Введение в машинное обучение

Здравствуйте, уважаемые слушатели! Тема нашей лекции – Введение в машинное обучение (ML) .

План лекции:

1. Профессии, связанные с (ML)
2. Примеры применения ML в различных задачах
3. ПО для решения задач ML
4. Установка Python
5. Знакомство с Ipython
6. Синтаксис Python

1. Вступительное слово

В начале подготовки курса мы поставили цель заинтересовать обучающихся и показать, что машинное обучение (ML) не такая сложная наука. В данном курсе даются краткие теоретические и практические материалы об использовании отдельных алгоритмов машинного обучения. Так как курс является вводным, он рассчитан для обучающихся, не обладающим специальными знаниями в области ML. Для успешного прохождения курса понадобятся базовые знания языка Python, которые могут быть использованы для решения практических задач.

Впрочем, если обучающийся не имеет никакого навыка в программировании, но при этом интересуются анализом данных, он без всякого сомнения могут начать изучения этого курса.

2. Профессии, связанные с ML

На сегодняшний день компьютер является неотъемлемой частью нашей жизни. Сейчас сложно представить, как раньше люди обходились без него. Хотя в начале века он был роскошью для многих и применялся в основном для работы и вычислений.

Сегодняшним огромным возможностям вычислительных машин, мы обязаны не вычислительным машинам, а в основном, специалистам по ML. Другими словами, специалист по ML это программист, который с помощью данных и алгоритмов обучает вычислительную машину (искусственный интеллект) как правильно решить поставленную задачу.

Зачастую многие путают следующие профессии, считая, что все они специалисты по ML и решают одну и ту же задачу:

Data Analyst (аналитик данных);

Data Mining Specialist (специалист по интеллектуальной обработке данных);

Data Scientist (ученый по данным).

Стоит отметить, что не существует каких-либо четко разграниченных официальных определений каждой из этих профессий. Но для ясности, для дальнейшего понимания, мы предлагаем свою версию того, чем же эти профессии отличаются друг от друга.

Data Analyst (аналитик данных) — специалист, который помогает принимать правильные решения на основе анализа данных. В ходе работы он собирает данные, анализирует и выявляет закономерности. Результат своих исследований он предоставляет в виде графиков и диаграмм.

Data Mining Specialist (специалист по интеллектуальной обработке данных) — специалист, который не только анализирует данные, но еще создает прогнозируемые модели. В процессе обработки данных он использует машинное обучение (Machine Learning).

Data Scientist (ученый по данным) — специалист-ученый, который обладает знаниями и может делать работы, присущие вышеуказанным специалистам, а тому же он может получить новые знания на основе анализа данных.

3. Примеры применения ML в различных задачах

ML лежит в основе многих инновационных технологий искусственного интеллекта. Нужно выделить следующие сферы, где применение ML привели к технологическим прорывам:

Робототехника: беспилотные автомобили, роботы-пылесосы, трекеры сна, физической активности и здоровья и т.д.

Маркетинг: поисковые системы Google и Яндекс, социальные сети FaceBook, ВКонтакте, Instagram и т.д.

Безопасность: системы распознавания лиц, отпечатков пальцев, номера машин и т.д.

Финансовый сектор и страхование: Более точные биржевые прогнозы и оценка капитализации брендов, решения о выдаче кредитных продуктов частным лицам и предприятиям, определение стоимости и целесообразности страховки и даже снижение очередей в офисах при параллельном сокращении издержек на персонал.

Общественное питание: предложения для гостей с учетом загрузки посадочных мест в ресторанах и кафе, функционируют сервисы по планированию закупок для поваров и т.д.

Медицина: автоматизированные системы диагностики в медицине – путь увеличения широты и глубины охвата симптомов, оперативности, достоверности

Добыча полезных ископаемых: анализ почвы доказывает или опровергает наличие полезных ископаемых, помогает очертить площадь будущей разработки и т.д.

Сельское хозяйство: системы классификации и распознавания применяются для распознавания и прогнозирования размеров урожая по данным космических наблюдений, а также для уменьшения ручного труда при сортировке плодов по форме, цвету и размерам и т.п.

4. ПО для решения задач ML

В данном курсе язык Python был выбран в качестве основного языка программирования для освоения и реализации практического материала, так как имеет богатый набор стандартной библиотеки. В изучении машинного обучения в рамках данного курса будут применяться библиотеки, реализующие множество классических алгоритмов машинного обучения, импорт и экспорт данных и их визуализацию. На слайде перечислены библиотеки, которые будут использоваться при выполнении задач данного курса:

Обработка массивов и матриц
- **numpy**

Обработка данных, включая импорт и экспорт данных
- **pandas**

Анализ данных
- **scipy**
- **scikit-learn**

Визуализация данных
- **matplotlib**
- **seaborn**

и т.д.

