качество) *f(v)* модели МО.

Генетический алгоритм при отборе признаков

Возникает биологическая интерпретация: вектора=особи, *f(v)*=функция приспособленности, цисла в векторе=гены.

Работа генетического алгоритма:

1. Сгенерировать случайным образом k особей, вычислить функцию приспособленности каждой особи. Далее пункты 2-6 делаются в цикле.

Генетический алгоритм при отборе признаков

2. Выбрать (с некоторым вероятностным распределением) пару особей для размножения.
3. С помощью смешения генотипов родителей создать двоих потомков и вычислить для потомков функции приспособленности.
5. (Мутация) С вероятностью *e* наступает следующее событие: в произвольной особи один ген мутирует.
6. Две наименее приспособленные особи погибают и удаляются из популяции.

Комментарии к работе генетического алгоритма

Выбор особей для размножения, как правило, пропорционален их функции приспособленности.

Генотипы родителей можно смешивать, например, так:

(1,1,0,0,1) – родитель 1

(0,0,1,1,1) – родитель 2

Случайным образом генерируем подмножество индексов *I*:

(1,1,0,0,1) – родитель 1

(0,0,1,1,1) – родитель 2

которое определяет обмен родительских генов :

(0,1,0,1,1) – потомок 1

(1,0,1,0,1) – потомок 2

Мутация означает изменение некоторого гена на противоположный :

(1,1,0,0,1) -> (1,1,1,0,1)

Генетический алгоритм работает, пока не стабилизируется функция приспособленности самой лучшей особи.

Синтез новых признаков

Зачем это делать?

1. Из нескольких плохих признаков можно состряпать один хороший.
2. Улучшение работы моделей МО.

Методы получения новых признаков

1. Нормализация (приведение признаков к одному масштабу). Без этого метрические методы МО работают плохо.
2. Логарифмирование. Для борьбы с большими числами и получения нормального распределения значений признака.
3. Житейская логика. Например, если мы предсказываем анорексию у девушек из Playboy, то значимым тут признаком является «индекс массы тела», а не «рост» и «вес» по отдельности.

Методы получения новых признаков

Выделение признаков для картинок, текстов, видео – это отдельная тема.

Про преобразование категориальных признаков см. [2]

Метод главных компонент (PCA)

Общий смысл метода

Пусть объекты обладают *m* нецелевыми признаками. Требуется сжать информацию, задаваемую этими признаками, в *k (k<m)* новых признаков.

Рассмотрим самый простейший случай, даны объекты из таблицы:

Очевидно, что один признак тут лишний -

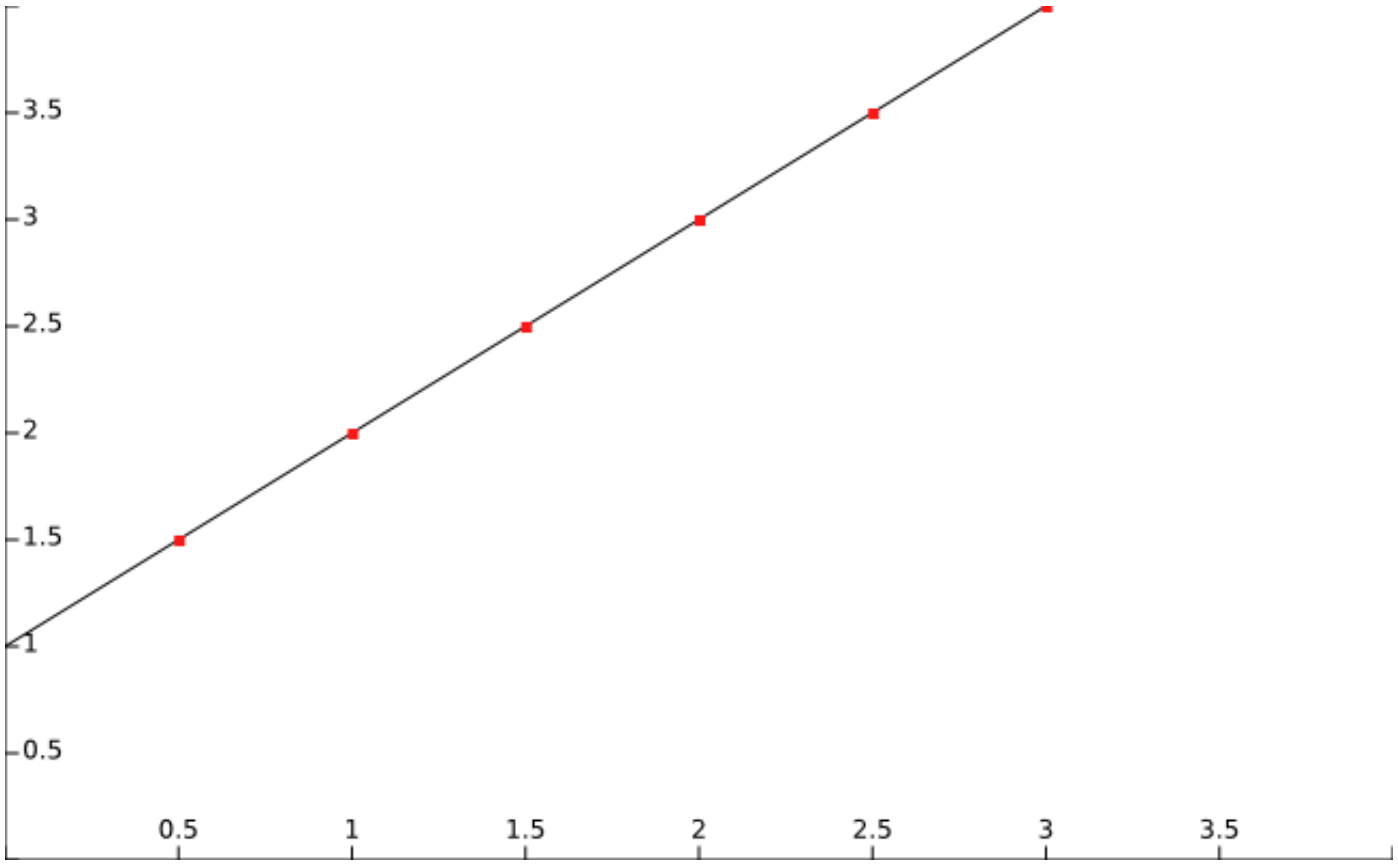
его можно выпилить!

