web of semantic

Journal of Interdisciplinary Science

Research Article Peer-review, Open Access

Алгоритмы, Используемые В Машинном Обучении, И Их Библиотеки На Python

Гульнора Ёдгоровна Буронова [1]

Файзуллаев Азизбек Отабекович [2]

Аннотация

Машинное обучение (Machine Learning, ML) – это область искусственного интеллекта, которая фокусируется на создании систем, способных обучаться на основе данных, выявлять паттерны и делать прогнозы или принимать решения без явного программирования.

[1] *Бухарский государственный университет, п.ф.ф.д., доцент*

[2] *Бухарский государственный университет, группа 1-2КИДТ-22, студент 2 курс*

Существует несколько основных типов машинного обучения:

1. Обучение с учителем (Supervised Learning):

- Классификация: это задача, при которой модель пытается предсказать метку класса для новых данных на основе обучающего набора данных с известными метками классов. Примеры включают определение, является ли электронное письмо спамом или нет.
- Регрессия: в этом случае модель предсказывает непрерывные значения на основе обучающих данных. Например, предсказание цены дома на основе его характеристик.

2. Обучение без учителя (Unsupervised Learning):

- Кластеризация: задача заключается в группировке данных на основе их сходства без заранее известных меток классов. Это помогает выявить скрытые структуры в данных.
- Снижение размерности: в этой задаче целью является уменьшение размерности данных, сохраняя при этом как можно больше информации. Это может помочь улучшить производительность модели и упростить анализ данных.

3. Обучение с подкреплением (Reinforcement Learning):

- В данном случае агент обучается путем принятия последовательности действий в окружающей среде с целью максимизации награды. Примерами являются обучение игровым стратегиям или управление роботами.

Машинное обучение (ML) является одной из самых значимых и динамично развивающихся областей в сфере технологий, преобразующей множество отраслей. Эта технология позволяет компаниям и организациям извлекать ценную информацию из больших объемов данных, автоматизировать процессы и принимать более обоснованные решения. В данной статье мы рассмотрим, как машинное обучение применяется в различных отраслях, включая здравоохранение, финансы, транспорт и другие.

Основные алгоритмы машинного обучения:

1.PCA (Анализ главных компонентов)

Это математический алгоритм уменьшения размерности наборов данных для упрощения количества переменных при сохранении большей части информации. Этот компромисс между точностью и простотой широко используется для поиска закономерностей в больших наборах данных (Рис. 1).

