

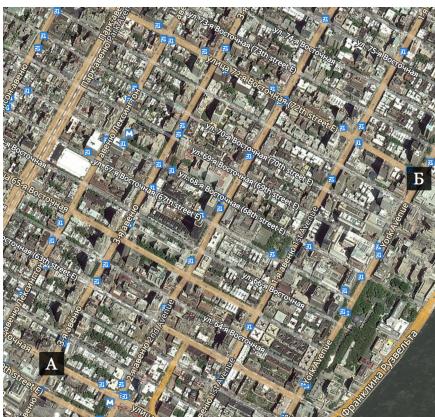
Семинар 2-3: сегментация клиентов и кластеризация

Задача 1

- a) У нас есть точки A(1, 1), B(2, 2), C(3, 0). Нарисуйте их на плоскости и посчитайте между ними расстояние Евклида, Манхэттенское и Чебышева:

$$\begin{aligned}\rho_1(A, B) &= \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} \\ \rho_2(A, B) &= |x_B - x_A| + |y_B - y_A| \\ \rho_3(A, B) &= \max(|x_B - x_A|, |y_B - y_A|)\end{aligned}\quad (1)$$

- б) Какое расстояние будете использовать для того, чтобы добраться из точки А в точку Б? Почему?



- в) Какое расстояние вы бы использовали для измерения похожести текстов или генома?

В тексте была сделана опечатка

В тексте была сделано очепятка

CTGGG**CTA**AAA**GGT**CCCTTAGCC..TTTAGAAAAA.GGCCATTAGG**AA**TTGC
CTGGG**ACT**AAA....CCTTAGC**T**ATTTAC**AAAAA**TGGCCATTAGG...TTGC

Решение:

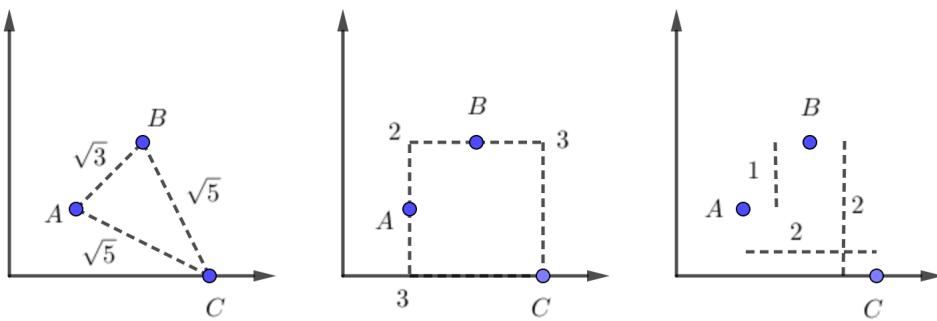
Начнём с расстояний. Для примера посчитаем все три между точками А и В:

$$\begin{aligned}\rho_1(A, B) &= \sqrt{(2-1)^2 + (2-1)^2} = \sqrt{2} \\ \rho_2(A, B) &= |2-1| + |2-1| = 1+1 = 2 \\ \rho_3(A, B) &= \max(|2-1|, |2-1|) = \max(1, 1) = 1\end{aligned}\quad (2)$$

По аналогии посчитаем все расстояния и выпишем результаты в табличку:

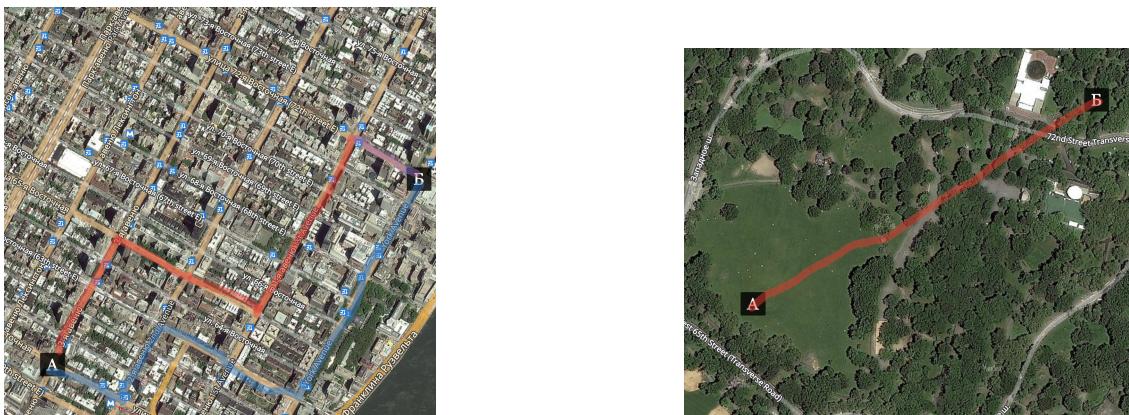
	Евклидово	Манхеттенское	Чебышева
AB	$\sqrt{2}$	2	1
AC	$\sqrt{5}$	3	2
BC	$\sqrt{5}$	3	2

Изобразим все три ситуации на декартовой плоскости. В каждой ситуации расстояния считаются «по пунктиру». Для расстояния Евклида по диагонали, для Манхеттенского расстояния сначала по одной оси, затем по другой, для расстояния Чебышёва берётся только одна координата - та, по которой дальше всего идти.



Для случая а) мы находимся в городе. Мы не можем идти напролом через здания, нам приходится идти по улицам. Логичным будет использовать манхеттенское расстояние. Обратите внимание, что оно обладает интересным свойством: неважно где сворачивать. Идя ко красной дороге, мы пройдём ровно столько же, если бы пошли по синей.

В случае б) можно использовать расстояние Евклида, так как мы идём по полю да по лесу.