Во-вторых, оставшийся признак можно преобразовать так, что его отклонение будут минимальным.

А как это сделать?

	P1	P2	Y
A	0.5	1.5	1
B	1	2	1
C	1.5	2.5	0
D	2	3	0
E	2.5	3.5	0
F	3	4	1

Нанесем объекты из примера на плоскость



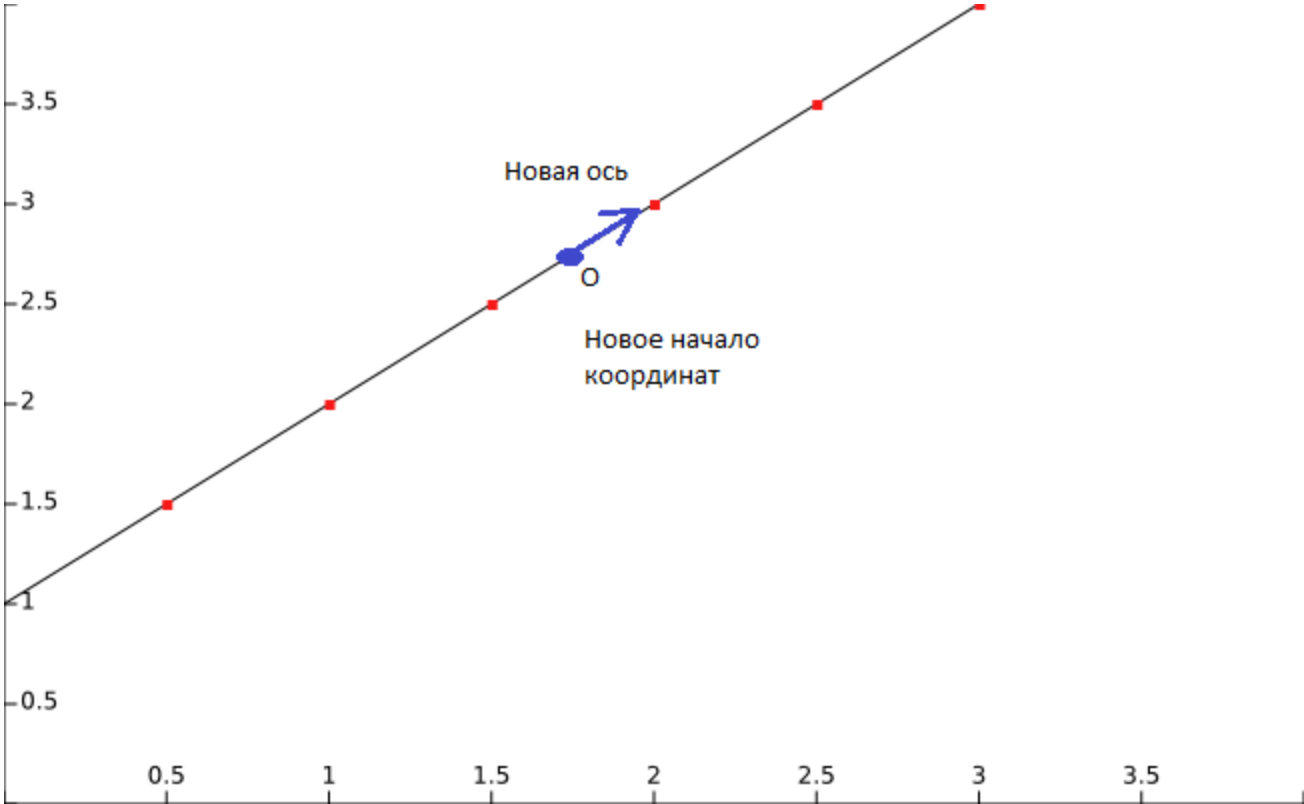
Переход в пространство меньшей размерности

Мы видим, что мы можем работать на этой прямой, вместо целой плоскости.

