УДК 004

Фартушнов Н.С.
студент магистратуры 2 курса
Поволжский Государственный Университет Телекоммуникаций и Информатики
научный руководитель: Бахарева Н.Ф., д.т.н., профессор
Россия, г. Самара

БИБЛИОТЕКИ ЯЗЫКА PYTHON ДЛЯ МАШИННОГО ОБУЧЕНИЯ, ИХ ВОЗМОЖНОСТИ И ПРЕИМУЩЕСТВА

Аннотация: Данная работа может помочь понять, чем хорош язык python в машинном обучении. Синтаксис Python проще и выше уровнем по сравнению с Java, C и C++. У него живое сообщество, культура с открытым исходным кодом, сотни высококачественных библиотек, ориентированных на машинное обучение, и огромная база поддержки от гигантов в индустрии (например, Google, Dropbox, Airbnb и др.). Эта статья будет посвящена некоторым библиотекам и возможностям Python, ориентированным на машинное обучение.

Ключевые слова: машинное обучение, python, numpy, pandas, matplotlib, seaborn, scickit-learn.

Fartushnov N.S.
magistracy student
2nd year, Povolzhskiy State University of Telecommunications and Informatics
Scientific supervisor: Bahareva N.F., Dr. of Engineering, professor
Russia, Samara

PYTHON LANGUAGE LIBRARIES FOR MACHINE LEARNING, THEIR OPPORTUNITIES AND ADVANTAGES

Annotation: This work may help to understand what python is good for in machine learning. Python syntax is simpler and higher than Java, C, and C ++. It has a vibrant community, an open-source culture, hundreds of high-quality machine learning-oriented libraries, and a huge support base from industry giants (like Google, Dropbox, Airbnb, etc.). This article will focus on some Python machine-learning libraries and features.

Keywords: machine learning, python, numpy, pandas, matplotlib, seaborn, scikit-learn.

Существует несколько базовых пакетов/библиотек Python, которые необходимо освоить для эффективного машинного обучения. Фундаментальные библиотеки, которые нужно знать и осваивать.

1. Numpy

NumPy является основным пакетом, необходимым для высокопроизводительных научных вычислений и анализа данных в языке Python. Это основа, на которой построены почти все инструменты более высокого уровня, такие как Pandas и sckit-learning. TensorFlow использует массивы NumPy как фундаментальный строительный блок, поверх которого они строили свои объекты Tensor и графический поток для задач глубокого обучения. Многие операции NumPy реализуются в C, что делает их сверхбыстрыми. Для информатики и современных задач машинного обучения это неоценимое преимущество. Возможности библиотеки представлен на рис. 1.

