



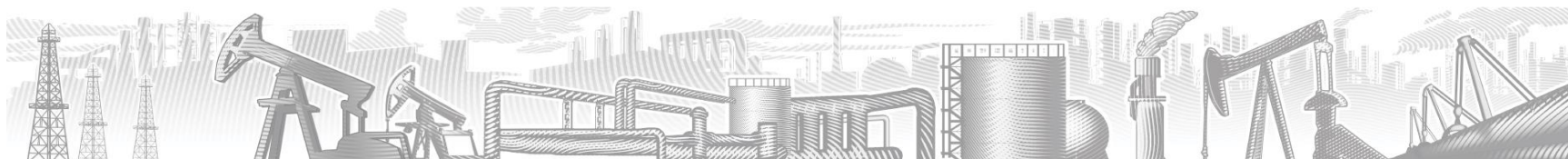
**УФИМСКИЙ ГОСУДАРСТВЕННЫЙ
НЕФТЯНОЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**
ИНСТИТУТ НЕФТИ И ГАЗА (филиал в г. Октябрьском)



Кафедра «Информационные технологии, математика и естественные науки»

Курс ДОП
ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И НЕЙРОННЫЕ СЕТИ

Разработал д.т.н. Тынчеров Камиль Талятович



Учебные вопросы лекции

1. Особенности применения Python для СИИ
2. Алгоритмы машинного обучения.



<https://cyberguru.tech/%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5/%D0%B8%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9-%D0%B8%D0%BD%D1%82%D0%B5%D0%BB%D0%B%D0%B5%D0%BA%D1%82-%D1%81-python>

Почему Python лучше для AI?

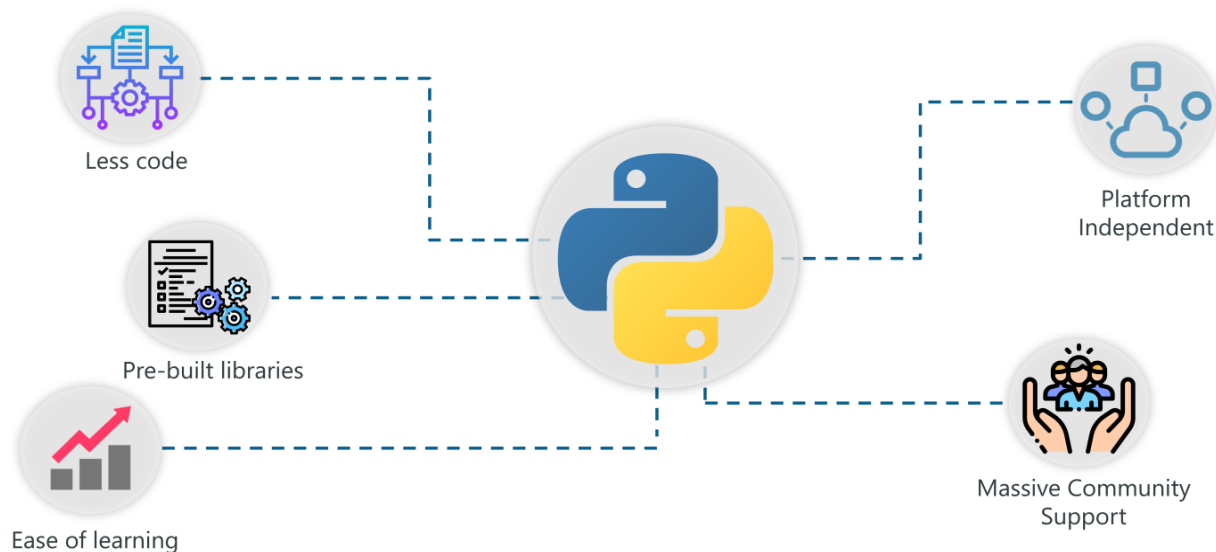
Искусственный интеллект существует уже более полувека, и его достижения растут в геометрической прогрессии. Спрос на ИИ находится на пике, и если вы хотите узнать об искусственном интеллекте, вы попали в нужное место.

Эта лекция по искусственному интеллекту с Python поможет вам понять все концепции искусственного интеллекта с практическими реализациями на Python.

Многие люди спрашивают: «Какой язык программирования лучше всего подходит для ИИ?» или «Почему Python для ИИ?» Несмотря на то, что Python является языком общего назначения, он нашел применение в самых сложных технологиях, таких как искусственный интеллект, машинное обучение, глубокое обучение и так далее. Почему Python приобрел такую большую популярность во всех этих областях?

Рассмотрим список причин, по которым Python является одним из лучших для разработчика, разработчика данных, инженера по машинному обучению и т. д.:

Почему Python лучше для AI?