Чтобы отклонение нового признака было минимальным, на прямой нужно поставить точку отсчета так, чтобы суммарное отклонение объектов от точки отсчета было минимальным.

Новая ось координат и новое начало координат определяют значения нового признака: «координата объекта в новой системе координат»

Начало отсчета имеет координаты (1.75,2.75)



Новая таблица

Теперь вместо старых признаков P1,P2 мы пишем значение нового признака R1=«координаты объекта в новой системе координат с началом в (1.75,2.75)».

БЫЛО:

	P1	P2	Y
A	0.5	1.5	1
B	1	2	1
C	1.5	2.5	0
D	2	3	0
E	2.5	3.5	0
F	3	4	1

СТАЛО:

	R1	Y
A	-1.75	1
B	-1.05	1
C	-0.35	0
D	0.35	0
E	1.05	0
F	1.75	1

Случай двух признаков

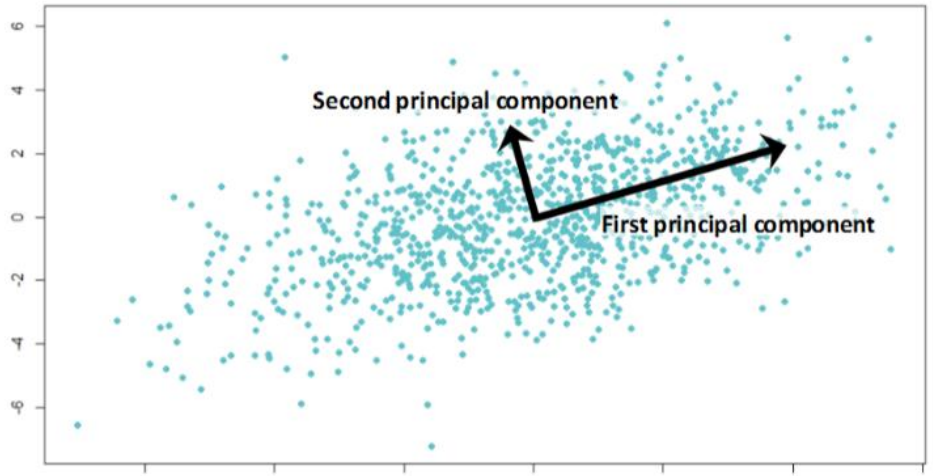
Разобранные пример был тривиален – данные сами «ложились» в подпространство меньшей размерности.

А как быть в общем случае?

Когда объекты уже не ложатся

на прямую?

Намёк на картинке:

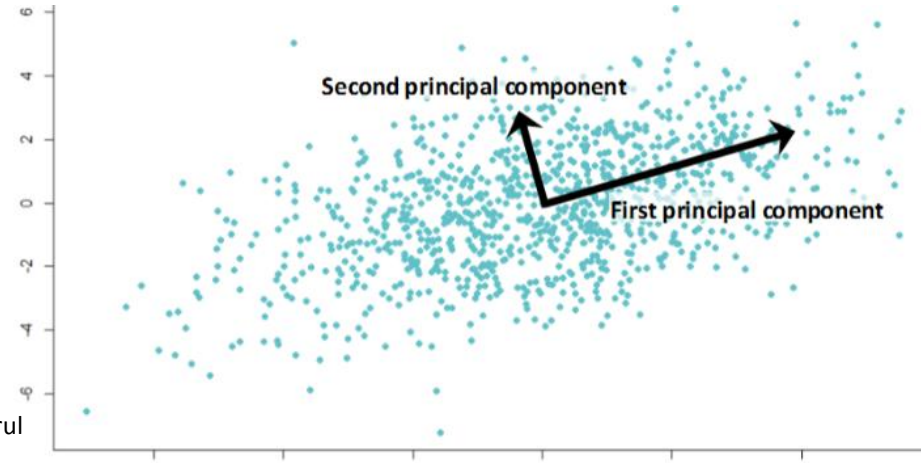


https://www.linkedin.com/pulse/dos-donts-principal-component-analysis-syed-sadat

Случай двух признаков

Если «старых признаков» 2шт. (как на картинке), то:

1) Найти новое начало координат по правилу: сумма отклонений всех объектов до этой точки минимальна.
2) Найти первую ось координат по правилу: сумма отклонений всех объектов до этой прямой минимальна.
3) Вторая ось вычисляется автоматически – из условия перпендикулярности к первой оси.