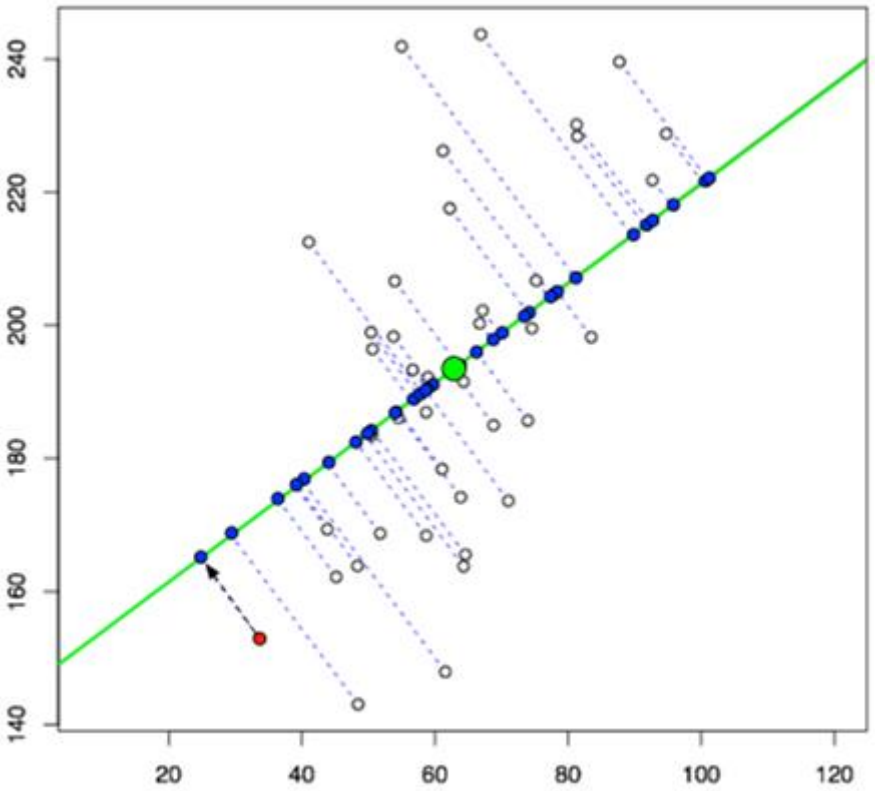

Рис. 1. PCA (Анализ главных компонентов)

2. SVD (разложение по сингулярным значениям)

В сфере машинного обучения SVD позволяет преобразовывать данные в пространство, в котором можно легко различать категории. Этот алгоритм разбивает матрицу на три разные матрицы. Например, при обработке изображений используется уменьшенное количество векторов для восстановления изображения, достаточно близкого к оригиналу (Рис. 2).

Рис. 2 Разложение по сингулярным значениям

3. LDA (линейный дискриминантный анализ)

Линейный дискриминантный анализ (LDA) — это метод классификации, при котором ранее были идентифицированы две или более группы, а свежие наблюдения классифицируются в одну из них на основе их особенностей.

Он отличается от PCA, поскольку LDA обнаруживает подпространство объектов, которое оптимизирует разделение групп, в то время как PCA игнорирует метку класса и фокусируется на выявлении направления наибольшего отклонения набора данных.

Этот алгоритм использует теорему Байеса, вероятностную теорему, используемую для определения вероятности события на основе его связи с другим событием.

Он часто используется в распознавании лиц, идентификации клиентов и в медицине для определения статуса заболевания пациента.

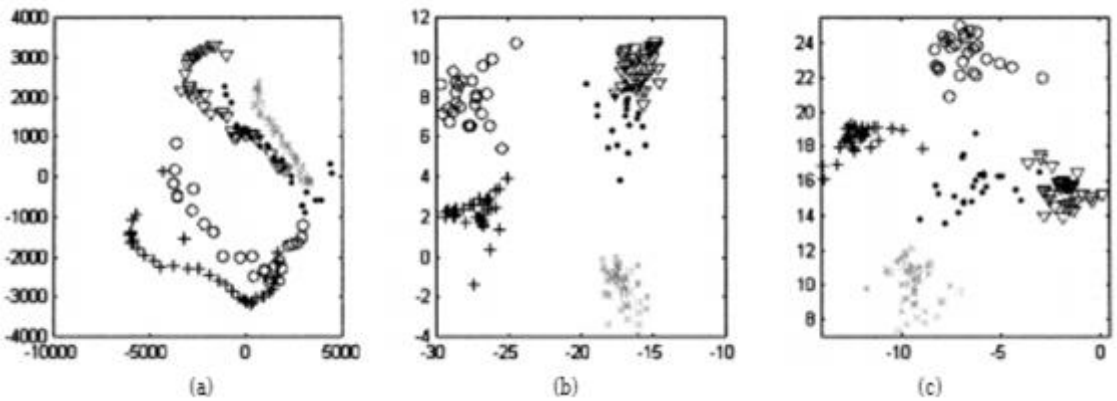

Рис. 3 Линейный дискриминантный анализ

4. Линейная регрессия

Линейная регрессия — один из наиболее известных и понятных алгоритмов в статистике и машинном обучении.

Прогностическое моделирование в первую очередь касается минимизации ошибки модели или, другими словами, как можно более точного прогнозирования. Мы будем заимствовать алгоритмы из разных областей, включая статистику, и использовать их в этих целях.

Линейную регрессию можно представить в виде уравнения, которое описывает прямую, наиболее точно показывающую взаимосвязь между входными переменными X и выходными переменными Y. Для составления этого уравнения нужно найти определённые коэффициенты B для входных переменных.