- **Меньше кода:** реализация ИИ требует множества алгоритмов. Благодаря поддержке Python для предопределенных пакетов нам не нужно кодировать алгоритмы. И чтобы упростить задачу, Python предоставляет методологию «check as you code», которая снижает нагрузку на тестирование кода.
- **Готовые библиотеки:** в Python есть сотни готовых библиотек для реализации различных алгоритмов машинного обучения и глубокого обучения. Поэтому каждый раз, когда вы хотите запустить алгоритм для набора данных, все, что вам нужно сделать, это установить и загрузить необходимые пакеты с помощью одной команды. Примеры готовых библиотек – NumPy, Keras, Tensorflow, Pytorch и так далее.
- **Простота обучения:** Python использует очень простой синтаксис, который можно использовать для реализации простых вычислений, таких как добавление двух строк к сложным процессам, таким как построение модели машинного обучения.
- **Независимость от платформ:** Python может работать на нескольких платформах, включая Windows, MacOS, Linux, Unix и т. д. При переносе кода с одной платформы на другую вы можете использовать такие пакеты, как PyInstaller, которые позаботятся о любых проблемах с зависимостями.
- **Массовая поддержка сообщества:** у Python огромное сообщество пользователей, которое всегда помогает, когда мы сталкиваемся с ошибками кодирования. Помимо огромного количества поклонников, в Python есть несколько сообществ, групп и форумов, где программисты публикуют свои ошибки и помогают друг другу.

Tensorflow : эта библиотека, разработанная Google, широко используется при написании алгоритмов машинного обучения и выполнении сложных вычислений с использованием нейронных сетей.

Scikit-Learn: Scikit-learn – это библиотека Python, связанная с NumPy и SciPy. Он считается одной из лучших библиотек для работы со сложными данными.

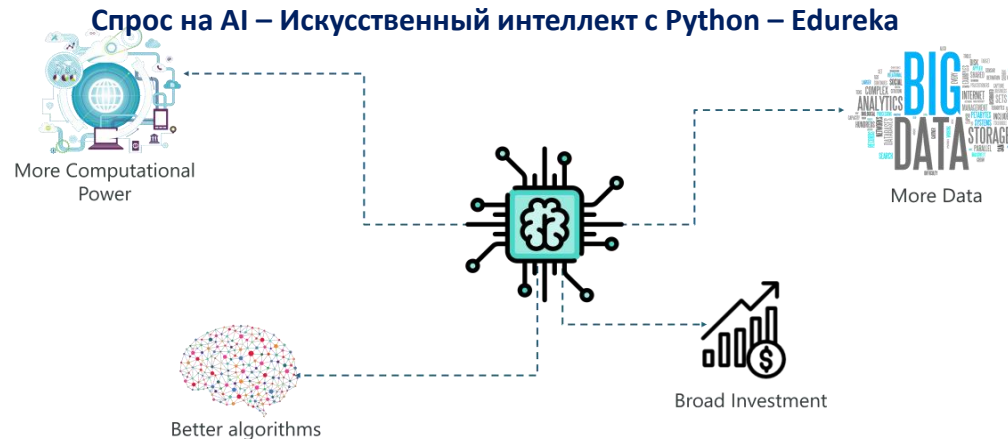
NumPy: Numpy – это библиотека python, специально используемая для вычисления научных / математических данных.

Theano: Theano – это функциональная библиотека, которая эффективно считает и вычисляет математические выражения с использованием многомерных массивов.

Keras: Эта библиотека упрощает реализацию нейронных сетей. Он также обладает лучшими функциональными возможностями для вычислительных моделей, оценки наборов данных, визуализации графиков и многого другого.

NLTK: NLTK или **Natural Language ToolKit** – это библиотека Python с открытым исходным кодом, специально созданная для обработки естественного языка, анализа текста и анализа текста.

Причины популярности Python в AI



- **Большая вычислительная мощность:** Реализация ИИ требует больших вычислительных мощностей, так как построение моделей ИИ требует сложных вычислений и использования сложных нейронных сетей. Изобретение графических процессоров сделало это возможным.. Наконец, мы можем выполнять высокоуровневые вычисления и реализовывать сложные алгоритмы.
- **Генерация данных:** за последние годы мы генерировали неизмеримое количество данных. Такие данные необходимо анализировать и обрабатывать с использованием алгоритмов машинного обучения и других методов искусственного интеллекта.
- **Эффективные алгоритмы.** За последнее десятилетие мы успешно разработали современные алгоритмы, которые включают в себя реализацию глубоких нейронных сетей.
- **Широкие инвестиции.** По мере того, как технологические гиганты, такие как Tesla, Netflix и Facebook, начали вкладывать средства в искусственный интеллект, они приобрели большую популярность, что привело к увеличению спроса на системы на основе искусственного интеллекта.
- **Рост Искусственного Интеллекта** является экспоненциальным, он также ускоряет рост экономики. Так что сейчас самое подходящее время для вас, чтобы попасть в область искусственного интеллекта.