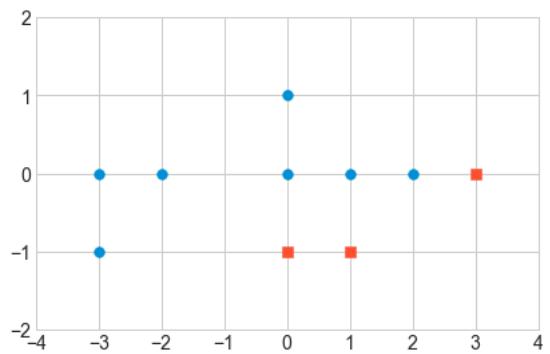
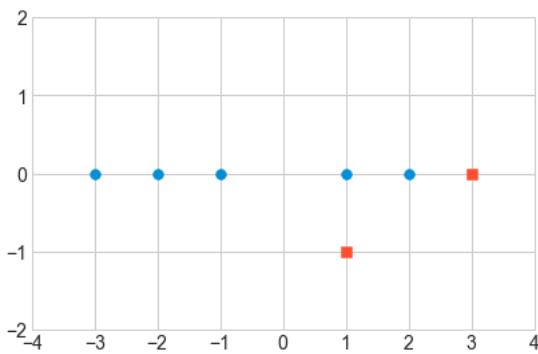
Для того, чтобы понять насколько похожи тексты, можно использовать расстояние Левенштейна. Это минимальное количество операций вставки одного символа, удаления одного символа и замены одного символа на другой, необходимых для превращения одной строки в другую.

Ровно его же можно использовать для последовательностей нуклеотидов. В нашем случае зелёным показано сколько надо нуклеотидов заменить, а красным сколько надо вставить. Расстояние левенштейна между представленными в примере текстами: 2 (буквы о и ч). Между последовательностями: 14.

Задача 2

Жокей Святополк решил открыть несколько новых ларьков с шаурмой¹. Перед открытием он подумал о потенциальных покупателях и выяснил, где на районе находятся общежития. На картинках ниже они отмечены синими точками. Святополк понимает, что все общежития, расположенные в районе, можно сегментировать по их географическому положению и, исходя из этого, расположить палатки с шаурмой. Сделать это ему хотелось бы с помощью алгоритма K-means:

1. Ставим ларьки с шаурмой в случайных местах;
2. Смотрим в какой кому ближе идти;
3. Двигаем ларьки ближе к центрам их популярности;
4. Снова смотрим и двигаем;
5. Повторяем так много раз, пока алгоритм не сойдётся и движение не прекратится.



Красными точками отмечены стартовые точки для палаток. В первом районе Святополк ставит две палатки, во втором районе три палатки. Помогите Святополку с сегментацией! Сколько итераций понадобилось сделать до полной сходимости алгоритма? Сколько объектов вошли в каждый из кластеров?

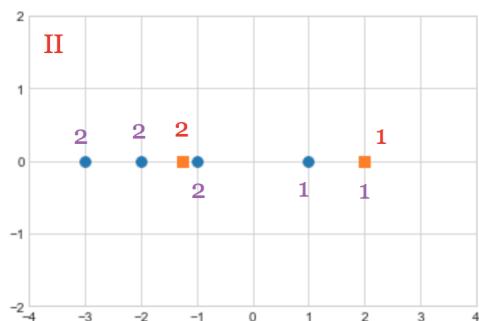
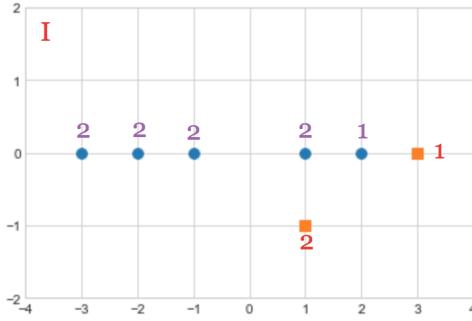
- Используйте для кластеризации Евклидово расстояние.
- Используйте для кластеризации Манхэттенское расстояние.
- В этой задаче мы сами предложили вам для кластеризации начальные точки (красные квадраты). На практике начальное приближение центроидов обычно генерирует компьютер. Изменится ли разбиение на кластеры, если изменить стартовые точки?

Решение:

- Первый случай элементарный. Все расстояния видны на глазок. Даже ничего считать не надо. В ходе первой итерации сразу понятно, что все левые точки отходят второму кластеру и только одна первому. Центр первого кластера переезжает в своего единственного последователя. Центр второго кластера найдём как средневзвешенное координат его последователей:

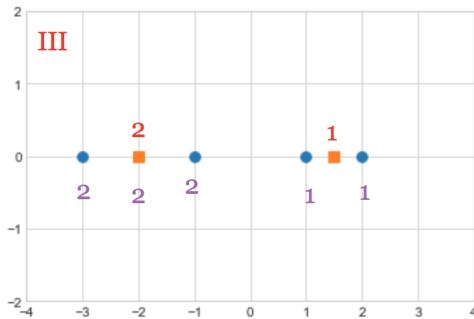
¹По мотивам https://vas3k.ru/blog/machine_learning/

$$\begin{aligned}x_2 &= \frac{1}{4} \cdot (-3 - 2 - 1 + 1) = -\frac{5}{4} \\y_2 &= \frac{1}{4} \cdot (0 + 0 + 0 + 0) = 0\end{aligned}\tag{3}$$



На второй итерации последователи появляются у обоих ларьков. Первый центр оказывается ровно между своими двумя последователями в точке $(0, 1.5)$. Второй центр пересчитаем как средневзвешеной троих последователей:

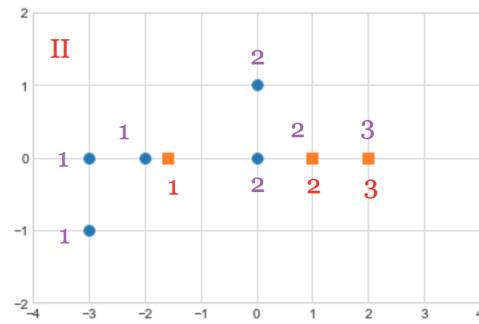
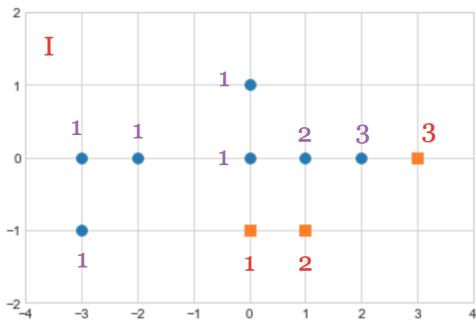
$$\begin{aligned}x_2 &= \frac{1}{3} \cdot (-3 - 2 - 1) = -2 \\y_2 &= \frac{1}{3} \cdot (0 + 0 + 0) = 0\end{aligned}\tag{4}$$



После второй итерации центры больше не обмениваются последователями. Это означает, что алгоритм сошёлся и мы нашли оптимальные точки для ларьков. Первый находится между двумя правыми общагами, второй в одной из левых общаг. Кому-то повезло.