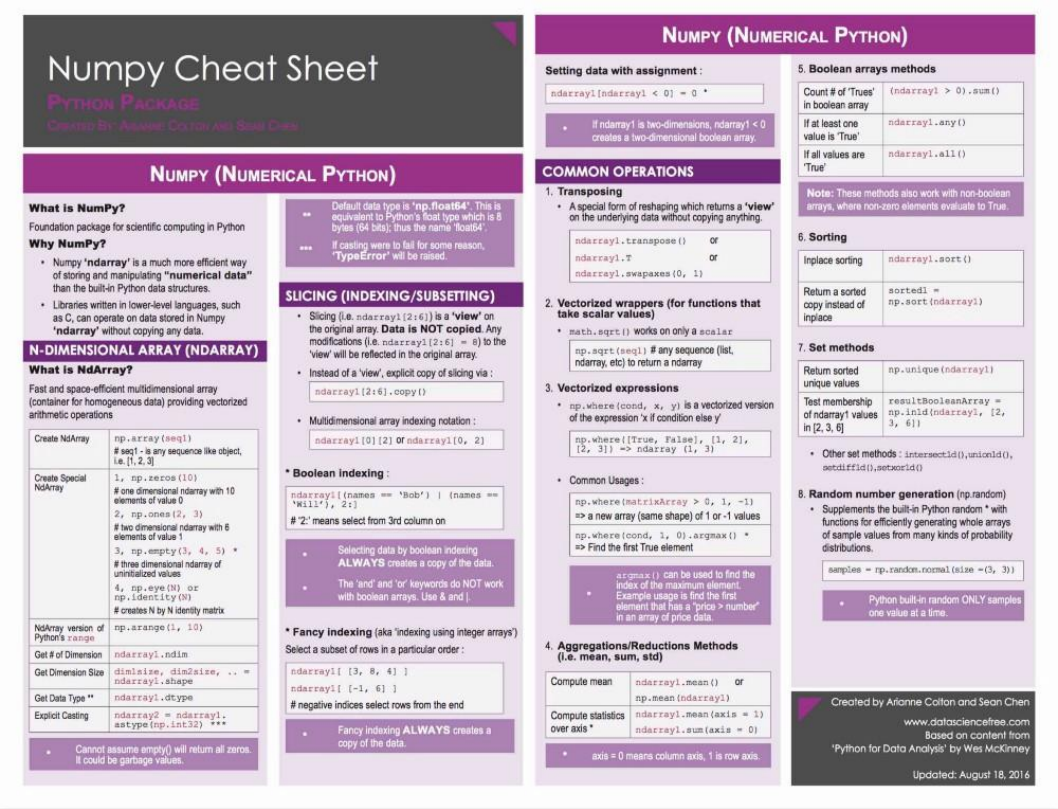

Рис. 1 – Возможности библиотеки NumPy

2. Pandas

Это самая популярная библиотека в машинном обучении Python для проведения анализа данных общего назначения. Pandas построен на массиве Numpy, тем самым сохраняя функцию быстрой скорости выполнения и предлагая множество функций проектирования данных, включая:

— Чтение/запись множества различных форматов данных

— Выбор подмножеств данных

— Расчет по строкам и столбцам вниз

— Поиск и заполнение отсутствующих данных

— Применение операций к независимым группам в данных

— Преобразование данных в различные формы

— Совмещение нескольких наборов данных

— Расширенные функциональные возможности временных рядов

— Визуализация через Matplotlib и Seaborn

На рис. 2 показаны возможности данной библиотеки.