1. Особенности применения Python для СИИ

Различные типы искусственного интеллекта

- **Узкий искусственный интеллект**. Обычно известный как узкий ИИ, искусственный узкий интеллект подразумевает применение ИИ только для выполнения определенных задач. Существующие системы на основе ИИ, которые утверждают, что используют «искусственный интеллект», фактически работают как слабый ИИ. Алекса является хорошим примером узкого интеллекта. Он работает в ограниченном предопределенном диапазоне функций. Алекса не имеет подлинного интеллекта или самосознания. Поисковая система Google Sophia, автомобили с автоматическим управлением и даже знаменитая AlphaGo подпадают под категорию слабого ИИ.

- **Общий искусственный интеллект**. Обычно известный как общий ИИ, искусственный общий интеллект включает в себя машины, которые обладают способностью выполнять любые интеллектуальные задачи, которые может выполнять человек. Видите ли, машины не обладают человеческими способностями, у них есть мощный процессор, который может выполнять высокоуровневые вычисления, но они еще не способны мыслить и рассуждать, как человек. Есть много экспертов, которые сомневаются в том, что AGI когда-либо будет возможен, и есть также много людей, которые задаются вопросом, нужно ли это. Стивен Хокинг, например, предупредил: «Общий ИИ взлетит сам по себе, и его дизайн будет постоянно расти. Люди, которые ограничены медленной биологической эволюцией, не могут конкурировать и будут вытеснены».

- **Супер искусственный интеллект**. Искусственный супер-интеллект – это термин, обозначающий время, когда возможности компьютеров превосходят людей. ASI в настоящее время рассматривается как гипотетическая ситуация, как это показано в фильмах и научно-фантастических книгах, где машины захватили весь мир. Тем не менее, вдохновители технологий, такие как Элон Маск, считают, что ASI захватит мир к 2040 году!

Прежде чем идти дальше, позвольте мне прояснить очень распространенное заблуждение. Задаёт этот вопрос каждый новичок: В чем разница между ИИ, машинным обучением и глубоким обучением?

1. Особенности применения Python для СИИ

Технологии AI, ML и DL

Люди склонны считать, что искусственный интеллект, машинное обучение и глубокое обучение – это одно и то же, поскольку они имеют общие приложения. Например, Siri – это приложение ИИ, машинного обучения и глубокого обучения.

Так, как эти технологии связаны?

- Искусственный интеллект – это наука о том, как заставить машины имитировать поведение людей.
- Машинное обучение – это подмножество искусственного интеллекта (AI), которое фокусируется на том, чтобы заставить машины принимать решения, предоставляя им данные.
- Глубокое обучение – это подмножество машинного обучения, которое использует концепцию нейронных сетей для решения сложных задач.

Подводя итог, ИИ, машинное обучение и глубокое обучение являются взаимосвязанными областями. Машинное обучение и глубокое обучение помогают искусственному интеллекту, предоставляя набор алгоритмов и нейронных сетей для решения задач, управляемых данными.

Тем не менее, искусственный интеллект не ограничивается только машинным обучением и глубоким обучением. Он охватывает широкий спектр областей, включая обработку естественного языка (NLP), обнаружение объектов, компьютерное зрение, робототехнику, экспертные системы и так далее.

Теперь давайте начнём с машинного обучения.

Термин «машинное обучение» был впервые введен Артуром Самуэлем в 1959 году. Оглядываясь назад, этот год был, вероятно, самым значительным с точки зрения технического прогресса.

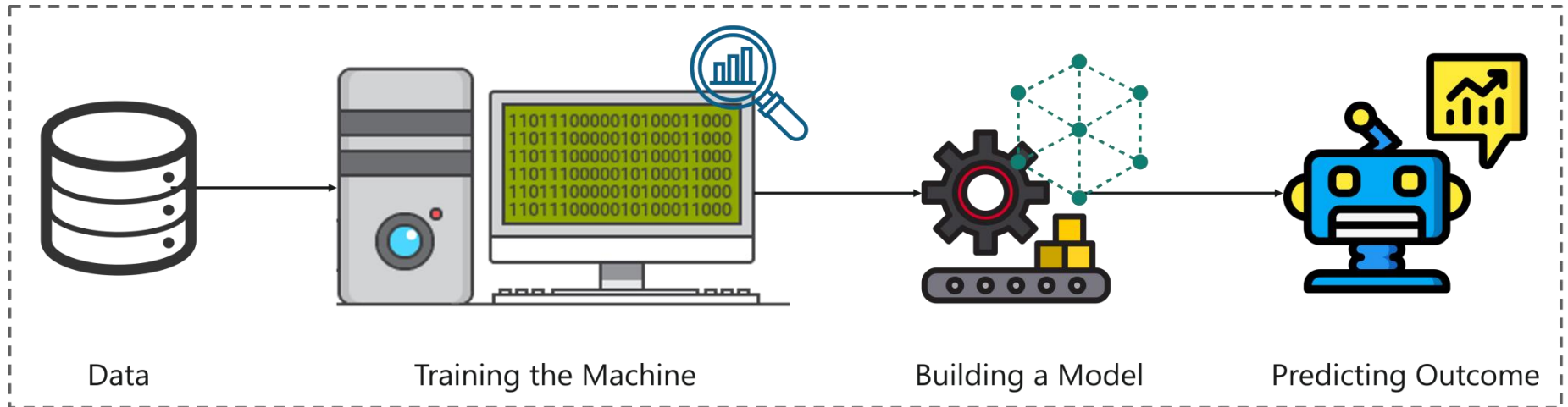
Проще говоря, **машинное обучение** – это подмножество искусственного интеллекта (ИИ), которое дает машинам возможность обучаться автоматически, снабжая его тоннами данных и позволяя улучшить его с помощью опыта. Таким образом, машинное обучение – это практика заставить машины решать проблемы, приобретая способность мыслить.