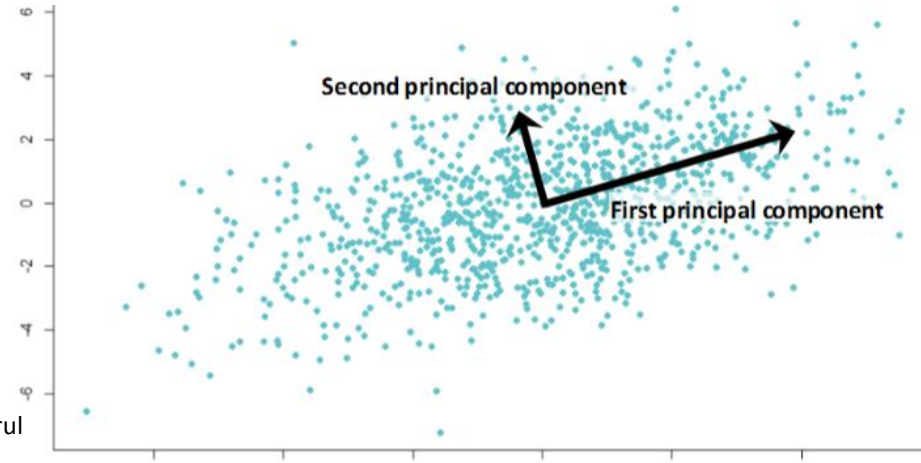
https://www.linkedin.com/pulse/dos-donts-principal-component-analysis-syed-sadat-nazrul

Если признаков >2

… то поступаем аналогично:

1) Найти новое начало координат по правилу: сумма отклонений всех объектов до этой точки минимальна.
2) Найти первую ось координат по правилу: сумма отклонений всех объектов до этой прямой минимальна.
3) Вычислить гиперплоскость, перпендикулярную оси из п.1). Далее процесс аналогично продолжается внутри этой гиперплоскости.

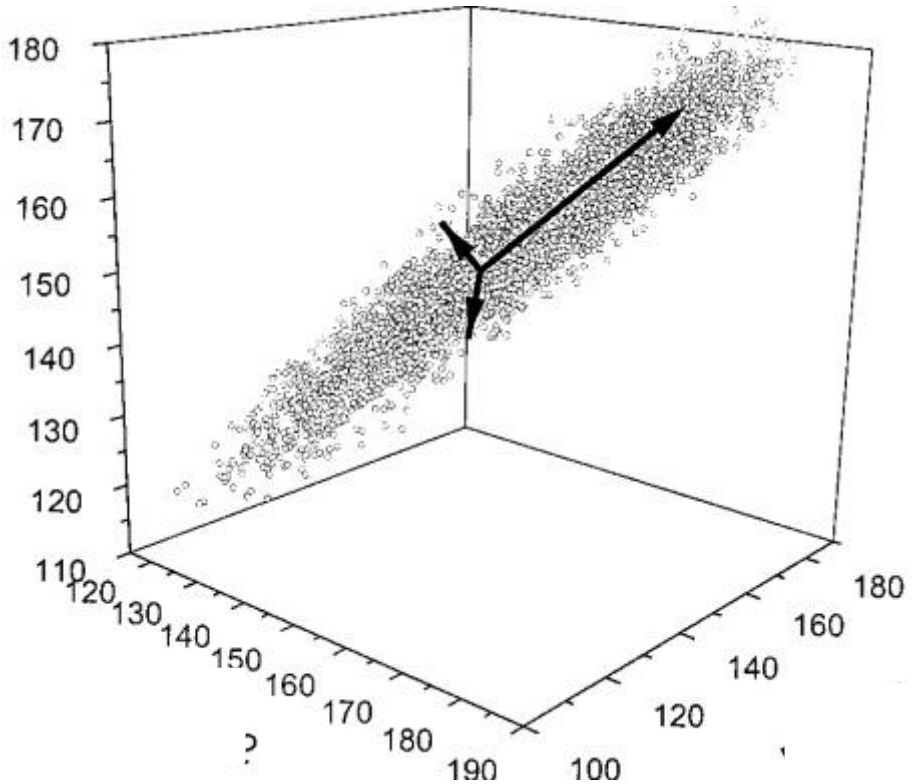
Что на выходе?



https://www.linkedin.com/pulse/dos-donts-principal-component-analysis-syed-sadat-nazrul

Если признаков >2

На выходе вы получите m осей (главных компонент). Если планируется оставить лишь k признаков, то нужно взять значения первых k главных компонент – это и будут новые признаки..



https://www.osapublishing.org/abstract.cfm?uri=ao-41-2-320

Синтез новых объектов

Зачем это делать?

Если тренировочная выборка объектов несбалансирована (то есть доля объектов одного класса гораздо больше доли объектов второго класса), то могут возникнуть проблемы.

Например, такая: алгоритм предсказания просто забудет про меньший класс и все объекты будет относить к большему классу.

С этим нужно что-то делать!!!

Методы балансировки выборки

1. Удаление выбросов – они тоже мешают работе алгоритмов предсказания.
2. Undersampling – удаление объектов большего класса. Объекты большего класса можно кластеризовать, а потом из каждого кластера оставить лишь эталонные объекты (объекты из середины кластера).
3. Oversampling – размножение (клонирование) объектов меньшего класса.
4. Создание синтетических объектов (см. след. слайд)

Синтетические объекты (SMOTE-алгоритм)

По паре объектов A,B можно построить синтетический объект как их линейную комбинацию aA+(1-a)B, где a – случайное число из отрезка [0,1].
Например, при a=0.1 объекты

Объект	Рост	Вес	Пол (Y)
A	200	100	1
B	150	50	0

Дают синтетический объект

Объект	Рост	Вес	Пол (Y)
C	155	55	0.1

Проблемка: категориальные признаки

Категориальные признаки при вычислении лин. комбинации могут потерять смысл. Например, пол=0.1
Возможные пути решения:
1.