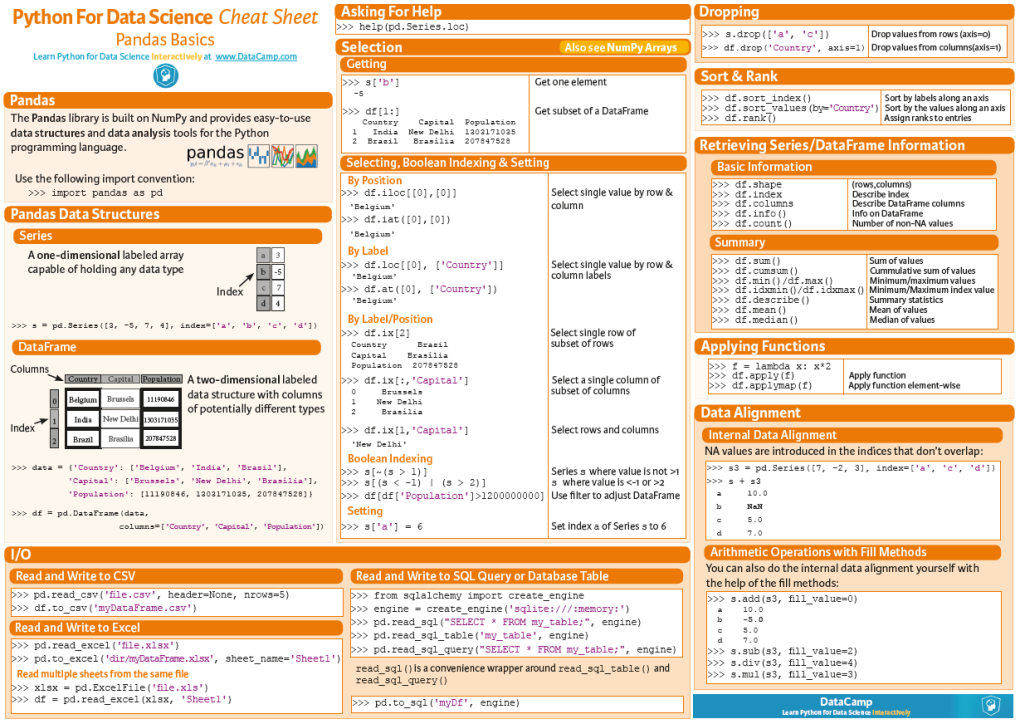

Рис. 2 – Возможности библиотеки Pandas

3. Matplotlib и Seaborn

Визуализация данных и описание с помощью данных - это важные навыки, которые необходимы каждому ученому в области обработки данных для эффективной передачи информации, полученной в результате анализа, любой аудитории. Это не менее важно в стремлении к мастерству в машинном обучении (ML). Очень часто в процессе ML, вы должны выполнить исследовательский анализ набора данных, прежде чем принять решение о применении конкретного ML алгоритма.

Matplotlib является наиболее широко используемой библиотекой визуализации 2-D Python, оснащенной богатым набором команд и интерфейсов для создания высококачественной графики из имеющихся данных. Ниже представлены графики построенные с помощью Matplotlib (рис. 3).

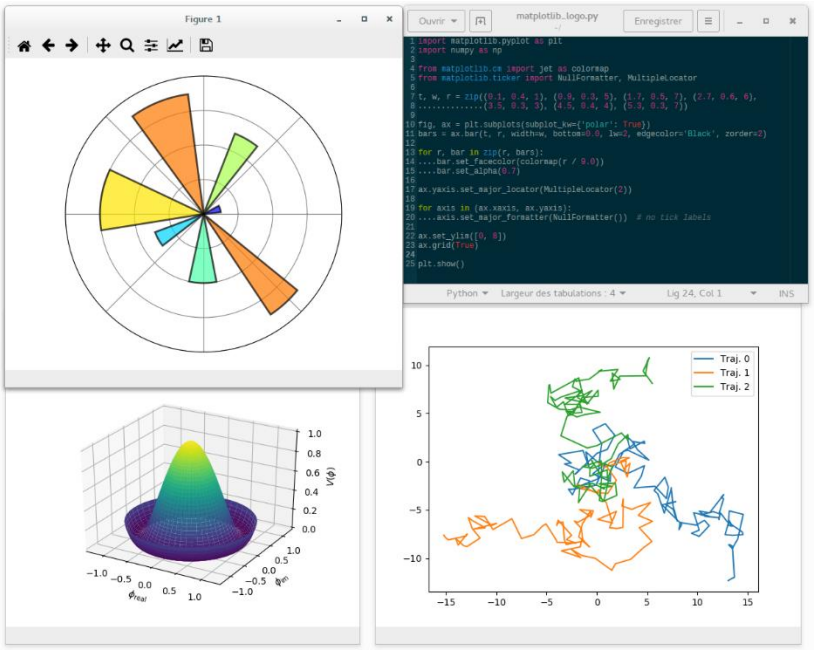

Рис. 3 - Графики построенные с помощью библиотеки Matplotlib

Seaborn - это еще одна отличная библиотека визуализации, ориентированная на статистическую печать. Seaborn предоставляет API (с гибким выбором стиля печати и цветов по умолчанию) поверх Matplotlib, определяет простые функции высокого уровня для общих статистических типов печати и интегрируется с функциональностью, обеспечиваемой Pandas (рис. 4).

