Уткин Данил  
23.07.20  
Математика и Python для анализа данных

**Задачи:**

Изучить пункт 2 недели “Линейная алгебра. Векторы”. В пункте изучить следующие темы: “Знакомство с линейной алгеброй”, “Векторные пространства”, “Линейная независимость”, “Операции в векторных пространствах”.  
  
**Тесты:**

Прорешать следующие тесты: ”Базовые понятия линейной алгебры”, “Линейная независимость и размерность”, “Векторные пространства и NumPy”.

Приступить к написанию конспекта по сегодняшним темамю.

**Конспект**

**Знакомство с линейной алгеброй.**

Векторы и матрицы – это ключевые понятия с помощью которых мы будем описывать данные или информацию в машинном обучении.

*Рассмотрим следующий пример задачи анализа данных*. =>

Представьте, что у нас есть сеть магазинов, и для каждого из них нужно предсказать прибыль в следующем месяце. Зачем это нужно? Например, зная, что в одном из магазинов прибыль упадет, мы можем принять меры и избежать этого.

Признаковое описание

В этой задаче магазины являются объектами. Это то, для чего мы делаем предсказание. Компьютер не понимает, что такое магазин. Ему нужно описать каждый объект с помощью чисел. Эти числа называются признаками, характеристиками объекта. Какие признаки могут быть в этой задаче?

X = (50, 47, 52, 55, 5, 1, 2, 55.73, 37.59, 1)

Разумно, если прибыль в следующем месяце зависит от прибыли в предыдущих месяцах. Поэтому давайте сделаем 4 признака, которые описывают прибыль в прошлом месяце, в месяце до него и так далее.

У нас это будут числа 50, 47, 52 и 55.

Далее разумно предположить, что прибыль магазина зависит от того, сколько акций он проводит. Представьте, что магазины торгуют товарами из трех категорий: кондитерские изделия, овощи и фрукты и мясо. Тогда можно для каждой из этих категорий записать, сколько акций планируется в следующем месяце провести.

Это будут числа 5, 1 и 2, то есть 5 акций будут по категории «кондитерские изделия».

Также магазин где-то расположен, и его прибыль зависит от его географической позиции. Поэтому давайте тоже добавим ее в виде широты и долготы.

Наконец, кажется правильным, что прибыль магазина зависит от того, сколько праздничных дней будет в следующем месяце. (последнее число)

Все эти числа образуют набор из 10. Такой набор и

называется **вектором**, то есть это некоторая группа чисел определенного количества.

Другой пример:  
  
X = (14.23, 1.71, 3.43, 5.64, 3.92)

Химические анализы

Представьте, что мы хотим определять для конкретной бутылки вина, из какого сорта винограда она сделана.

Зачем это нужно? Например, мы можем отличать качественные вина от некачественных, те, которые

сделаны из дорогого сорта винограда, от тех, которые сделаны непонятно из чего.

Какие признаки могут быть здесь? Наверное, это результаты химических анализов, например содержания алкоголя или щелочность, или интенсивность света.

Все они образуют **вектор**, на основе которого мы будем делать прогнозы.

Еще пример:

Каждый человек характеризуется своим геномом.

Геном определяет, каким он вырастет, какого цвета волосы у него будут. И именно геном определяет некоторые наследственные заболевания, например рак.

Поэтому мы можем поставить такую задачу: *предсказать по геному человека, заболеет ли он раком в течение 5 лет?*

Заболеет он раком или нет, зависит от мутаций, или отклонений, в его геноме. Всего мутаций могут быть сотни тысяч. Поэтому логично описать каждого конкретного человека вектором, длина которого равна возможному числу мутаций в человеке, и единицы будут стоять в позициях, соответствующих тем мутациям, которые произошли конкретно у него. По этому вектору будет делаться предсказание, заболеет он или нет.

X = (0, 1, 0, 0, 1, …)

А если представить что количество объектов несколько.  
например, несколько магазинов или несколько бутылок вина. В этом случае данные обретают двумерную структуру: у нас есть объекты и есть признаки. Логично составить их в таблицу, в которой каждая строка соответствует одному объекту, то есть содержит все признаки, которые описывает этот объект, а каждый столбец соответствует одному признаку.