Объект	Рост	Вес	Пол (Y)
C	155	55	0.1

Проблемка: категориальные признаки

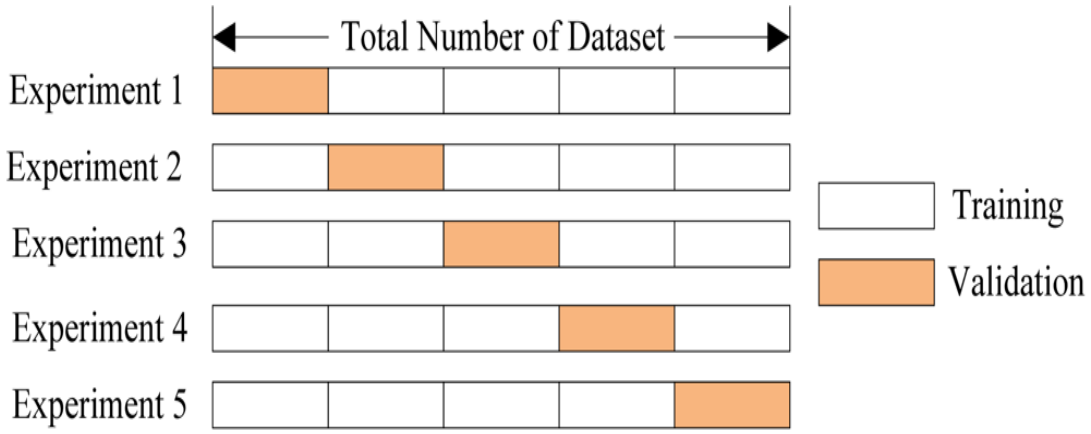
Категориальные признаки при вычислении лин. комбинации могут потерять смысл. Например, пол=0.1
Возможные пути решения:

1. Округлить до ближайшего допустимого значения.
2. Провести случайное испытание в соответствии с полученной вероятностью.
3. Сделать признак числовым. Если это делается для целевого признака, то задача классификации превращается в задачу регрессии.

Объект	Рост	Вес	Пол (Y)
C	155	55	0.1

Не забывайте про еще одно правило:

Если выборка все-таки несбалансирована, то нужно соблюдать пропорции классов при разбиении на тестовую и тренировочную выборку. При кросс-валидации пропорция должна соблюдаться для каждого фолда.



https://habrahabr.ru/company/ods/blog/328372/

Объекты с разной историей

Очень часто бывает, что большинство объектов имеет маленькую (неинтересную) историю, а объектов с большой историей мало.

Философская проблема в том, что и объект с малой историей и объект с большой историей с точки зрения МО равноправны (как 2 разных строки в таблице).

Как придать больший вес объектам второго типа?

Расщепление

Объект с длинной историей можно расщепить на несколько других объектов.

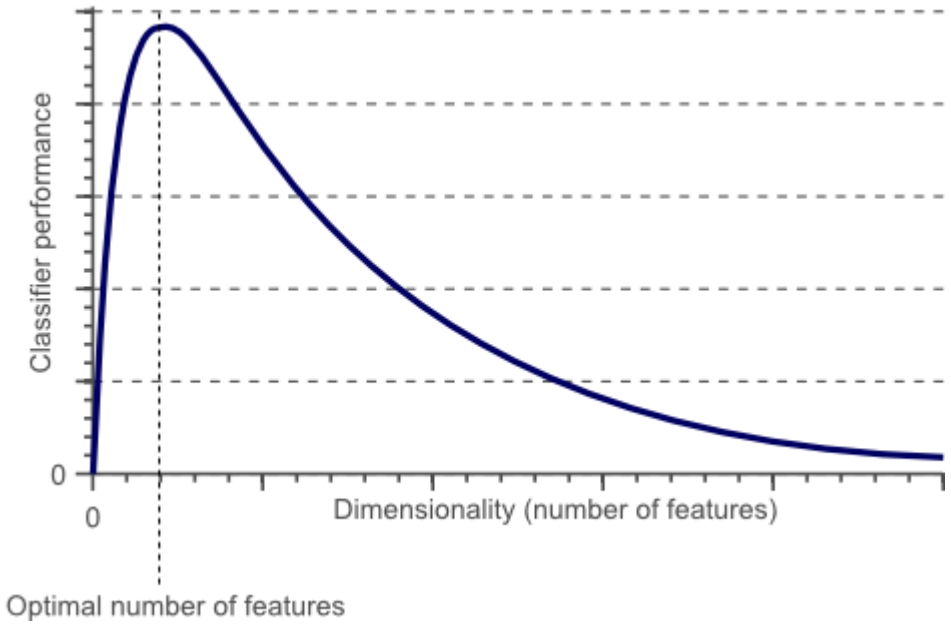
	01.01	02.01	03.01	04.01	05.01	06.01	07.01	08.01	09.01
duration	100	200	300	400	500	600	700	800	900