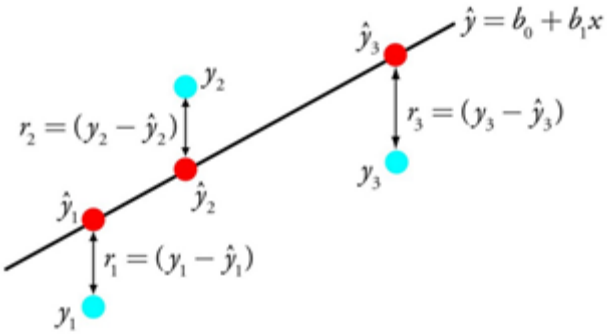

Рис. 4 Линейная регрессия

Например: Y = B0 + B1 X

Зная X, мы должны найти Y, и цель линейной регрессии заключается в поиске значений коэффициентов B0 и B1.

Для оценки регрессионной модели используются различные методы вроде линейной алгебры или метода наименьших квадратов.

Линейная регрессия существует уже более 200 лет, и за это время её успели тщательно изучить. Так что вот пара практических правил: уберите похожие (коррелирующие) переменные и избавьтесь от шума в данных, если это возможно. Линейная регрессия — быстрый и простой алгоритм, который хорошо подходит в качестве первого алгоритма для изучения.

5. *Логистическая регрессия*

Логистическая регрессия похожа на линейную тем, что в ней тоже требуется найти значения коэффициентов для входных переменных. Разница заключается в том, что выходное значение преобразуется с помощью нелинейной или логистической функции. Её хорошо использовать для задач бинарной классификации.

Логистическая функция выглядит как большая буква S и

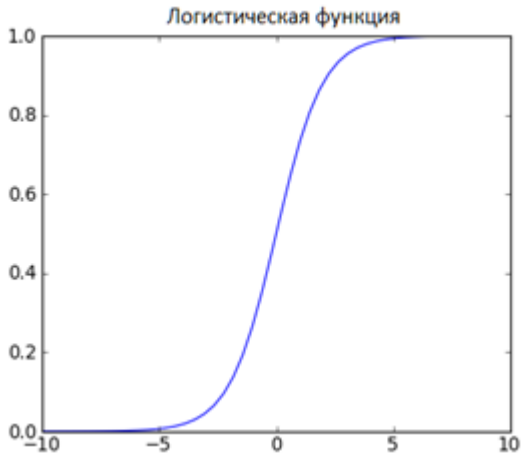

Рис. 5 Логическая регрессия

преобразовывает любое значение в число в пределах от 0 до 1. Это весьма полезно, так как мы можем применить правило к выходу логистической функции для привязки к 0 и 1 (например, если результат функции меньше 0.5, то на выходе получаем 1) и предсказания класса (Рис. 5).

6. Деревья принятия решений

Дерево решений можно представить в виде двоичного дерева, знакомого многим по алгоритмам и структурам данных. Каждый узел представляет собой входную переменную и точку разделения для этой переменной (при условии, что переменная — число).

Листовые узлы — это выходная переменная, которая используется для предсказания. Предсказания производятся путём прохода по дереву к листовому узлу и вывода значения класса на этом узле.

Деревья быстро обучаются и делают предсказания. Кроме того, они точны для широкого круга задач и не требуют особой подготовки данных.