Эта структура называется **матрицей**. То есть мы будем понимать матрицу как некоторую *таблицу с двумя измерениями*, в которую записаны некоторые числа.

**Векторные пространства.**

**Векторное пространство** — это множество, которое обладает определенными свойствами. Тогда **вектором** будет называться любой элемент этого пространства.

**Векторное пространство** — это некоторое множество *V*, на котором задано две операции:

1. сложение двух векторов

2. умножение вектора на число

При этом операции должны быть замкнутыми, то есть сумма двух векторов должна давать снова вектор, а умножение любого числа на любой вектор должно давать снова опять же вектор.

При этом сложение и умножение должны удовлетворять ряду аксиом, которые соответствуют нашей интуиции о том, как выглядят сложение и умножение. Дело в том, что элементами векторного пространства могут быть сложные объекты, необязательно числа. И операции могут вводиться довольно сложные. Но при этом интуиция не должна нас подводить, благодаря этим аксиомам.

**Аксиомы:**

**1) ∀ a, b ∈ V, a + b = b + a (коммутативность)**

**2) ∀ a, b, с ∈ V, (a + b) + с = (b + с) + a (ассоциативность)**

**3) ∃ 0 ∈ V, ∀ a ∈ V, a + 0 = 0 + a = 0**

**4) ∀ a ∈ V, ∃ (-a), a + (-a) = (-a) + a = 0**

Если все выше указанные аксиомы были выполнены, то это **абелева группа**

**5) 1 ∈ R, a ∈ V, 1 × a = a**

**6) α, β ∈ R, a ∈ V, (α** **× β) × a = α × (β × a) (ассоциативность относительно умножения чисел)**

**7) α × (a + b) = αa + αb (дистрибутивность умножения относительно сложения векторов)**

**8) (α + β) \* a = αa + βa (дистрибутивность умножения относительно сложения чисел)**

Если все свойства выполняются, то **V** называется линейным пространством, а его элементы векторами.

Мы будем работать только с простыми векторными пространствами, которые называются **евклидовыми**. Это пространства, которые состоят из векторов, каждый элемент которых — это вещественные числа.

**Евклидовы** векторные пространства обозначаются красивой буквой R с верхним индексом n, где n — это число элементов в каждом векторе этого пространства. Их можно визуализировать. Например, R2, то есть все векторы из 2-х элементов, — это точки на плоскости. R3, то есть все векторы из 3-х элементов, — это точки в пространстве.

R2 R3

a a

b R1

R1 R2 b

**R2 R3**

Как вводятся операции в евклидовом пространстве?  
(поэлементно)

**Сложение векторов:**

**a = (α1, α2, α3, …, αn)**

**b = (β 1, β 2, β 3, …, β n)**

**a + b = (α1 + β 1, α2 + β 2, α3  + β 3, …, αn  + β n)**

**Умножение вектора на число:**

**a = (α1, α2, α3, …, αn)**

**α ∈ R, α × a = (α × α1, α × α2, α × α3, …, α × αn)**

Пример:

Мы хотим определить сорт винограда, из которого сделано вино, для каждой бутылки.