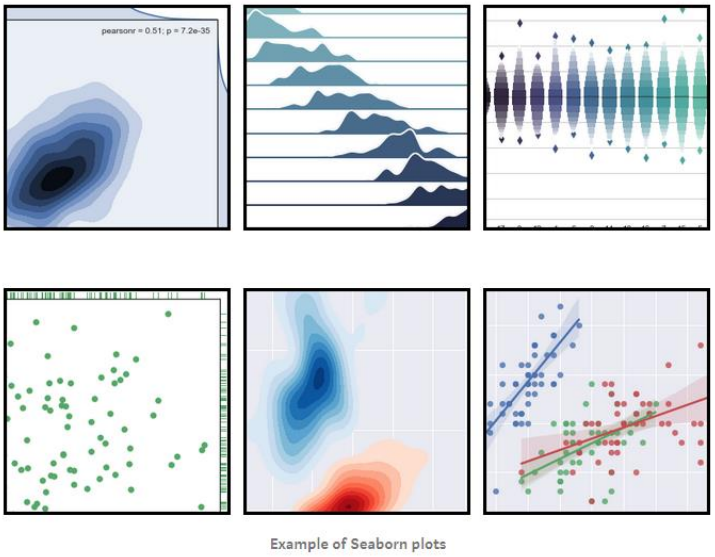

Рис. 4 – Примеры графиков Seaborn

4. Scickit-learn

Scickit-learn - самый важный общий пакет машинного обучения Python, который вы должны освоить. Он имеет различные алгоритмы классификации, регрессии и кластеризации, включая вспомогательные векторные машины, случайные леса, повышение градиента, k-средства и DBSCAN, и предназначен для взаимодействия с числовыми и научными библиотеками Python, NumPy и SciPy. Он предоставляет ряд контролируемых и без контроля алгоритмов обучения через последовательный интерфейс. Различные библиотеки имеют уровни надежности и поддержки, необходимые для использования в производственных системах. Это означает, что основное внимание уделяется таким проблемам, как простота использования, качество кода, совместная работа, документация и производительность.

Некоторые скрытые возможности Scickit-learn.

Scickit-learn - отличный пакет для обучения начинающим и опытным специалистам. Однако даже опытные специалисты по ML могут быть не осведомлены обо всех скрытых возможностях этого пакета, которые могут существенно помочь в выполнении их задачи. Я попытаюсь перечислить несколько из этих относительно менее известных методов/интерфейсов, доступных в Scickit-learn.

Pipeline: Может быть использован для преобразования цепочки нескольких оценок в одну. Это полезно, так как часто существует фиксированная последовательность шагов при обработке данных, например, выбор признаков, нормализация и классификация.

Grid-search: Hyper-parameters - это параметры, которые непосредственно не изучаются в оценщиках. В Scickit-learn они передаются в качестве аргументов конструктору классов оценщика. Можно и рекомендуется искать наилучший балл перекрестной проверки в

пространстве гиперпараметров. Любой параметр, предоставляемый при построении устройства оценки, может быть оптимизирован таким образом.

Кривые валидации: Каждый оценщик имеет свои преимущества и недостатки. Ошибка обобщения может быть разложена в терминах смещения, дисперсии и шума. Отклонением оценщика является его средняя ошибка для различных наборов обучения. Отклонение оценщика показывает, насколько он чувствителен к различным наборам учебных материалов. Шум является свойством данных. Очень полезно построить график влияния одного гиперпараметра на балл обучения и балл проверки, чтобы выяснить, является ли оценка избыточной или недостаточной для некоторых значений гиперпараметра. Scickit-learn имеет встроенный метод для этого (рис. 5).

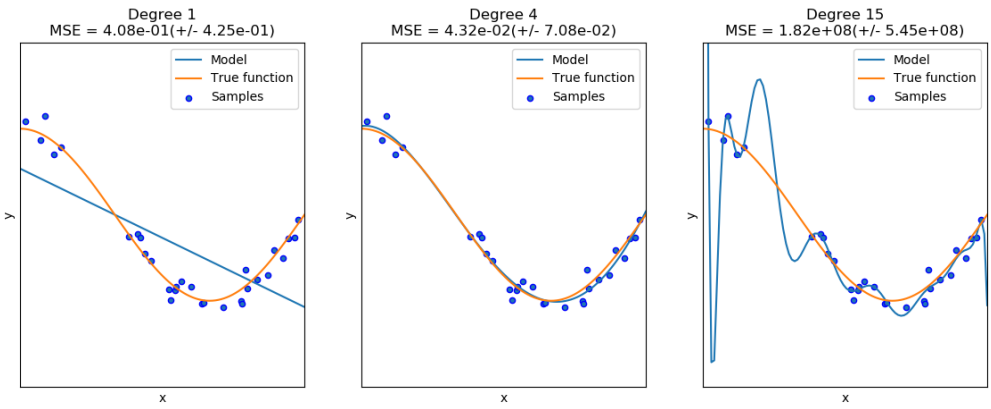

Рис. 5 – Построенные кривые валидации

Однокамерное кодирование категориальных данных: Это чрезвычайно распространенная задача предварительной обработки данных для преобразования входных категориальных признаков в двоичные кодировки "один в k" для использования в задачах классификации или прогнозирования (например, логистическая регрессия со смешанными числовыми и текстовыми признаками). Scickit-learn предлагает мощные, но простые методы для достижения этой цели. Они работают

непосредственно на массивах Pandas dataframe или Numpy, тем самым освобождая пользователя для записи любой специальной функции map/apply для этих преобразований.