5. Установка Python

Существует множество вариантов установки Python. Одна из них установка Python в составе дистрибутива Anaconda. Дистрибутив можно скачать с помощью данной ссылки: https://www.anaconda.com/. После выбрать версию для используемой Вами операционной системы и следовать указаниям. Запустить интерактивную оболочку IPython можно с помощью команды ipython notebook. В результате исполнения этой команды IPython должен открыться в новой вкладке браузера.

6. Знакомство с IPython

После в браузере откроется новая вкладка. С помощью кнопки «new» можно создать новый файл типа «IPython notebook» с расширением .ipynb.

Стоит отметить, множество вариации экспорта текущей тетради: .py, HTML или PDF. Для этого нужно зайти в подменю «Download As» меню «File».

В начале изучения каждого языка программирования пишем код выводящий «Hello, world!». Для этого мы используем функцию print, с помощью данной функции можно вывести на экран переменные любого типа:

```
a = 'Hello world!'
print(a)
```

OUT: Hello, world!

Также, Ipython можно применить в качестве калькулятора. Например:

1/3
OUT: 0.3333333333333333

round (1/3,2)
OUT: 0.33

В примере используется функция round, для округления выражения задаваемой её вторым аргументов.

В тетради имеется несколько типов ячеек помимо с кодом на Python. Чтобы изменить тип ячейки необходимо выбрать нужный Вам в выпадающем списке в панели инструментов.

7. Синтаксис IPython

В машинном обучении немаловажную роль играет тип данных. Как мы оговаривались в начале в Python можно работать с переменными любого типа. Для начала нужно начать с числовых типов данных. Для определения типов данных используем функцию type(). Например:

type (8.)
OUT: float

Так как мы ввели число с плавающей точкой, тип у нее float. Помимо float (вещественные числа), в «Питоне» имеется и другие типы числовых данных:

int (целое число). Пример:
b = 8
type (b)
OUT: int

complex (комплексное число). Пример:
x = complex(1, 2)
print(x)
OUT: (1+2j)

type(x)
OUT: complex

В Python динамическая типизация. Другими словами переменные могут менять свой тип с помощью функции astype().

Также, в Python есть логический тип данных, bool. True это 1, a False это 0. Так как bool это логический тип данных, к нему можно применять логические операции «и», «или». Давайте применим эти знания на практике. Пример:

a = True
b = False
print (a + b, a + a, b + b)
OUT: 1 2 0

type (a)
OUT: bool

print (a and b)
OUT: False

```
print (a or b)
OUT: True
```

В Python есть тип none, который означает отсутствие значения. Многие часто путают, думая что он равен 0 или False.

```
z = None
print(z)
OUT: None

type(z)
OUT:NoneType
```

У данного типа есть свое название: NoneType. В отличие от других типов его не получится преобразовать к какому-то другому типу.

```
print(int(z))
OUT: int() argument must be a string, a bytes-like object or a number, not 'NoneType'
```

Тип str (строка). Пример:

```
a = 'python'
type(a)
OUT: str
```

Str также можно складывать. Это происходит следующим образом:

```
x = 'Machine'
y = 'learning'
z =x + ' ' + y
z
OUT: 'Machine learning'
```

Как вы заметили в Python можно выводить переменные без функции print(). Можно приводить к нижнему регистру или к верхнему регистру.

```
print(x.upper())
print(x.lower())
OUT:        MACHINE
            machine
```

Можно получить длину строки.

```
print(len(x))
OUT: 7
```

Можно получать различные элементы строки.

```
print(x[2])
print(x[3:7])
OUT:  c
      hine
```

Можно разбивает строку по какой-то другой строке.

```
splt = 'Machine learning'.split(' ')
print(splt)
OUT: ['Machine', 'learning']

type (splt)
```

OUT: list

List – один из типов массива. С помощью него можно хранить объекты разных типов. Но сначала нужно ввести понятие индексации, который работает также как в str.

```
splt[1]
OUT: 'learning'
```

Также, можем добавлять элемент массива с помощью метода append и удалять элементы.

```
splt.append('1')
splt
OUT: ['Machine', 'learning', '1']

del splt[-2]
splt
OUT: ['Machine', '1']
```

Tuple (кортеж) – еще один тип массива. В отличии от list его объекты не изменяются, а также использует обычные скобки.

```
t = ('Machine', 'learning', '1')
type (t)
OUT: tuple
```

Dict (словарь) - набор комбинаций ключа и значений.

```
d = {'Machine': 1, 'learning': 3, '1': 2}
type (d)
OUT: dict
```

На этом наш вводный урок заканчивается.