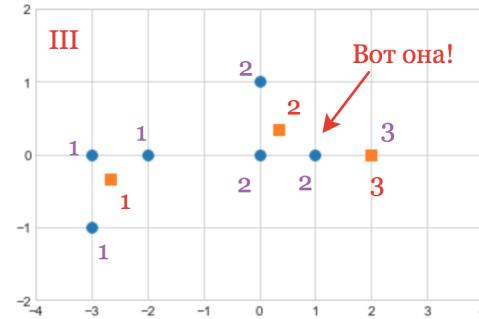
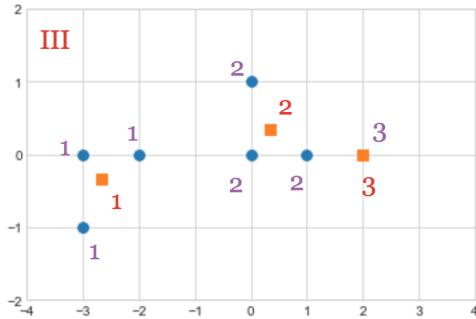
Проделаем всё то же самое со второй ситуацией. На первом шаге второй и третий ларёк переезжают в конкретные общаги. Для первого ларька по-честному нужно посчитать координаты нового центра:

$$\begin{aligned}x_1 &= \frac{1}{5} \cdot (0 + 0 - 2 - 3 - 3) = -\frac{8}{5} \\y_1 &= \frac{1}{5} \cdot (1 - 1 + 0 + 0 + 0) = 0\end{aligned}\tag{5}$$



На второй итерации часть последователей перебегает от первого ларька ко второму. Для третьего ларька ничего не меняется. Снова пересчитываем координаты новых центров:

$$\begin{aligned}
 x_1 &= \frac{1}{3} \cdot (-2 - 3 - 3) = -\frac{8}{3} \\
 y_1 &= \frac{1}{3} \cdot (0 + 0 - 1) = -\frac{1}{3} \\
 x_2 &= \frac{1}{3} \cdot (0 + 0 + 1) = \frac{1}{3} \\
 y_2 &= \frac{1}{3} \cdot (0 + 0 + 1) = \frac{1}{3}
 \end{aligned} \tag{6}$$



После второй итерации вроде как всё стабилизируется. Есть только одна подозрительная точка, отмеченная на рисунке надписью, которую сложно не заметить. Давайте по-честному посчитаем расстояние от неё до центров второго и третьего кластеров:

$$\begin{aligned}
 \rho_3 &= \sqrt{(1 - 2)^2 + (0 - 0)^2} = 1 \\
 \rho_2 &= \sqrt{(1 - 1/3)^2 + (0 - 1/3)^2} \approx 0.74
 \end{aligned} \tag{7}$$

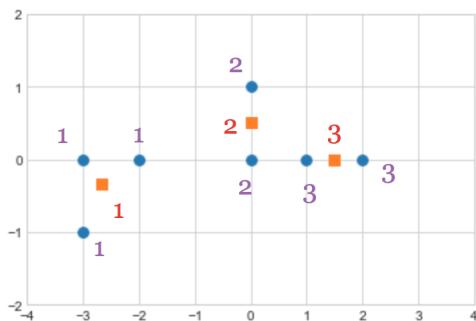
Оказывается, что она ближе ко второму ларьку. На этом, после двух итераций алгоритм прекращает работу.

- б) В первой ситуации результат будет ровно таким же. Мы с какого-то момента начинаем считать расстояния только по оси x. Из-за этого результат выходит одинаковым.

Во второй ситуации первые две итерации будут ровно такими же. В самом конце неожиданно окажется, что спорная точка действительно является спорной.

$$\begin{aligned}
 \rho_3 &= |1 - 2| + |0 - 0| = 1 \\
 \rho_2 &= |1 - 1/3| + |0 - 1/3| = 1
 \end{aligned} \tag{8}$$

Не очень понятно, к какому кластеру её отнести. Можно либо остановить на этом алгоритм, либо сделать ещё один шаг и тогда кластеры окончательно зафиксируются.



- в) В первом случае нет. Кластеры выделены очень хорошо. Во втором случае да. Если взять в качестве начальных точек $(0, -1)$, $(0, 2.1)$, $(3, 0)$, то в ходе Евклидового $k - \text{means}$ мы прийдём к такой же ситуации, как в случае Манхэттенской метрики в конце прошлого пункта.

Задача 3

Начальник Аристарх был в командировке. Там он услышал про иерархическую агломеративную кластеризацию. По приезду, находясь в состоянии восторга, он записал в свой блокнот следующие четыре наблюдения:

n	x	z
1	8	6
2	6	10
3	2	4
4	4	2

После он отдал блокнот маркетологу Савелию. Аристарх хочет, чтобы Савелий провел агломеративную иерархическую кластеризацию. На совещании было решено использовать в качестве расстояния между объектами обычное Евклидово расстояние. Расстояние между кластерами решено определять по принципу дальнего соседа. Помогите Савелию с агломеративной иерархической кластеризацией. И не забудьте нарисовать дендрограмму. Начальники любят красивые картинки.

Решение:

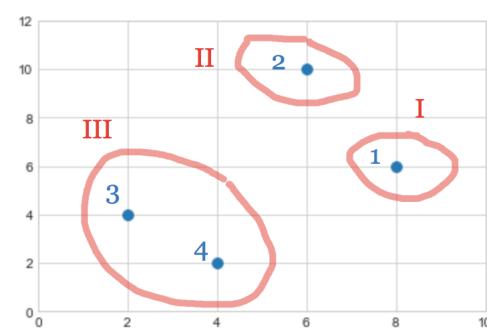
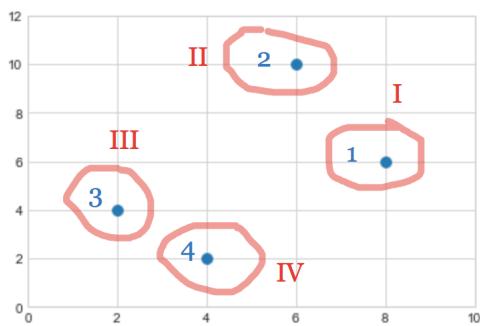
Вспомним алгоритм агломерационной иерархической кластеризации:

1. Начинаем с того, что высыпаем на каждую точку свой кластер
2. Сортируем попарные расстояния между центрами кластеров по возрастанию
3. Берём пару ближайших кластеров, склеиваем их в один и пересчитываем центр кластера
4. Повторяем п. 2 и 3 до тех пор, пока все данные не склеятся в один кластер

Для удобства сразу же посчитаем расстояния между всеми точками и занесём его в табличку:

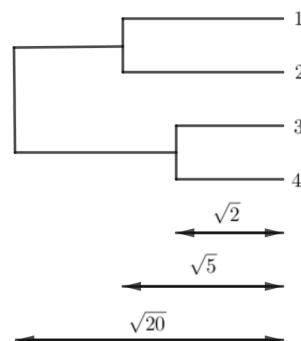
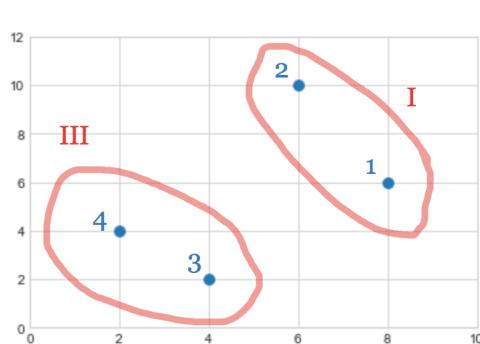
12	$\sqrt{5}$
13	$\sqrt{10}$
14	$\sqrt{8}$
23	$\sqrt{13}$
24	$\sqrt{20}$
34	$\sqrt{2}$

На первой итерации каждое наблюдение это отдельный кластер. Самое короткое расстояние между 3 и 4 наблюдениями. Сольём их в единый кластер.



Кластера IV больше нет. Теперь нам надо посчитать расстояние между оставшимися кластерами. Расстояние от I до II соответствует $\sqrt{5}$, как и раньше. Расстояние от III до I надо выбрать по принципу дальнего соседа. Третье наблюдение дальше от первого, чем четвёртое. Выходит, что расстояние между III и I равно $\sqrt{10}$. Четвёртое наблюдение дальше от второго, чем третье. Расстояние между III и II равно $\sqrt{20}$. Сливаем между собой кластеры I и II.

В игре остаются кластеры I и III. Давайте выясним какое между ними расстояние. Оно, по-прежнему, формируется по принципу дальнего соседа. Среди четырёх расстояний: 13, 14, 23, 24 нам нужно выбрать самое большое. Это расстояние от второй точки до четвёртой, $\sqrt{20}$.



Теперь мы можем построить дендрограмму. Сначала соединились 3 и 4 наблюдения, расстояние между ними составило $\sqrt{2}$. После 1 и 2 с расстоянием между ними $\sqrt{5}$. После слились два кластера с расстоянием между ними $\sqrt{20}$. На дендрограмме длина линии до точки слияния соответствует расстоянию между кластерами.

Задача 4

Маркетолог с аналитическим складом ума Оля (она кстати говоря ещё и фрилансер) занимается заказом от туристической фирмы. Ей нужно сделать с помощью методов машинного обучения сегментацию клиентов. На этапе предобработки фичей Оля столкнулась с двумя проблемами: категориальной переменной x , в которой записаны курорты и текстовой переменной z , в которой дан отзыв о курорте:

n	x	z
1	Испания	Нежился на пляже
2	Крым	Копали яму на пляже
3	Дача	Копал картошку
4	Крым	Ел картошки и картошку

- а) Что такое категориальная переменная? Почему её надо как-то предобрабатывать?
- б) Почему нельзя сделать замену Крым = 1, Дача = 2, Испания = 3? Что такое ОНЕ-кодирование? Как будет выглядеть наша табличка с переменными после ОНЕ?
- в) Почему нельзя сделать ОНЕ для текстовой переменной?
- г) Какие этапы предобработки тестов вы знаете? На самом деле вы их скорее всего не знаете и мы их сейчас обсудим.
- д) Сделайте для корпуса текстов из задачки tf-idf.

Решение:

- а) Категориальная переменная - это переменная, которая принимает значения из некоторого конечного множества. В данном случае переменная x принимает значение из множества курортов. Модели обучаются на каких-то числовых данных. Наша категориальная переменная не числовая. Чтобы обучить модель, мы должны переработать её в числовой вид.
- б) Нельзя сделать замену Крым = 1, Дача = 2, Испания = 3, потому что в таком случае модель подумает, что на множестве курортов есть отношение порядка, то есть, что Крым лучше Дачи, а Дача лучше Испании. Это далеко не так. Для каждого человека на множестве курортов свой субъективный порядок. Модель же должна быть объективной.

Для того, чтобы нормально преобразовать курорты в цифры, обычно используют one hot encoding (одно горячее кодирование))00))0).

Для каждой категории вводится своя дамми-переменная, принимающая значение 1, если человек был на этом курорте и значение 0, если не был. Наша табличка преображается. Одна категориальная переменная превращается в несколько бинарных.

Обратите внимание, что полученная таблица избыточна.

n	x_{isp}	x_{kr}	x_{da}
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

Легко можно избавиться от одной из переменных и не потерять при этом информацию. Если мы выбросим переменную x_{da} , мы увидим в оставшихся двух переменных нули, сразу же поймём, что человек был на даче.

Дополнительный материал для саморазвития: лучше выбросить одну избыточную переменную, иначе можно попасть в не очень хорошую ситуацию, которая называется **даммиловушкой**. Предположим, что мы оцениваем модель линейной регрессии:

$$y_i = \beta_0 + \beta_1 \cdot x_{isp} + \text{beta}_2 \cdot x_{kr} + \beta_3 \cdot x_{da}.$$

В такой ситуации мы нарываемся на линейную зависимость. Константа равна сумме дамм переменных. Мы попадаем в ситуацию мультиколлинеарности. Если мы будем учить модель без регуляризатора, у нас ничего не выйдет. **Конец дополнительного материала для саморазвития.**

- в) Потому что практически каждый текст уникален и наше ОНЕ скорее всего приведёт к появлению $n - 1$ дополнительной переменной. На будет недостаточно n наблюдений, чтобы оценить адекватную модель. Приходится искать более изящные пути работы с текстами.
- г) Для текстов существует несколько этапов предобработки. Обычно это токенизация, нормализация (стемминг или лемматизация), очистка от стоп-слов. Подробнее об этом написано в юпитерском блокноте для текущего семинара.
- д) Разобьём все тексты на слова, очистим их от стоп-слов и нормальзуем. После, если в наблюдении встречается данное слово, будем ставить в табличке количество упоминаний, а если нет 0.