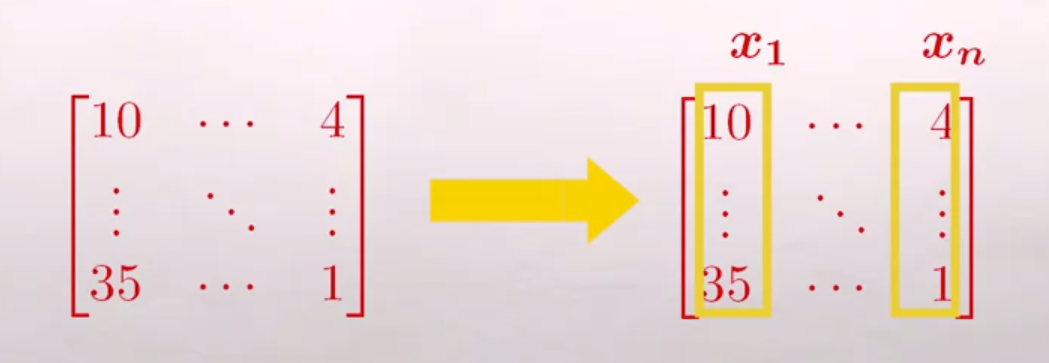
В этом случае векторное пространство — это все возможные описания вина. Все возможные признаковые описания бутылок. При этом конкретный вектор — это описание конкретной бутылки. Эти векторы можно, например, усреднять. То есть, к примеру, мы можем посчитать среднее признаковое описание для конкретного сорта. Для этого уже можно делать некоторый анализ данных.

**Линейная независимость**

Пример:

Допустим, у нас есть 100 магазинов, каждый из которых описывается десятью признаками, десятью числами. Тогда мы можем составить таблицу размера 100 x 10, где каждая строка будет соответствовать признаковому описанию

одного магазина, а каждый столбец — набору значений одного признака на всех магазинах.



Раньше в качестве векторов мы рассматривали строки этой таблицы, то есть каждый вектор являлся признаковым описанием одного магазина. Теперь ситуация будет противоположна, в качестве векторов будем рассматривать столбцы этой таблицы.

То есть один вектор будет содержать значение одного признака на всех магазинах.

x1 = 1000x2

Что, если вектор первого признака равен вектору второго признака, умноженного на 1000? *Это может означать, что и первый, и второй признаки — это суммарный вес товаров, которые завезут в магазин в следующем месяце.*

Но при этом первый признак равен этому весу в *граммах*, а второй — в *килограммах*. Понятно, что это *избыточная информация*. Нам не нужно знать значения этого признака и в одной, и в другой единице измерения. Достаточно знать лишь одно из этого.  
  
Эта ситуация, в которой один вектор выражается как сумма с коэффициентами других векторов, называется **линейной зависимостью**.

**Линейная зависимость** — это довольно плохо. Дело в том, что она приводит к избыточности информации. Мы храним те векторы, которые на самом деле *выражаются через остальные* довольно простым способом. Также мы тратим дополнительные ресурсы на хранение лишней информации.

Тоже не очень хорошо: приводит к лишним денежным затратам.

Также линейная зависимость вредит некоторым методам машинного обучения, например линейным.

Пусть дан набор векторов x1, ..., xn. xi-тая — это один вектор, значение признаков, значение одного признака на всех объектах. Эти векторы **линейно зависимы**, если существуют такие числа β1, ..., βn, что хотя бы одно из них... хотя бы один из них не равен нулю, и при этом сумма векторов с этими коэффициентами равна нулю.

**Система векторов линейно зависимая**, если один из них выражается через все остальные.

Система векторов линейно независимая, если для неё не выполнено определение линейной зависимости. Или, что то же самое, уравнение β1 \* х1 + β2 \* х2 +... + βn \* xn равно нулю только в том случае, если все коэффициенты β1, β2, ..., βn равны нулю. Это и есть опеределение **линейной независимости**.

x1 = (2, 1, 2)

x2 = (4, 2, 4)

x3 = (1, 2, 3)

Пусть дано три вектора из евклидова пространства R3,

то есть векторы длины три. Первый из них имеет вид (2, 1, 2), второй — (4, 2, 4), третий — (1, 2, 3). Если мы возьмём первый вектор, умножим его на два и вычтем второй вектор, то получим нулевой вектор.

2x1 – x2 + 0x3 = 0

Это значит, что эта система векторов линейно зависима. А что будет, если мы заменим первую координату во втором векторе с четвёрки на пятёрку?

Оказывается, система станет линейно независимой.