Но как машина может принимать решения?

Если вы передадите машине большое количество данных, она научится интерпретировать, обрабатывать и анализировать эти данные с помощью алгоритмов машинного обучения.

1. Особенности применения Python для СИИ

Что такое машинное обучение – Искусственный интеллект с Python



Процесс машинного обучения начинается с подачи на машину большого количества данных. Затем машина обучается на этих данных, чтобы обнаружить скрытые идеи и закономерности. Эти идеи используются для построения модели машинного обучения с использованием алгоритма для решения проблемы.

Теперь, когда мы знаем, что такое Машинное обучение, давайте рассмотрим различные способы, с помощью которых машины могут учиться.

1. Особенности применения Python для СИИ

Типы машинного обучения

Машина может научиться решать проблему, используя любой из следующих трех подходов:

- Контролируемое обучение
- Обучение без учителя
- Усиленное обучение

Контролируемое обучение – это техника, в которой мы обучаем или обучаем машину, используя данные, которые хорошо обозначены. Чтобы понять контролируемое обучение, давайте рассмотрим аналогию. В детстве мы все нуждались в руководстве для решения математических задач. Наши учителя помогли нам понять, что такое сложение и как оно делается. Точно так же вы можете думать о контролируемом обучении как о типе машинного обучения, которое включает в себя руководство. Помеченный набор данных – это учитель, который научит вас понимать закономерности в данных. Помеченный набор данных – это не что иное, как набор обучающих данных.

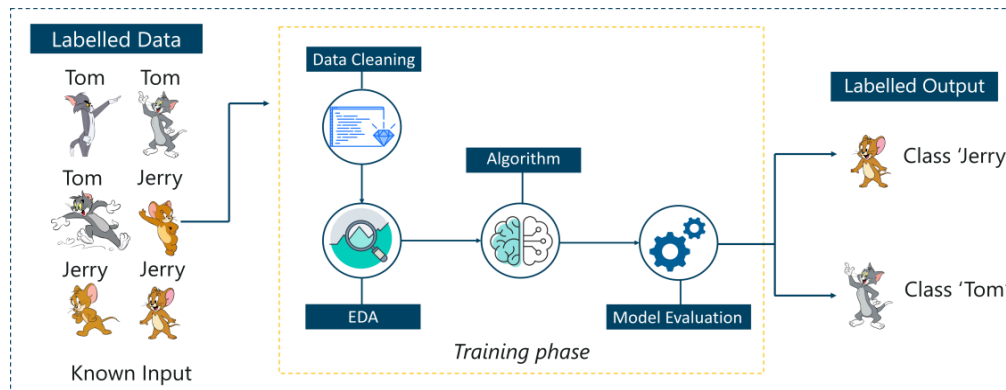


Рисунок. Обучение под наблюдением – Искусственный интеллект с Python – Edureka.

Здесь мы подаем компьютерные изображения Тома и Джерри, и цель состоит в том, чтобы машина идентифицировала и классифицировала изображения на две группы (изображения Тома и изображения Джерри). Набор обучающих данных, который подается на модель, помечен, как, например, мы говорим машине: «Так выглядит Том, а это Джерри». Тем самым вы тренируете машину, используя помеченные данные. В контролируемом обучении четко определенная фаза обучения выполняется с помощью помеченных данных.

1. Особенности применения Python для СИИ

Типы машинного обучения

Обучение без учителя

Обучение без учителя включает в себя обучение с использованием немаркированных данных и позволяет модели воздействовать на эту информацию без руководства.

Думайте о неконтролируемом обучении как о умном ребенке, который учится без какого-либо руководства. В этом типе машинного обучения модель не снабжается помеченными данными, поскольку в модели нет подсказки, что «это изображение – это Том, а это – Джерри», она самостоятельно выясняет закономерности и различия между Томом и Джерри, принимая тонны данных.

Например, он определяет характерные черты Тома, такие как заостренные уши, больший размер и т. Д., Чтобы понять, что это изображение относится к типу 1. Аналогично, он находит такие признаки в Джерри и знает, что это изображение относится к типу 2.

Поэтому он классифицирует изображения на два разных класса, не зная, кто такой Том или Джерри.