	нежиться	пляж	копать	яма	картошка	есть
1	1	1	0	0	0	0
2	0	1	1	1	0	0
3	0	0	1	0	1	0
4	0	0	0	0	2	1

При достаточно большом числе текстов в корпусе, такую табличку можно использовать для обучения. Однако можно построить чуть более умные переменные. Для этого используется подход tf-idf. Предпосылки подхода:

- Порядок слов неважен.
- Если слово встречается в документе часто и оно не является стоп-словом, скорее всего, оно важное. Эту тенденцию отражает показатель tf (сокращение от английского term frequency, частота слова). Чтобы получить tf, нам просто нужно нормализовать матрицу, полученную выше на размер словаря.

- Если слово встречается в других документах реже, чем в данном, то, скорее всего, оно также важное, так как описывает специфику документа и отличает его от других. Эту тенденцию отражает показатель idf (сокращение от inverse document frequency, обратная частота документа). Обычно idf расчитывают как:

$$\ln \left(\frac{N}{n_i} \right),$$

где N – объём словаря, n_i – количество документов, в которых встретилось слово i .

Логарифм берётся, чтобы сгладить очень большие числа. Перемножив tf и idf мы учтём оба факта важности слова и получим tf-idf представление текста.

Построим матрицу для tf . В этом случае нормализация идёт по строкам:

	нежиться	пляж	копать	яма	картошка	есть
1	1/6	1/6	0	0	0	0
2	0	1/6	1/6	1/6	0	0
3	0	0	1/6	0	1/6	0
4	0	0	0	0	2/6	1/6

Теперь построим idf . В случае idf мы работаем со столбцами:

	нежиться	пляж	копать	яма	картошка	есть
1	$\ln 4$	$\ln 2$	0	0	0	0
2	0	$\ln 2$	$\ln 2$	$\ln 4$	0	0
3	0	0	$\ln 2$	0	$\ln 2$	0
4	0	0	0	0	$\ln 2$	$\ln 4$

Перемножаем эти две таблички и получаем матрицу из tf-idf , которую можно использовать для обучения модели.

	нежиться	пляж	копать	яма	картошка	есть
1	0.23	0.11	0	0	0	0
2	0	0.11	0.11	0.23	0	0
3	0	0	0.11	0	0.11	0
4	0	0	0	0	0.23	0.23

Если хочется, можно посчитать среднее idf для каждой переменной и отсортировать их в порядке важности. Это позволит оставить в модели, конструируемой по корпусу текстов, только самые важные фичи.

Ещё задачи!

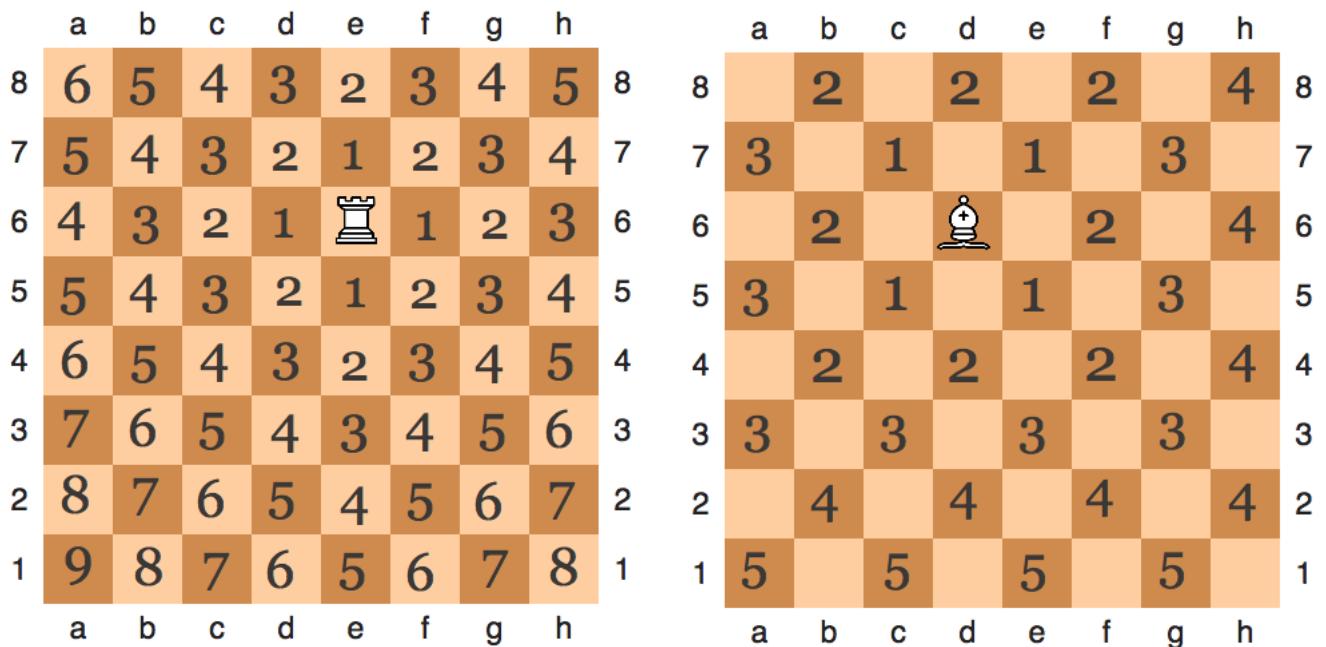
Задача 5

Представьте себе, что вы шахматная фигура. Какое расстояние вы бы использовали, чтобы измерить насколько далёкий путь вам предстоит пройти по шахматной доске. Придумайте метрику для каждой шахматной фигуры: ладья, слон, ферзь, король, слон, конь. Попытайтесь формализовать эту метрику в виде формулы.

Подсказка: попробуйте нарисовать шахматную доску на бумажке, поставить в какую-нибудь клетку фигуру, а после для каждой клетки выписать цифру: сколько до этой клетки идти фигуре. Это поможет придумать для формулы общий вид. Клетку $a1$ в координатной сетке примите за $(0, 0)$.