Методы с помощью, которых можно понять является ли матрицы линейно зависимой или независимой:

1. Метод Гауса

2. Вычислить ранг системы векторов

Пусть дано некоторое векторное пространство *V*. Максимальное число линейно независимых векторов в нём и называется размерностью этого пространства и обозначается как *dim V*. Например, для евклидова пространства Rn размерность равняется как раз таки n, то есть длины вектора в нем.

Dim Rn = n

Представьте себе систему векторов e1, e2, ..., en,

где вектор ei имеет следующий вид: в нём ровно одна координата — i‐тая — равна единице, а все остальные координаты равны нулю. Эти векторы линейно независимы,

но при этом любой другой вектор в Rn выражается линейно через них. Такой набор векторов, через который выражается любой другой вектор, называется базисом векторного пространства.

c1 = (1, 0, …, 0)

c2 = (0, 1, …, 0)

…

cn = (0, 0, …, 1)

**Операции в векторных пространствах**

Двумерный случай

**(1, 2)**

**(-2, 1)**

У нашего вектора можно измерить длину.

Это делается по теореме Пифагора. Нужно возвести в квадрат каждую координату, сложить и извлечь корень.

**Sqrt(1^2 + 2^2) = sqrt(5)**Добавим ещё один вектор с координатами −2 и 1.  
Можно измерить угол между ним и первым вектором с помощью транспортира, например. Он будет равен 90°.

Можно так измерить расстояние между этими двумя векторами. Для этого нужно найти разности по каждой координате, возвести в квадрат, сложить и извлечь корень.

**Sqrt((1 – (-2))^2 + (2 - 1)^2) = sqrt(10)**

**Норма**

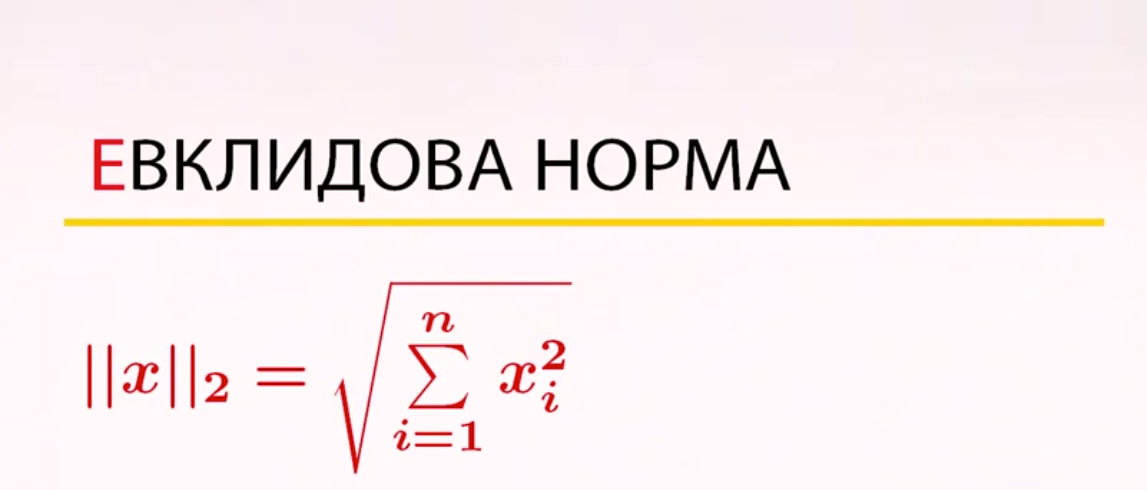
Начнём с длины вектора. Она вычисляется с помощью **нормы**. **Норма** — это некоторая функция от вектора, которая обозначается как x, слева и справа от которого рисуются по две палочки(||x||). Итак, норма — это функция, которая должна удовлетворять трем требованиям:

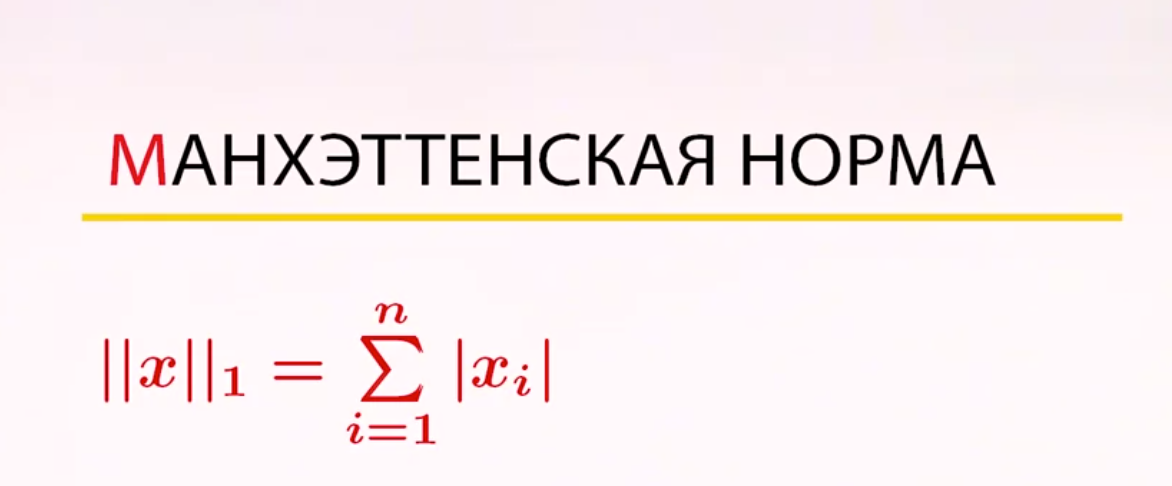
1. Норма должна быть равна нулю тогда и только тогда,   
когда сам вектор равен нулю.  
 **||x|| = 0, то x = 0**2.  Должно иметь место неравенство треугольника. Норма суммы двух векторов должна быть меньше или равна, чем сумма норм двух векторов.

**||x + y|| <= ||x|| + ||y||**3. Норма произведения числа α на вектор x должна равняться произведению модуля α на норму вектора x.

**||** **α**  **× x|| = |α| × ||x||**

Векторное пространство, в котором введено понятие нормы, называется **нормированным пространством**.





**Метрика:**

Далее обобщим понятие расстояния. Для него можно тоже ввести строгое определение, там будет несколько требований, но можно поступить проще. В евклидовом пространстве, зная норму, можно ввести расстояние между векторами x и y. По сути, это будет разность между вектором x и вектором y, у которой вычисляется норма.

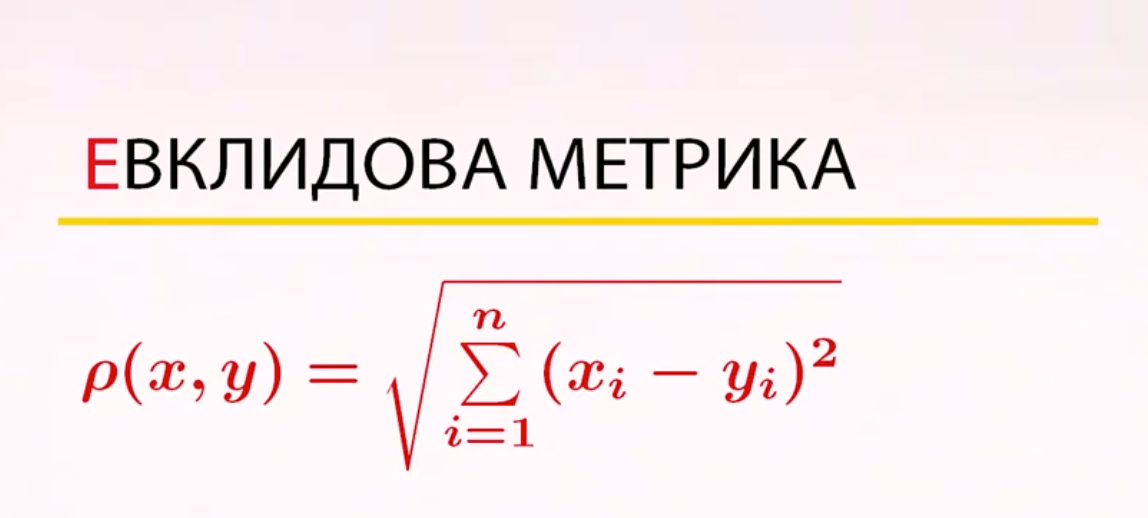
Расстояние иногда ещё называют **метрикой**. По сути, то, как мы вводим **метрику** или расстояние через норму, соответствует геометрическим представлениям.