Генерация полиномиальных элементов: для бесчисленных задач регрессионного моделирования. Часто полезно усложнять модель, рассматривая нелинейные элементы входных данных. Простой и распространенный метод - полиномиальные особенности, которые могут получить термины высокого порядка и взаимодействия элементов. Scickit-learn имеет готовую функцию для генерации таких перекрестных терминов более высокого порядка из заданного набора признаков и выбора пользователем высшей степени полинома.

Генераторы наборов данных: Scickit-learn включает в себя различные генераторы случайных выборок, которые могут быть использованы для построения искусственных наборов данных контролируемого размера и сложности. Он имеет функции для классификации, кластеризации, регрессии, декомпозиции матрицы и многокомпонентного тестирования (рис. 6).

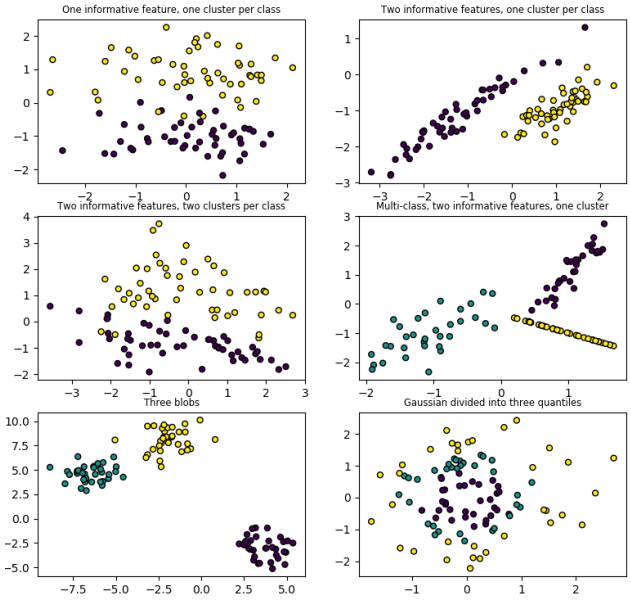

Рис. 6 – Графики кластеризации

5. Практика интерактивного машинного обучения

Проект Jupyter родился из проекта IPython в 2014 году и быстро развивался для поддержки интерактивной науки данных и научных вычислений на всех основных языках программирования. Нет сомнений, что это сильно повлияло на то, как специалист по данным может быстро протестировать и прототипировать свою идею и продемонстрировать работу сверстникам и сообществу с открытым исходным кодом.

Однако обучение и эксперименты с данными дают большой эффект присутствия, когда пользователь может интерактивно контролировать параметры модели и видеть эффект (почти) в реальном времени. Большинство распространенных визуализаций в Jupiter являются статическими.

Но если вы хотите больше управления, вы хотите изменить переменные простым нажатием на кнопку мыши, тогда можно использовать виджет IPython.

Виджеты - это насыщенные событиями объекты питона, которые имеют представление в браузере, часто в виде элемента управления, такого как ползунок, текстовое поле и т.д., через внешний (HTML/JavaScript) канал визуализации.

Использованные источники:

1. Иванов, В.М. Интеллектуальные системы : учебное пособие / В.М. Иванов. — Екатеринбург : Изд-во Урал. ун-та, 2015. — 92 с.
2. Остроух, А.В. Интеллектуальные информационные системы и технологии: Монография / А.В. Остроух, Н.Е. Суркова. – Красноярск: Научно-инновационный центр, 2015. – 370 с.
3. Вьюгин, В.В. Математические основы машинного обучения и прогнозирования: учебное пособие / В.В. Вьюгин. — Москва: 2013. — 387 с.