Объект	dur0	dur-1	dur-2	dur-3	dur-4
A	500	400	300	200	100
B	600	500	400	300	200
C	700	600	500	400	300
D	800	700	600	500	400
E	900	800	700	600	500

Проклятье размерности

Когда много признаков…

Если признаков очень много, то (воз)можно заметить такой эффект:



Причем этот эффект можно наблюдать и для независимых признаков!

https://www.datasciencecentral.com/profiles/blogs/about-the-curse-of-dimensionality

А кого касается проклятие размерности?

Проклятию размерности в первую очередь подвержены метрические алгоритмы.

Значит, в многомерном пространстве начинаются неполадки с метрикой.

Как выглядит распределение расстояний между объектами при увеличении размерности пространства?

Есть объяснение (не знаю, насколько убедительное), использующее закон больших чисел.

Другое объяснение см. на следующих слайдах.

Почему возникает проклятие?

Объяснение 1: В многомерном пространстве сложно сформировать репрезентативную тренировочную выборку.

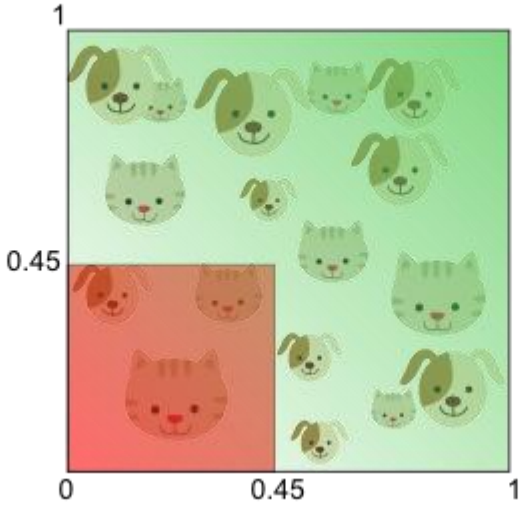
Мы знаем, что тренировочная выборка должна быть репрезентативной – это необходимо, чтобы модель могла обучиться.

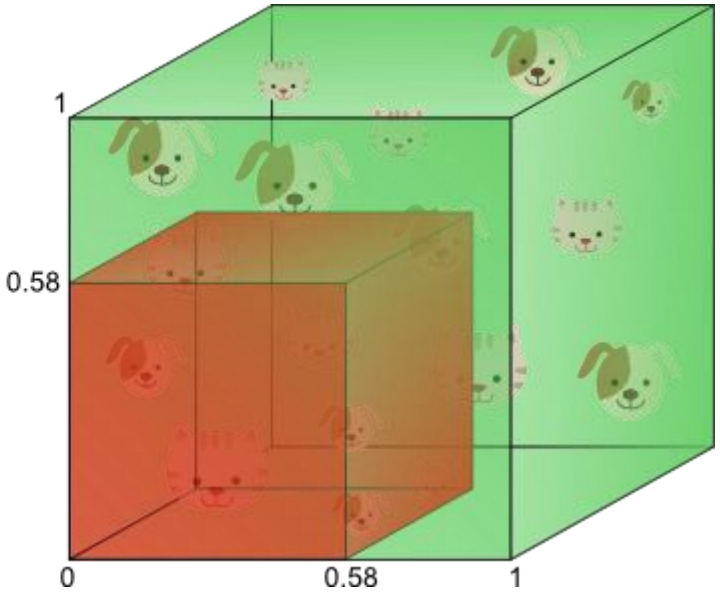
Предположим, что в нашей задаче зависимость обнаруживается, если тренировочная выборка составляет 20% от множества всех возможных объектов.

Почему возникает проклятие?

Когда пространство двумерное, то 20% от всех объектов получаются как содержимое квадрата со стороной 0.45 (0.45*0.45=0.2). А в трехмерном пространстве уже требуется брать куб со стороной 0.58.

То есть с ростом размерности тренировочная выборка должна расти экспоненциально!





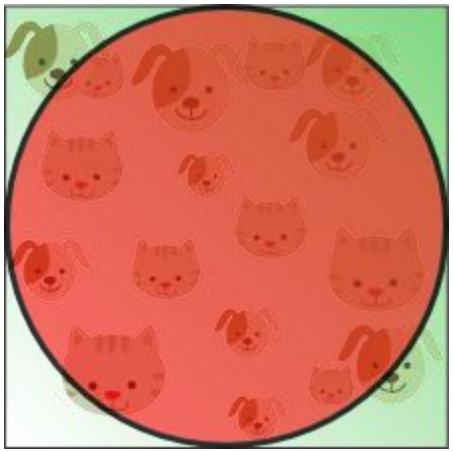
https://www.datasciencecentral.com/profiles/blogs/about-the-curse-of-dimensionality