Решение:

Будем измерять расстояние в клетках, которые были пройдены фигурой. Другой подход – измерять расстояние в ходах. Это не очень коректно, так как в таком случае будет непонятно какая у фигур сокрость.



Расстояние, которое преодолевает ладья будет измеряться с помощью манхэттенской метрики.

$$\rho(x, y) = |x_2 - x_1| + |y_2 - y_1|$$

Будем считать, что у клетки $a1$ координаты $(0, 0)$. Тогда, если ладья хочет попаст из клетки $e6$ в клетку $b1$, то есть из $(4, 5)$ в $(1, 0)$, ей надо будет пройти $|1 - 4| + |0 - 5| = 3 + 5 = 8$ клеток. Обратите внимание, что все точки вокруг ладьи по мере отдаленя от неё выстраиваются в концентрические ромбики с одинаковыми расстояниями.

Для слона ситуация будет аналогичной. Для него используется манхэттенское расстояние, но

доска для него повёрнута под углом 45 градусов.

	a	b	c	d	e	f	g	h		a	b	c	d	e	f	g	h		
8	5	4	3	2	2	2	2	2	8	8	2	3	2	3	2	3	2	3	8
7	5	4	3	2	1	1	1	2	7	7	3	4	1	2	1	4	3	2	7
6	5	4	3	2	1	†	1	2	6	6	2	1	2	3	2	1	2	3	6
5	5	4	3	2	1	1	1	2	5	5	3	2	3	3	2	3	2	5	5
4	5	4	3	2	2	2	2	2	4	4	2	1	2	3	2	1	2	3	4
3	5	4	3	3	3	3	3	3	3	3	3	4	1	2	1	4	3	2	3
2	5	4	4	4	4	4	4	4	2	2	2	3	2	3	2	3	2	3	2
1	5	5	5	5	5	5	5	5	1	1	3	2	3	2	3	2	3	4	1
	a	b	c	d	e	f	g	h		a	b	c	d	e	f	g	h		

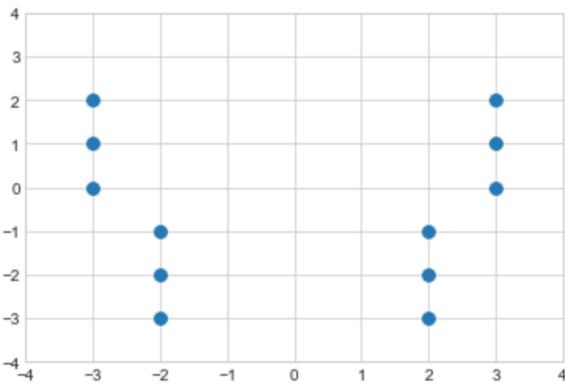
Для короля и ферзя ситуация будет немного другой. Из-за того, что они могутходить и по вертикали и по диагонали, их расстояние считается по формуле:

$$\rho(x, y) = \max(|x_2 - x_1|, |y_2 - y_1|).$$

Тогда, если король хочет попасть из клетки f6 в клетку b1, то есть из (5, 5) в (1, 0), он должен пройти $\max(|1-5|, |0-5|) = \max(4, 5) = 5$. По сравнению с ладьёй, у короля есть преимущество. Для него клетки выстраиваются в концентрические квадраты. Диагонали для него в одношаговой доступности, когда для ладьи они в двухшаговой доступности. Расстояние, описанное выше называется расстоянием Чебышёва.

Для коня ситуация оказывается самой загадочной. Из-за того, что он ходит буквой Г, расстояния, рассчитываемые для него оказываются нелинейными. Их нельзя формализовать в виде простой формулы, но можно нарисовать. Концентрические линии для коня оказываются совсем чудными. Клетка, до которой идти 4 хода с точки зрения привычных для нас расстояний, оказывается к коню ближе, чем клетка, до которой идти 3 хода.

Задача 6

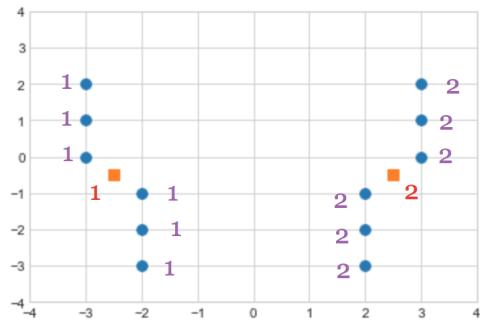
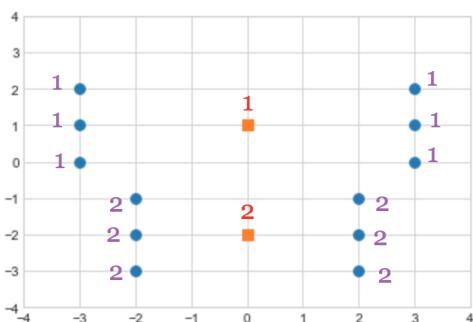


- Примените метод K-means с $K = 2$, $K = 3$, $K = 4$ и $K = 5$. Начальные точки каждый раз выбирайте случайно. Спишитесь с другими людьми и выясните какие точки они выбрали для инициализации. Для каких K у вас получились одинаковые результаты? Почему?
- Правда ли, что для всех рассмотренных K оба метода разбивают выборку на одинаковые кластеры? Всегда ли так происходит? Докажите это или приведите контр-пример.
- В первом пункте вы попробовали провести кластеризацию для разных K . Какое из K является оптимальным? Для того, чтобы определить это используйте сумму квадратов расстояний от точек до центров кластеров.
- Примените метод агломеративной иерархической кластеризации. Нарисуйте дендрограмму. Руководствуясь дендрограммой выберете оптимальное количество кластеров. Обоснуйте свой выбор.

Решение:

К несчастью, у семинариста, который пишет решение, нет друзей. Поэтому ему не с кем свериться. И это печально. В принципе, я детально описал как работает k-means в одной из задач выше, поэтому здесь сразу будут итоговые результаты. В решении не будем дробить задачу на подпункты. Сначала сделаем всё, что идеологически касается k-means, потом займёмся иерархической кластеризацией.