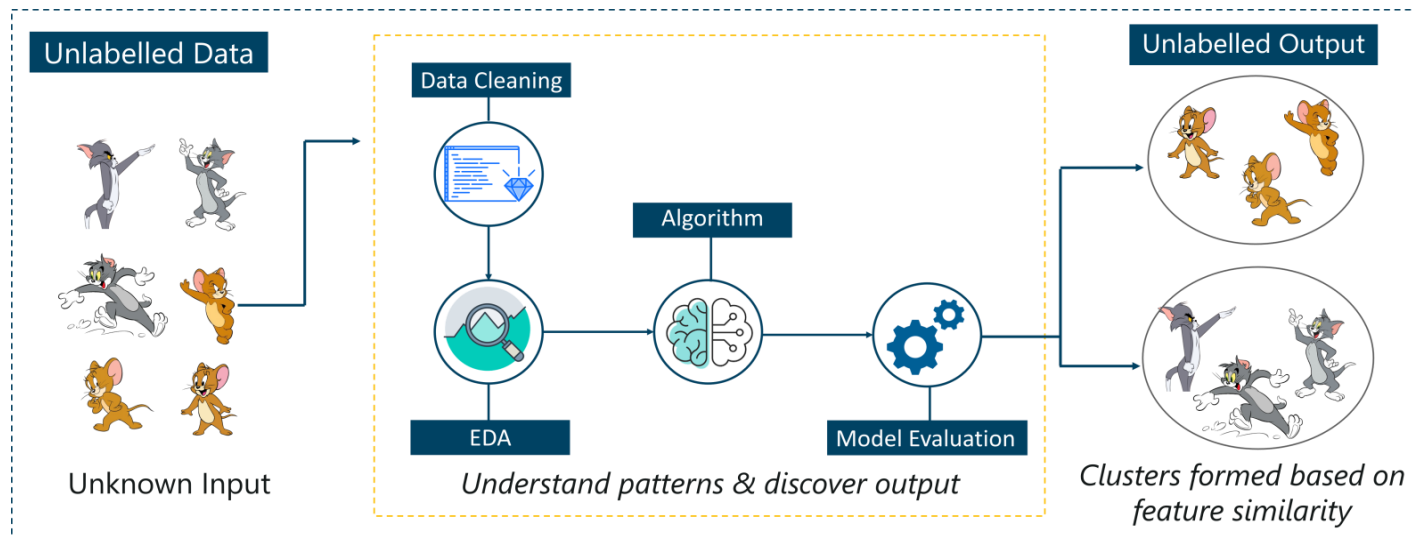


Рисунок. Обучение без учителя – Искусственный интеллект с Python – Edureka.

1. Особенности применения Python для СИИ

Типы машинного обучения

Усиленное обучения

Усиленное обучение является частью машинного обучения, когда объект помещается в среду, и он учится вести себя в этой среде, выполняя определенные действия и наблюдая за вознаграждениями, которые он получает от этих действий.

Представьте, что вас высадили на изолированном острове!

Что бы вы сделали?

Паника? Да, конечно. Но со временем вы научитесь жить на острове. Вы изучите окружающую среду, поймете климатические условия, тип пищи, которая там растет, опасности острова и т. Д.

Именно так работает Усиленное обучение, он включает в себя объекта (вы застряли на острове), который помещается в неизвестную среду (остров), где он должен учиться, наблюдая и выполняя действия, которые приводят к вознаграждениям.

Усиленное обучение в основном используется в таких областях, как машинное обучение, автомобили с самостоятельным вождением, AlphaGo и т. Д. Таким образом, подводятся итоги типов машинного обучения, посмотрим на тип проблем, которые решаются с помощью машинного обучения.

1. Особенности применения Python для СИИ

Какие проблемы может решить машинное обучение?

Существует три основных категории проблем, которые можно решить с помощью машинного обучения:

Регрессия

В этом типе проблемы результат является непрерывной величиной. Например, если вы хотите предсказать скорость автомобиля с учетом расстояния, это проблема регрессии. Проблемы регрессии могут быть решены с помощью алгоритмов контролируемого обучения, таких как линейная регрессия.

Классификация

В этом типе вывод является категориальным значением. Классификация электронных писем по двум классам: спам и не-спам – это проблема классификации, которую можно решить с помощью алгоритмов классификации Контролируемого обучения, таких как машины опорных векторов, наивный байесовский алгоритм, логистическая регрессия и т. Д.

Кластеризация

Этот тип проблемы включает в себя назначение входных данных в два или более кластеров на основе сходства функций. Например, кластеризация зрителей в схожие группы в зависимости от их интересов, возраста, географии и т. д. может быть выполнена с использованием алгоритмов неконтролируемого обучения, таких как K-Means Clustering.