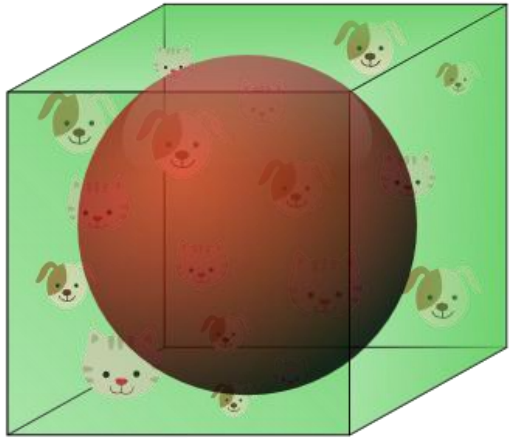
Почему возникает проклятие?

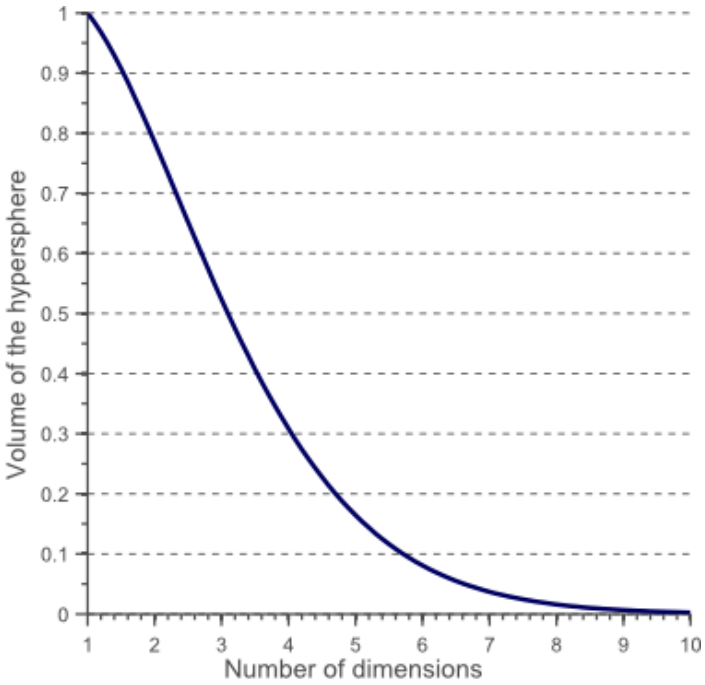
Объяснение 2: В многомерном пространстве объекты располагаются по периферии.

Будем считать долю пространства, находящегося в максимальной сфере с центром в начале координат.

Мера такого множества стремится к 0 с ростом размерности.





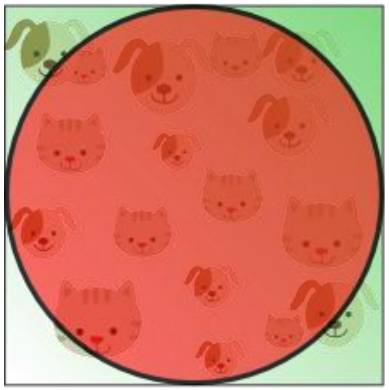


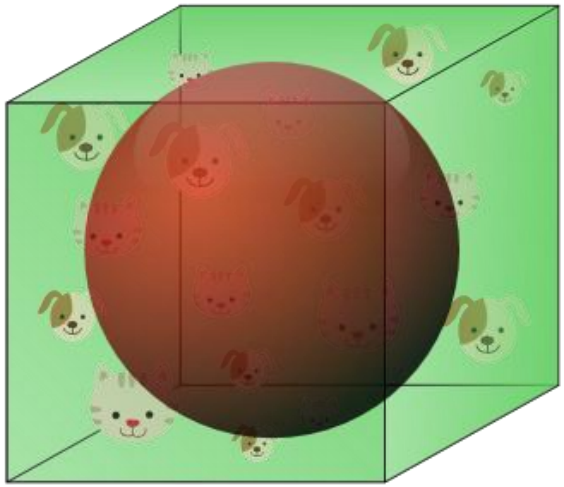
https://www.datasciencecentral.com/profiles/blogs/about-the-curse-of-dimensionality

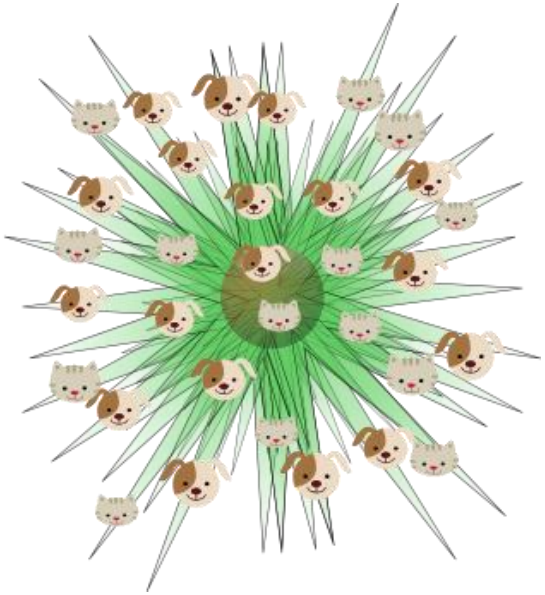
Почему возникает проклятие?