Итоговое решение для $K = 2$ зависит от начальных точек. Вот два примера итоговых положений точек, в которые может прийти алгоритм:



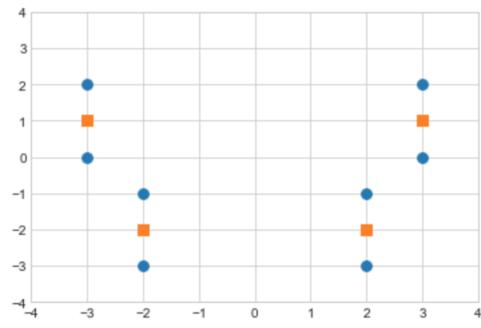
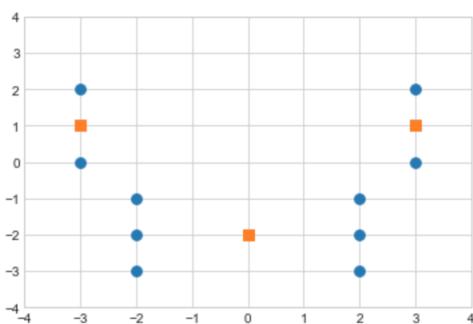
Аналогично можно придумать ситуацию для любого K , чтобы разбиения получились разны-

ми. Так как я пишу решение и могу выбирать любые удобные для меня стартовые точки, я буду их всегда выбирать так, чтобы алгоритм сходился на первой же итерации ².

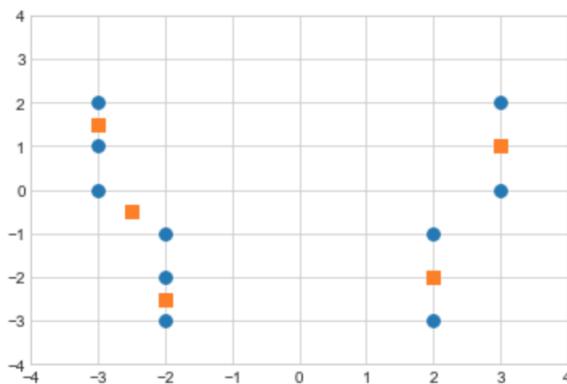
Для $K = 2$ я выбираю правую ситуацию, так как она при случайной инициализации точек более вероятна. Посчитаем для этого случая сумму квадратов расстояний до центра кластера.

$$2 \cdot 2 \cdot ((0.5^2 + 0.5^2) + (0.5^2 + 1.5^2) + (0.5^2 + 2.5^2)) = 38$$

Для $K = 3$ кластеризация скорее всего закончится следующим образом. Для $K = 4$ следущим.

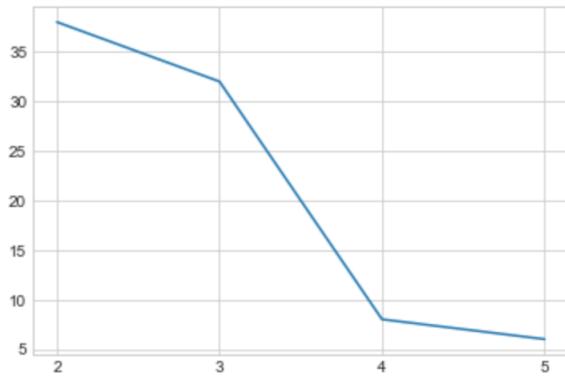


Для $K = 3$ получаем $2 \cdot (1 + 0 + 1 + 4 + 5 + 5) = 32$. Для $K = 4$ получаем $4 \cdot (1 + 0 + 1) = 8$. Для $K = 5$ одним из вариантов может быть ситуация ниже. Для неё суммой квадратов расстояний до центров кластеров будет $4 \cdot 0.5^2 + 2 \cdot (0.5^2 + 0.5^2) + 4 \cdot 1 = 6$.

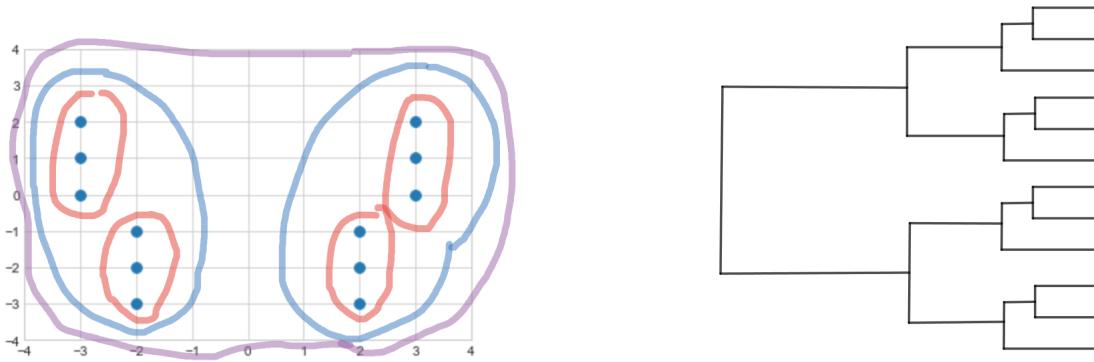


Построим график, где по оси x отложим k , а по оси y сумму квадратов расстояний. По нему видно, что после $k = 4$ расстояние убывает слишком медленно. Поэтому $k = 4$ оптимально.

²Вот хитрюшка то, а!



Теперь займёмся агломеративной кластеризацией. Очевидно, что в течение первых 6 итераций все маленькие кластеры объединяются в 4 кластера, подобных тем, что мы выделили в $k - \text{means}$ при $k = 4$. После левые кластеры сольются вместе, потом правые.