1. Особенности применения Python для СИИ

7 этапов (шагов) процесса машинного обучения

Процесс машинного обучения включает в себя построение прогнозирующей модели, которую можно использовать для поиска решения для постановки задачи. Чтобы понять процесс машинного обучения, давайте предположим, что вы столкнулись с проблемой, которую необходимо решить с помощью машинного обучения. Проблема состоит в том, чтобы предсказать появление дождя в вашем регионе с помощью машинного обучения. Следующие шаги выполняются в процессе машинного обучения:

Шаг 1: Определите цель постановки задачи На этом этапе мы должны понимать, что именно нужно прогнозировать. В нашем случае цель состоит в том, чтобы предсказать вероятность дождя, изучая погодные условия. Также важно делать пометки о том, какие данные можно использовать для решения этой проблемы или какой подход вы должны использовать, чтобы найти решение.

Шаг 2: Сбор данных На этом этапе вы должны задавать такие вопросы, как, Какие данные необходимы для решения этой проблемы? Доступны ли данные? Как я могу получить данные? Как только вы узнаете, какие типы данных требуются, вы должны понять, как вы можете получить эти данные. Сбор данных может быть сделан вручную или с помощью веб-поиска.

Однако, если вы новичок и хотите изучать машинное обучение, вам не нужно беспокоиться о получении данных. В сети есть тысячи источников данных, вы можете просто загрузить набор данных и приступить к работе. Возвращаясь к рассматриваемой проблеме, данные, необходимые для прогнозирования погоды, включают такие показатели, как уровень влажности, температура, давление, местность, независимо от того, живете ли вы на холмистой станции и т. д. Такие данные должны быть собраны и сохранены для анализа.

Шаг 3: Подготовка данных. Собранные вами данные почти никогда не имеют правильного формата. Вы столкнетесь со множеством несоответствий в наборе данных, таких как пропущенные значения, избыточные переменные, повторяющиеся значения и т. д. Устранение таких несоответствий очень важно, потому что они могут привести к ошибочным вычислениям и прогнозам. Поэтому на этом этапе вы сканируете набор данных на наличие несоответствий и тут же исправляете их.

1. Особенности применения Python для СИИ

7 этапов (шагов) процесса машинного обучения

Шаг 4: Исследовательский анализ данных Хватайте свои детективные очки, потому что этот этап – это погружение в данные и поиск всех тайн скрытых данных. EDA или исследовательский анализ данных – это этап мозгового штурма машинного обучения. Исследование данных включает в себя понимание закономерностей и тенденций в данных. На этом этапе все полезные идеи прорисованы и корреляции между переменными понятны. Например, в случае прогнозирования осадков, мы знаем, что существует большая вероятность дождя, если температура упала низко. Такие корреляции должны быть поняты и отображены на этом этапе.

Шаг 5: Построение модели машинного обучения Все идеи и шаблоны, полученные в ходе исследования данных, используются для построения модели машинного обучения. Этот этап всегда начинается с разделения набора данных на две части: данные обучения и данные тестирования. Данные обучения будут использованы для построения и анализа модели. Логика модели основана на внедряемом алгоритме машинного обучения. В случае прогнозирования количества осадков, поскольку выходные данные будут в форме True (если завтра будет дождь) или False (завтра нет дождя), мы можем использовать алгоритм классификации, такой как логистическая регрессия или дерево решений. Выбор правильного алгоритма зависит от типа проблемы, которую вы пытаетесь решить, набора данных и уровня сложности проблемы.

Шаг 6: Оценка и оптимизация модели После построения модели с использованием набора обучающих данных, наконец, пришло время протестировать модель. Набор данных тестирования используется для проверки эффективности модели и того, насколько точно она может предсказать результат. Как только точность рассчитана, любые дальнейшие улучшения в модели могут быть реализованы на этом этапе. Такие методы, как настройка параметров и перекрестная проверка, могут быть использованы для повышения производительности модели.

