**Библиотека NumPy NumPy**

— это базисный пакет для научных вычислений, требующих анализа большого количества данных, на Python. Является он ничем иным как библиотекой для языка программирования Python, которая предоставляет собой объект многомерного массива и различные производные объекты, такие как массивы и маскированные массивы, также NumPy предоставляет набор процедур для быстрых операций с массивами, а также включает в себя математические операции, логические, манипуляции с формами, сортировку, выбор, ввод-вывод данных, дискретные преобразования Фурье, базовую линейную алгебру, статистические операции, случайное моделирование и многое другое. Пакет NumPy был основан на таком объекте как ndarray, Он инкапсулирует n-мерные массивы однородных типов данных, реализует многие операции, которые выполняются в виде скомпилированного кода для улучшения производительности. Если сравнить массивы, которые предоставляет NumPy и стереотипными массивами данных от Python можно заметить достаточное количество немаловажных отличий: – массивы от NumPy имеют стандартизированный и фиксированный размер при создании, в отличие от стандартных списков Python которые входе выполнения программы могут расти динамически, NumPy при изменении размера создаст новый массив и удалит оригинал. – каждый элементы в массиве NumPy должен быть одного типа данных и, следовательно, иметь одинаковый размер в памяти, за исключением тех случаев если помимо массивов NumPy в коде были использованы стандартные массивы Python, что позволит создавать массивы элементов разного размера. – Массивы NumPy могут существенно упростить сложные математические операции и другие типы операций, в которых задействованы большие объемы данных, это позволит выполнить данные операции более эффективно и использовать при этом меньшее количество кода, чем задействовать стандартные последовательности Python. На сегодня большое количество самых разных научных и математических пакетов базируются на основе Python, но в большей степени они используют массивы основывающиеся на принципах NumPy, другими словами, для эффективного использования большого количества современного научного и математического программного обеспечения на основе стандартных пакетов предоставляемых Python недостаточно просто знать, как использовать встроенные типы последовательностей , а необходимо ознакомиться с тем, как использовать матрицы NumPy

В вопросах размера последовательности и скорости особенно важно то с какой скоростью выполняются научные расчеты. В качестве простого примера можно привести вычисление, в котором происходит умножение каждого элемента в одномерной строке на соответствующий элемент в другой строке той же длины. Если данные хранятся в двух стандартных списках Python, мы можем перебирать каждый элемент, это даст правильный ответ, но, если a и b содержат миллионы чисел, то вычисление может занять большое количество времени. NumPy же позволяет существенно уменьшить время затраченное на расчет использовав код на другом языке например С, что устранит все накладные расходы, связанные с интерпретацией кода Python и манипулированием объектами Python, при этом используя все преимущества кодирования на Python. NumPy позволяет выбирать принцип программирования исходя от поставленной задачи используя лучшие стороны разных языков. Библиотека осуществляет поэлементные операции по умолчанию», когда задействован ndarray, поэлементные операции выполняются быстро предварительно скомпилированным кодом

NumPy может осуществлять различные математические преобразования с той же скоростью с которой осуществляет и язык C, но с простотой кода, основанного на Python. Векторизация описывает отсутствие каких-либо явных циклов, индексации и так далее . В коде - эти вещи, конечно же, происходят за кулисами в оптимизированном и предварительно скомпилированном коде языка C. Векторный код имеет множество преимуществ, в числе которых: – код более лаконичен и легче читается; – использование меньшего количества строк кода, что обеспечивает минимальный процент ошибок; – код становится похож на типичную математическую нотацию, которая, как правило, облегчает корректное программирование математических базисов; векторизация приводит к более удобному и простому коду. Без векторизации код был бы переполнен неэффективными и трудночитаемыми циклами.

Широковещательная передача — термин, используемый для описания неявного поэлементного поведения операций реализованного в NumPy. Операции в данной библиотеке бывают: арифметические, логические, побитовые, функциональные и так далее, ведут они себя неявно в терминах элементов.