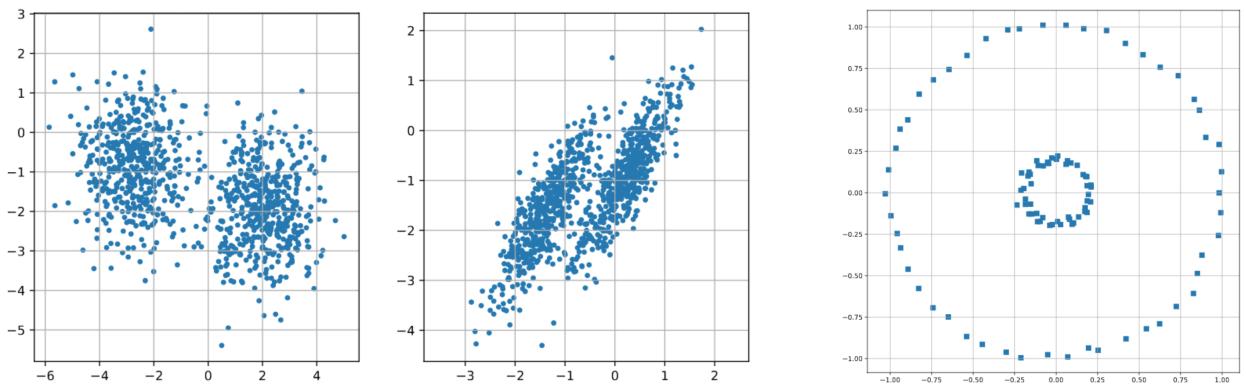


Дендрограмма будет выглядеть соответствующим образом. Сначала внутри маленьких кластеров соединяются две точки. Мы получаем четыре кластера с самыми короткими дистанциями объединения. После ко всем 4 маленьким дендрограммам добавляется ещё одна ветка. Она будет немного подлиннее, так как мы руководствуемся принципом дальнего соседа. Между первыми двумя точками каждой тройки расстояние будет равно единице. Между кластером из двух точек и третьей точкой расстояние окажется равно двум.

После мы соединяем между собой тройки, а потом объединяем всё в общий кластер. Оптимально было бы рубануть всё это дело, когда у нас осталось 4 кластера, так как в дальнейшем ветки становятся слишком длинными, то есть они соединяют слишком отдалённые точки, чего бы нам не хотелось.

Задача 7

Обозначьте расположение центроидов и границ кластеров после применения метода K-means с $K = 2$ на следующих данных:

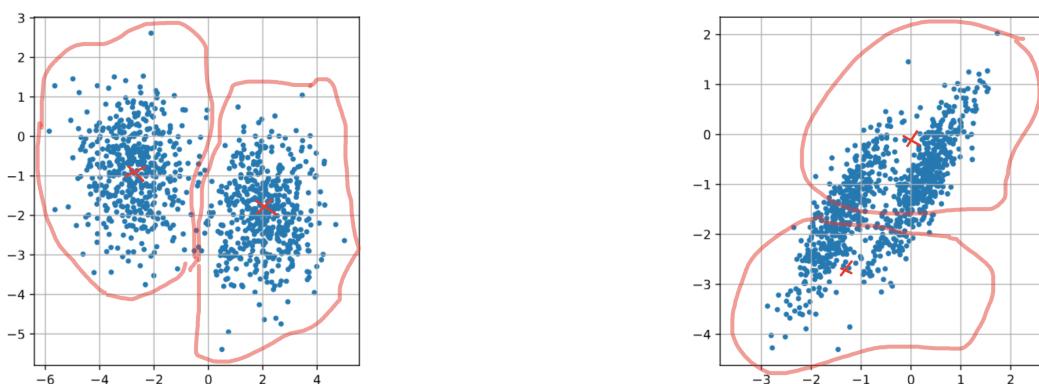


Для каких из этих ситуаций имеет смысл попробовать отличные от $k - \text{means}$ алгоритмы? Если вы подумали, что это изи-задание и на третьей картинке сходу взяли в один кластер внутреннюю окружность, а во второй внешнюю - вы балбес. Это неправильно. Идите и подумайте ещё.

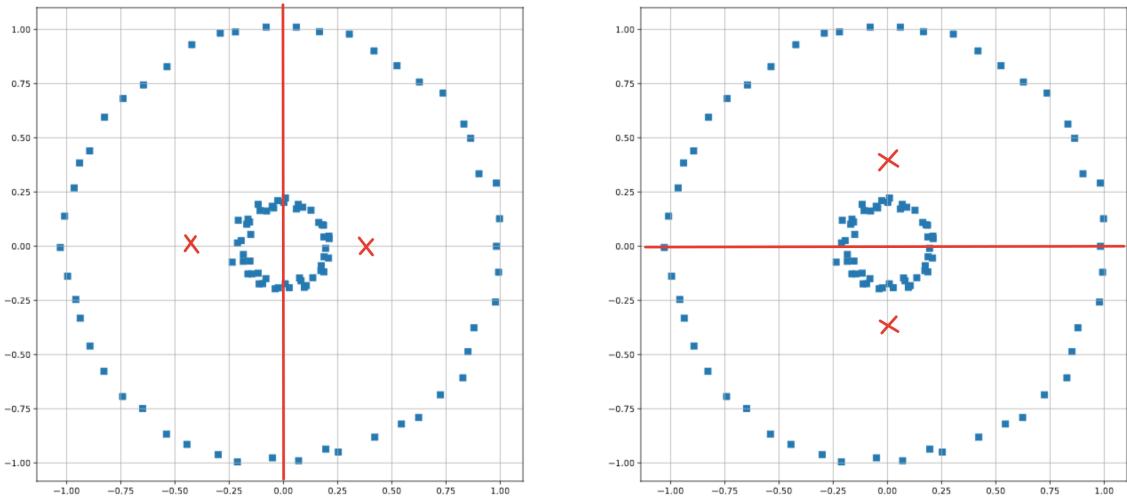
Решение:

В первой ситуации облака выделены довольно чётко. Скорее всего вне зависимости от инициализации, первое облако будет попадать в один кластер, а второе облако во второй. $k - \text{means}$ довольно хорошо работает с подобными выпуклыми кластерами.

Во второй ситуации кластеры расположены слишком близко друг к другу. Итоговый результат снова будет сильно зависеть от инициализации. Возможно, что кластеризация действительно сойдётся к центрам облаков, если перед нами какой-то граничный случай перехода от выпуклых облаков к вытянутым, но скорее всего, кластеризация закончится как-нибудь вот так:



В третьей ситуации ответ с двумя окружностями будет неправильным, потому что у нас считается расстояние от каждой точки до центра кластера. Нельзя выделить один центр так, чтобы до него было ближе всем точкам с внешней окружности, а второй центр так, чтобы до него было ближе всем точкам внутренней окружности. Скорее всего, окружность будет раздроблена на две части одним из множества способов. Как именно — зависит от инициализации. Вот два возможных варианта. Крестами отмечены центры кластеров.



Для второй и третьей ситуаций имеет смысл попробовать отличные от k – means алгоритмы. По идеи, с этими ситуациями хорошо должны справиться DBSCAN или агломеративная иерархическая кластеризация.