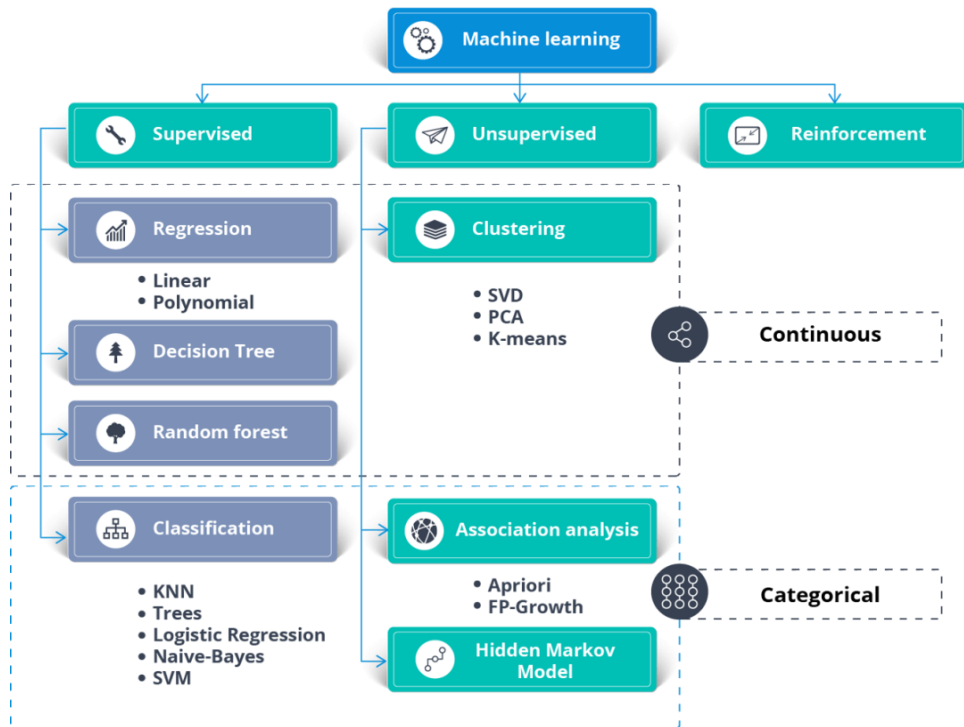
Шаг 7: Предсказания Как только модель оценена и улучшена, она, наконец, используется для прогнозирования. Конечным результатом может быть переменная категории (например, True или False) или непрерывное количество (например, прогнозируемая стоимость акции).

В нашем случае для прогнозирования выпадения осадков выходные данные будут категориальной переменной. Вот и весь процесс машинного обучения.

2. Алгоритмы машинного обучения.

7 этапов (шагов) процесса машинного обучения

Алгоритмы машинного обучения являются основной логикой каждой модели машинного обучения. Эти алгоритмы основаны на простых понятиях, таких как статистика и вероятность.



На рисунке показаны различные алгоритмы, используемые для решения проблемы с использованием машинного обучения.

Контролируемое обучение может использоваться для решения двух типов задач машинного обучения:

- регрессия
- классификация

Для решения задач регрессии вы можете использовать известный алгоритм линейной регрессии.

Проблемы классификации могут быть решены с использованием следующих алгоритмов классификации:

- Логистическая регрессия
- Древо решений
- Случайный Лес
- Наивный байесовский классификатор
- Метод опорных векторов
- k-ближайших соседей

2. Алгоритмы машинного обучения.

Машинное обучение с Python

Постановка задачи: Построить модель машинного обучения, которая будет предсказывать, будет ли завтра дождь, изучая прошлые данные.

Описание набора данных: Этот набор данных содержит около 145 тыс. наблюдений за суточными погодными условиями, наблюдаемыми на многочисленных австралийских метеостанциях. Набор данных имеет около 24 функций, и мы будем использовать 23 функции (переменные Predictor), чтобы предсказать целевую переменную, а именно, «Дождь завтра».

Эта целевая переменная (RainT Saturday) будет хранить два значения:

Да: означает, что завтра будет дождь

Нет: обозначает, что завтра не будет дождя

Следовательно, это явно проблема классификации. Модель машинного обучения классифицирует результаты на 2 класса: ДА или НЕТ.

Логика: Построить классификационные модели, чтобы предсказать, будет ли завтра дождь в зависимости от погодных условий.

Теперь, когда цель ясна, давайте поработаем над мозгом и начнем кодировать.

2. Алгоритмы машинного обучения.

Машинное обучение с Python

Шаг 1: Импортируйте необходимые библиотеки

```
# For linear algebra
import numpy as np
# For data processing
import pandas as pd
```

Шаг 2: Загрузите набор данных

```
#Load the data set
df = pd.read_csv('.. Desktop/weatherAUS.csv')
#Display the shape of the data set print('Size of weather data frame is :',df.shape)
#Display data
print(df[0:5])
```

Size of weather data frame is : (145460, 24)

	Date	Location	MinTemp ...	RainToday	RISK_MM	RainTomorrow
0	2008-12-01	Albury	13.4 ...		No	0.0
No 1	2008-12-02	Albury	7.4 ...		No	0.0
No 2	2008-12-03	Albury	12.9 ...		No	0.0
No 3	2008-12-04	Albury	9.2 ...		No	1.0
No 4	2008-12-05	Albury	17.5 ...		No	0.2
No						

Выводы.