Поэтому в многомерном пространстве почти все объекты находятся на периферии.

Если мы в центр пространства поместим гипотетический объект, то расстояние от него до остальных будет примерно одинаковым. Величина расстояния становится малоинформативной.







https://www.datasciencecentral.com/profiles/blogs/about-the-curse-of-dimensionality

Использованная литература

1. https://habrahabr.ru/company/ods/blog/325422/
2. https://alexanderdyakonov.wordpress.com/2016/08/03/python-категориальные-признаки/
3. https://ru.wikipedia.org/wiki/Логнормальное_распределение
4. https://habrahabr.ru/post/264915/ (про энтропию при отборе фич)
5. https://habrahabr.ru/post/270367/ (зачем распределения признаков делать нормальными)
6. https://www.datasciencecentral.com/profiles/blogs/about-the-curse-of-dimensionality (про проклятье размерности)

Приложение: стекинг

Стекинг

Это один из самых популярных (и общих) способов построения ансамблей алгоритмов, т.е. использования нескольких алгоритмов для решения одной задачи машинного обучения. Известно, что если обучить несколько разных алгоритмов, то в задаче регрессии их среднее, а в задаче классификации — голосование по большинству, часто превосходят по качеству все эти алгоритмы.

Стекинг

Возникает вопрос: а почему, собственно, мы обязаны использовать либо усреднение либо голосование по большинству?

Можно же ансамблирование доверить очередному алгоритму (т.н. «метаалгоритму») машинного обучения!

Зафиксирует терминологию:

базовые алгоритмы – выдают ответы (метапризнаки), которые используются для настройки метаалгоритма.

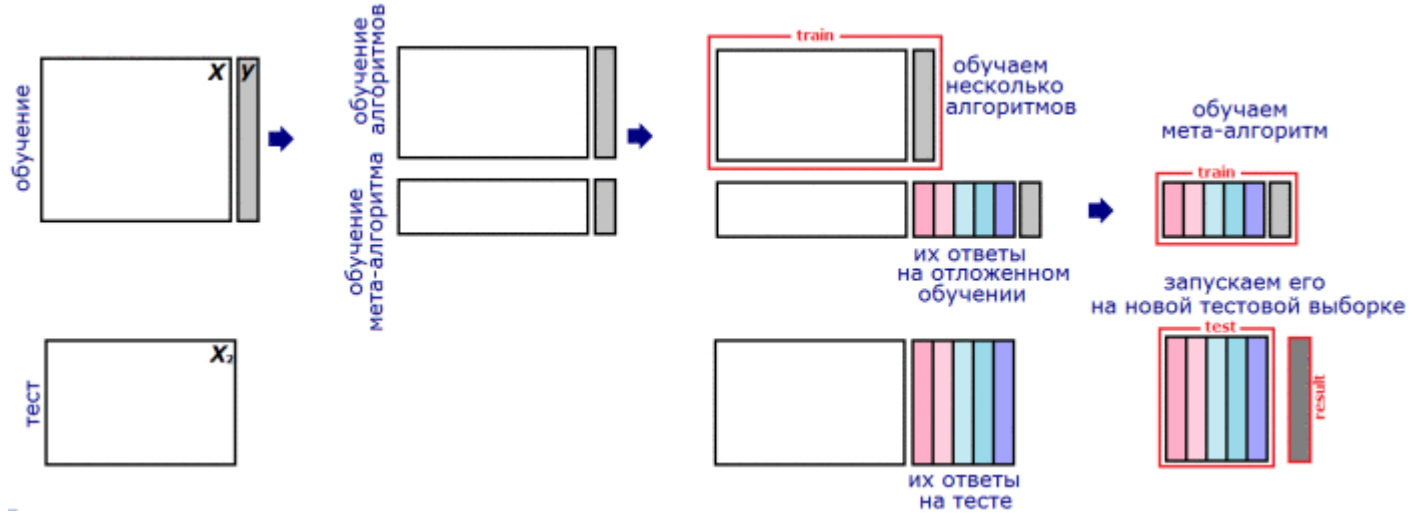
метаалгоритм – использует лишь ответы базовых алгоритмов и выдает окончательный ответ.

Простейшая схема стекинга

Тренировочную выборку делят на две части. На первой обучают базовые алгоритмы. Затем получают их ответы на второй части и на тестовой выборке.

Ответ каждого базового алгоритма – это новый признак (метапризнак). На метапризнаках второй части обучения настраивают метаалгоритм. Затем запускают его на метапризнаках теста и получают ответ.

Простейшая схема стекинга



Зачем еще разбивать тренировочную выборку?

Если этого не делать, то будет переобучение, поскольку в каждом метапризнаке будет «зашита» информация о значении всего целевого вектора.