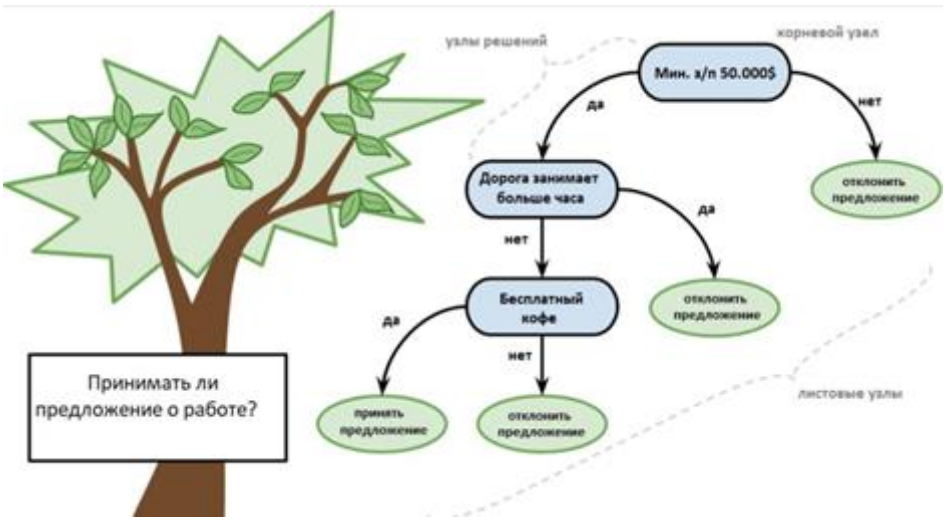

Рис. 6 Деревья принятия решений

Библиотеки и инструменты для машинного обучения на Python

Python предлагает множество полезных библиотек и инструментов для машинного обучения, вот некоторые из них:

- **NumPy**

✓ Функциональность: Библиотека NumPy предоставляет мощные функции для работы с многомерными массивами и матрицами, а также инструменты для выполнения различных математических операций. Ее функциональность включает в себя удобные методы для работы с массивами, линейной алгеброй, случайными числами и многое другое.

✓ Производительность: NumPy предоставляет эффективные реализации алгоритмов, написанных на языке программирования Python, благодаря использованию оптимизированных операций на C и Fortran. Это делает библиотеку NumPy одной из самых быстрых и эффективных для работы с массивами данных в Python.

Плюсы:

✓ Высокая производительность при работе с многомерными массивами данных.

✓ Обширный набор математических функций для работы с данными.

✓ Широко используется в научных вычислениях и машинном обучении.

Минусы:

✓ Не имеет возможностей для построения моделей машинного обучения напрямую.

✓ Иногда требует больше кода для выполнения определенных операций.

- **Pandas**

✓ Функциональность: Pandas — это библиотека для анализа данных, которая предоставляет удобные структуры данных и инструменты для работы с табличными данными. Она позволяет проводить предварительную обработку данных, выполнение операций над данными и подготовку данных для обучения моделей.

✓ Производительность: Pandas обладает хорошей производительностью и широкими возможностями для манипуляции данными. Она позволяет эффективно работать с большими объемами данных и выполнять сложные операции.

Плюсы:

✓ Удобная работа с табличными данными, включая чтение, запись и обработку данных.

✓ Поддержка множества операций для анализа и предобработки данных.

✓ Интеграция с другими библиотеками для машинного обучения.

Минусы:

✓ Может быть неэффективным при работе с очень большими объемами данных.

✓ Некоторые операции могут занимать больше времени из-за необходимости обработки данных в памяти.

- **Matplotlib**

✓ Функциональность: Библиотека Matplotlib в Python предоставляет широкие возможности для создания различных типов графиков и визуализаций, включая линейные графики, диаграммы, гистограммы и многое другое. Она позволяет настраивать графики до мельчайших деталей, что делает ее очень гибкой и мощной инструментом для визуализации данных.

✓ Производительность: Matplotlib может быть несколько медленной при построении сложных графиков с большим объемом данных. Однако, с правильной настройкой и оптимизацией кода, можно добиться приемлемой производительности даже при работе с большими объемами данных.

Плюсы:

✓ Мощные инструменты для визуализации данных, включая графики, диаграммы и диагностику моделей.

✓ Широкие возможности настройки внешнего вида графиков.

✓ Поддержка множества типов графиков.

Минусы:

✓ Некоторые пользователи могут считать его синтаксис сложным для начала работы.

✓ Не всегда удобно использовать для создания интерактивных графиков.

- **Scikit-learn**

✓ Функциональность: Scikit-learn — это широко используемая библиотека для машинного обучения, которая предоставляет инструменты для классификации, регрессии, кластеризации, предобработки данных и многих других задач.