1. Формализация *базы знаний* на основе *логической нейронной сети*, отображающей факты и правила вывода, позволяет производить формальные алгоритмические действия (удовлетворения целей) по получению новых фактов.
2. *База знаний* мыслящего существа должна наряду с фактами хранить хотя бы начальные абстрагированные правила вывода, обобщения, сообщенные в процессе обучения (от рождения – до смерти). Только факты и правила вывода лежат в основе дедуктивного *мышления*.
3. Следует отметить главенствующую роль перебора (поиска) в алгоритмах *логического вывода*. Перебор сопровождается анализом на противоречие промежуточных результатов фактам и правилам вывода, лежащим в основе *базы знаний*. (Здесь уже кроется возможность критического отношения к "современному" состоянию *БЗ*, противодействие сложившимся догмам!)
4. В согласии с главенствующим перебором при поиске решения или всего лишь гипотезы, реализуются "*backtracking*" другие многочисленные методы "ветвей и границ".
5. Понятно, что такого рода перебор, требующий высокого уровня ясности *мышления*, усидчивости и трудолюбия, не доступен широким массам трудящихся и нуждается в автоматизации.
6. Понятийная *база знаний*, представленная *логической нейронной сетью*, является важным формальным механизмом не только для получения новых фактов, но и для построения алгоритма новых обобщений на основе накопленного опыта и результатов вывода по текущему состоянию *БЗ*. Так могут формироваться гипотезы о новых правилах вывода. Это – путь к автоматизированной реализации индуктивного *мышления*, на основе теории познания, использующей понятия анализа и синтеза (знаний).

Выводы.

7. Проверка гипотез с помощью фактов, получаемых опытным путем, позволяет расширить понятийную нейронную сеть (*базу знаний*) новыми правилами вывода.
8. Целью построения формальных алгоритмов индуктивного *мышления* и, следовательно, его автоматизации является ускоренное получение новых обобщенных знаний – прогресс человечества.
9. ПРОЛОГ является гениальным языком, формализующим логический вывод. Формализация действий над ним с помощью механизмов логических *нейронных сетей* убеждает в продуктивности попыток автоматизации как дедуктивного, так и индуктивного *мышления*.
10. На уровне оперирования только лишь фактами ПРОЛОГ обладает большими возможностями целевого выбора информации, чем реляционные базы данных. Жаль, что этот язык *логического вывода*, возникший в период появления серьезных задач и продемонстрировавший свою эффективность, предан забвению, по-видимому, в связи с кризисным исчезновением таких задач.

1. Что собой представляет ПРОЛОГ-программа и как реализуется *backtracking* ?
2. Что представляет собой фактографическая *нейронная сеть*?
3. Что представляет собой понятийная *нейронная сеть*?
4. Достаточно ли в обыденной жизни пользоваться развитой фактографической *нейронной сетью*, осуществляя только дедуктивное мышление и пополняя ее новыми фактами?
5. Осуществимо ли индуктивное мышление на основе анализа и обобщения связей, сложившихся в фактографической *нейронной сети*?
6. Вопрос к размышлению: Достаточен ли аппарат логических *нейронных сетей* для анализа связей, сложившихся в фактографической *нейронной сети* – для их обобщения и формирования гипотез, или требуются алгоритмические механизмы *мышления* более высокого порядка?

1. Галушкин А.И **Нейронные сети: основы теории** М.: «Горячая линия – Телеком», 2010
2. Rosenblatt F. (Ф. Розенблат) **Perceptual generalization over transformation groups (Обобщение восприятий по группам преобразований)** Self-Organizing Systems, Pros. Interdisciplinary Conf., 5 – 6 May, 1959, 63 – 100 (Кибернетический сборник, 1962)
3. Поспелов Д.А **Моделирование рассуждений. Опыт анализа мыслительных актов.** М.; Радио и связь. 1989
4. Поспелов Д.А **Ситуационное управление. Теория и практика** М.: Наука, 1986
5. Барский А.Б. **Математическая логика событий и логические нейронные сети.** Приложение к журналу «Информационные технологии» №7, 2007
6. Барский А.Б. **Логические нейронные сети. Учебное пособие** М.: ИНТУИТ; БИНОМ. Лаборатория знаний. 2007
7. Барский А.Б. **Нейронные сети: распознавание, управление, принятие решений.** М.: Финансы и статистика, 2004, 2007
8. Барский А.Б. **Трёхмерная экранизация компьютерных объектов «живого» моделирования** Информационные технологии. 2010, №10
9. Барский А.Б. **Параллельные информационные технологии** М.: ИНТУИТ; БИНОМ. Лаборатория знаний. 2007
10. Ярушкина Н.Г. **Основы теории нечётких и гибридных систем.** М.: Финансы и статистика, 2004
11. Понс Ж, Форсайт Д. **Компьютерное зрение. Современный подход** М.: Издательский дом «Вильямс» 2004
12. Вахромцев Е.М. **Применение методов «живого» моделирования для производственного мониторинга** Информационные технологии. 2010
13. Ivan Bratko (И. Братко) **Programming for artificial intelligence (Программирование на языке Пролог для искусственного интеллекта)**
14. E. Kardelj – J. Stefan Institute. Yugoslavia. 1986 (М.: Мир, 1990)
15. 14. Поспелов Д.А **Десять «горячих точек» в исследованиях по искусственному интеллекту** Интеллектуальные системы (МГУ). Т. 1. Вып. 1...4. 1996
16. [НОУ ИНТУИТ | Лекция | Логическое программирование нейронной сети на базе языка ПРОЛОГ \(intuit.ru\)](http://intuit.ru)

Спасибо за внимание!