**(1, 2)**

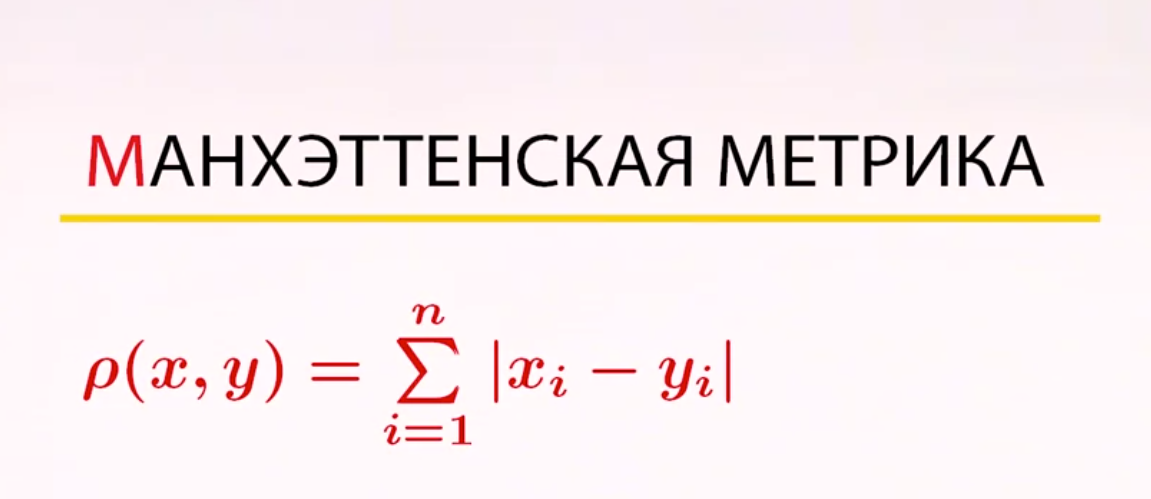
**(-2, 1)**

**P(x, y) = ||x - y||**

Если мы знаем евклидову норму, то можем ввести евклидову метрику. Это будет сумма квадратов разностей координат этих векторов, из которой в конце извлекается корень.



Также можно ввести манхэттенскую метрику. Это сумма модулей отклонений двух векторов.



**Как искать углы**

Итак, чтобы ввести понятие угла, нам понадобится скалярное произведение векторов x и y. В евклидовых пространствах оно вычисляется как сумма произведений координат.

**Норма: ||x|| = sqrt(<x, x>)**

**Расстояние: p(x, y) = ||x - y|| = sqrt(<x – y, x - y>)**

Оказывается, в евклидовых пространствах скалярное произведение x на y можно вычислить как произведение нормы x на норму y и на косинус угла между ними. Отсюда можно получить, что косинус угла между векторам x и y вычисляется как отношение скалярного произведения к произведению их норм.

**Cos(x, y) = <x, y> / ||x|| \* ||y||**

Заметим, что косинус угла — это, по сути, мера сонаправленности векторов. Если векторы параллельны, то косинус угла будет равен 1. Если же векторы

перпендикулярны или ортогональны, как называется в общем случае, то угол между ними равен 90°, а косинус 90° равен 0.

*Есть еще одни файлик ipy, там тоже конспект.(NumPy + векторы и матрицы)*

**Заключение**

Сегодня во время практики я вспомнил начало линейной алгебры первого курса первого семестра. Ознакомился с методами и функциями из библиотеки NumPy, который позволяют работать с векторами и матрицами очень просто.