✓ Производительность: Scikit-learn хорошо подходит для быстрого прототипирования моделей и обладает простым и понятным интерфейсом. Она также предлагает множество алгоритмов машинного обучения и инструменты для оценки моделей.

Плюсы:

- ✓ Простой и понятный интерфейс для построения моделей машинного обучения.
- ✓ Широкий выбор алгоритмов и инструментов для оценки моделей.
- ✓ Хорошо подходит для быстрого прототипирования моделей.

Минусы:

- ✓ Не всегда поддерживает самые новые и сложные алгоритмы машинного обучения.
- ✓ Может быть менее гибким для настройки некоторых аспектов моделей.

➢ **TensorFlow**

- ✓ Функциональность: TensorFlow — это библиотека с открытым исходным кодом, разработанная компанией Google, которая предоставляет инструменты для построения и обучения нейронных сетей и других моделей глубокого обучения.
- ✓ Производительность: TensorFlow обладает высокой производительностью и масштабируемостью, позволяя работать как на CPU, так и на GPU или TPU. Она широко используется в индустрии и исследованиях по глубокому обучению.

Плюсы:

- ✓ Высокая производительность при работе с нейронными сетями и глубоким обучением.
- ✓ Масштабируемость для работы на различных устройствах и платформах.
- ✓ Широкие возможности для исследований и разработки в области глубокого обучения.

Минусы:

- ✓ Может потребовать больше кода для построения простых моделей по сравнению с другими библиотеками.
- ✓ Некоторые пользователи могут считать его более сложным для начала работы.

➢ **Keras**

- ✓ Функциональность: Keras является высокоуровневым API для разработки нейронных сетей. Он предоставляет простой и интуитивно понятный интерфейс для создания моделей глубокого обучения. Keras можно использовать вместе с TensorFlow или Theano.
- ✓ Производительность: Keras обеспечивает хорошую производительность и может использовать GPU для ускорения обучения моделей.

Плюсы:

- ✓ Простой и интуитивно понятный интерфейс для построения нейронных сетей.
- ✓ Легко интегрируется с TensorFlow и другими библиотеками глубокого обучения.
- ✓ Подходит как для начинающих, так и для опытных специалистов.

Минусы:

- ✓ Может ограничивать гибкость при создании сложных моделей в сравнении с написанием кода на низком уровне.
- ✓ Некоторые продвинутые функциональности TensorFlow могут быть недоступны через Keras.

Заключение

В ходе выполнения самостоятельной работы по теме "Алгоритмы, используемые в машинном обучении, и их библиотеки на Python" были изучены основные принципы и методы машинного

обучения, а также рассмотрены ключевые библиотеки на языке программирования Python, предназначенные для работы с алгоритмами машинного обучения.

Анализ различных алгоритмов, таких как линейная регрессия, деревья принятия решений, метод опорных векторов (SVM), нейронные сети и другие, с использованием библиотек NumPy, Pandas, Scikit-learn, TensorFlow позволил глубже понять принципы их работы, преимущества и ограничения.

Изучение и практическое применение алгоритмов машинного обучения на Python позволило приобрести ценный опыт в области анализа данных, создания моделей прогнозирования и разработки интеллектуальных систем.

Дальнейшее изучение темы машинного обучения и его алгоритмов открывает широкие перспективы для применения в различных сферах, таких как финансы, медицина, технологии и другие области, где данные играют ключевую роль в принятии решений.

В заключении можно отметить, что освоение алгоритмов машинного обучения и использование соответствующих библиотек на Python является важным шагом для специалистов в области анализа данных и исследований, что позволяет создавать инновационные решения и улучшать процессы на основе данных.

Таким образом, выполнение данной самостоятельной работы расширило знания и навыки в области машинного обучения, подготовив к дальнейшему изучению и применению новейших методов и технологий в сфере искусственного интеллекта.

Список использованной литературы

1. Geeksforgeeks
2. Sky.pro
3. Habr.com
4. Hse.ru
5. Harvard.edu