

테이터사이언스

〈물리학 및 캡스톤〉

물리학과 이시연

2015550024

목차

1. 주제 소개
2. 기업의 예시 -넷플릭스
3. 학습 내용
 1. 데이터 스크래핑
 2. 머신러닝
 3. Star cluster Simulation
4. 참고문헌 및 출처

데이터 사이언스

주제 소개

데이터를
수집, 분석, 활용하는
모든 기술의 집합



인과관계와 상관관계 사이의 긴장

010001000

**BIG
DATA**

101011010

≠



과거에도 그랬으니 미래에도 이럴 것이다

과거 설명과 미래 예측은 근본적으로 다르다

빅데이터로 인한 3대 변화

- 자료 규모

- 기존의 표집에서 관찰 불가능한 소수 하위집단 세분화 관찰 가능

- 자료 구성

- 엄밀한, 통제된 정확성을 요구, 충족 불가능

- 분석 준거

- 인과관계 → 상관관계

빅데이터 기업의 실제 이용 사례

넷플릭스를 중심으로

NETFLIX



어떤 영화를 좋아할까?



어떤 영화에 투자할까?

사람들은 어떤 영화를 좋아할까

별점을 넘어서 개인화 시스템

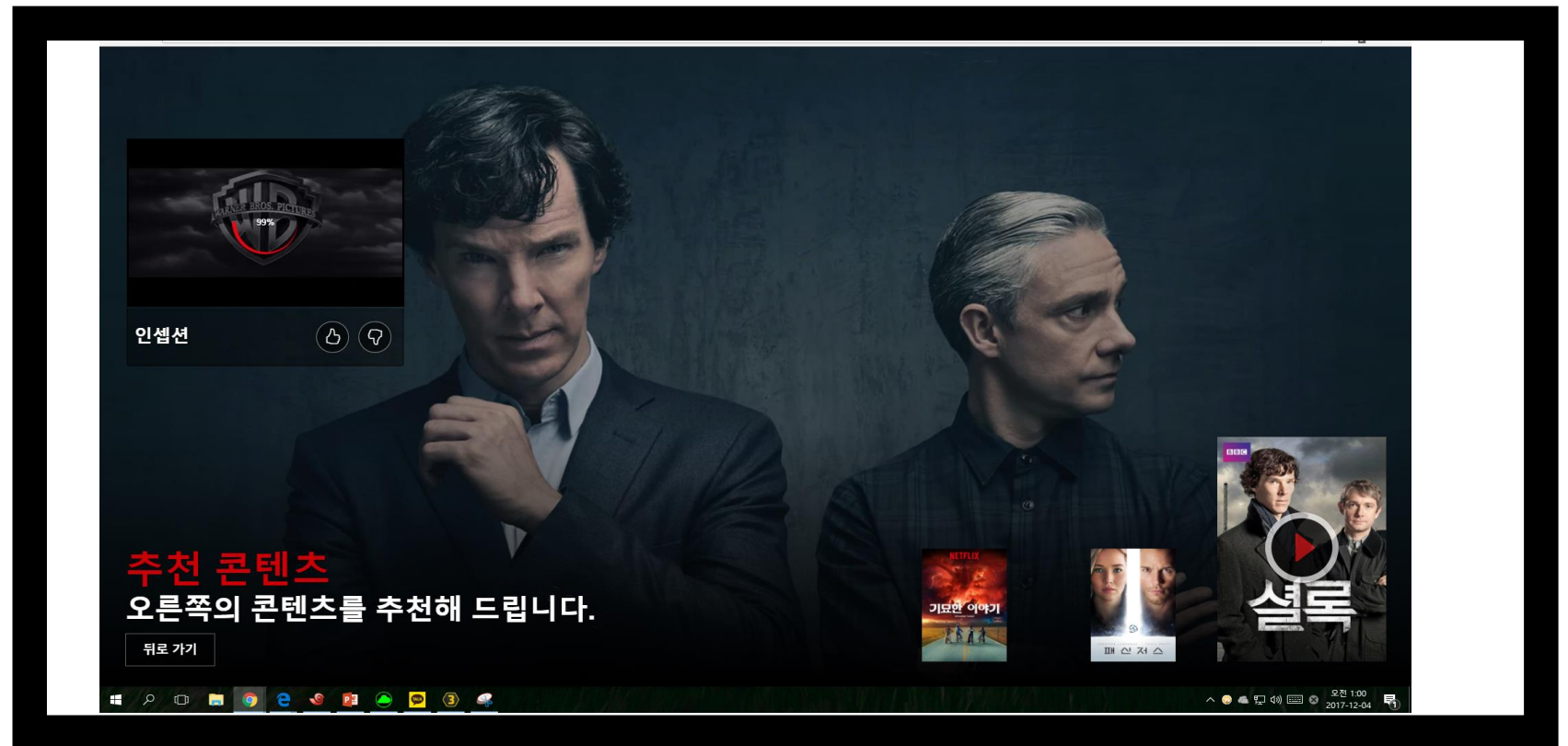
수많은 태그

A/B 테스트

영화 본 뒤 연이은 추천

실제 넷플릭스 접속 화면

- 사용자의 취향에 맞춰 접속시 뜨는 화면이 다르다.



넷플릭스의 정보 수집에 동의하십니까?

- ✓ 특정 영화 시청 중 일단정지, 되돌리기, 빨리가기 하는 지점
- ✓ 시청한 요일, 날짜, 시간
- ✓ 시청한 장소(주소)
- ✓ 시청에 사용한 기기
- ✓ 시청을 중단한 지점
- ✓ 시청 후 사용자가 준 별점
- ✓ 사용자의 영화 검색 내용
- ✓ 영화를 고르는 동안 하는 행위

협업 필터링

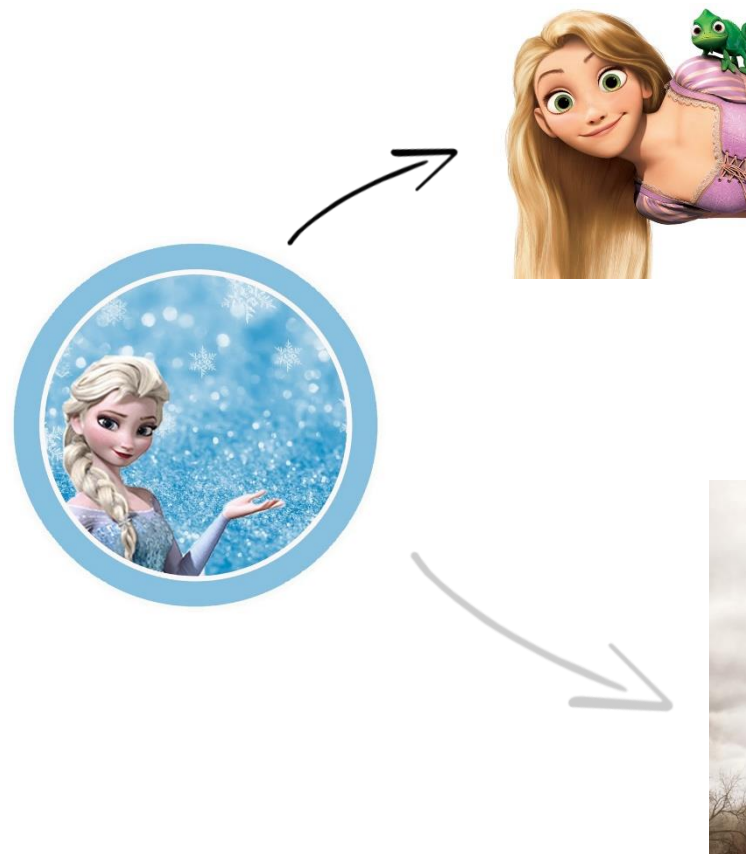
사용자 기반 알고리즘



시연 : 겨울왕국 5점 컨저링 2점 라퐁젤 4점
채은 : 겨울왕국 2점 컨저링 4점 라퐁젤 3점
지영 : 겨울왕국 4점 컨저링 1점 라퐁젤 ???



아이템 기반 알고리즘



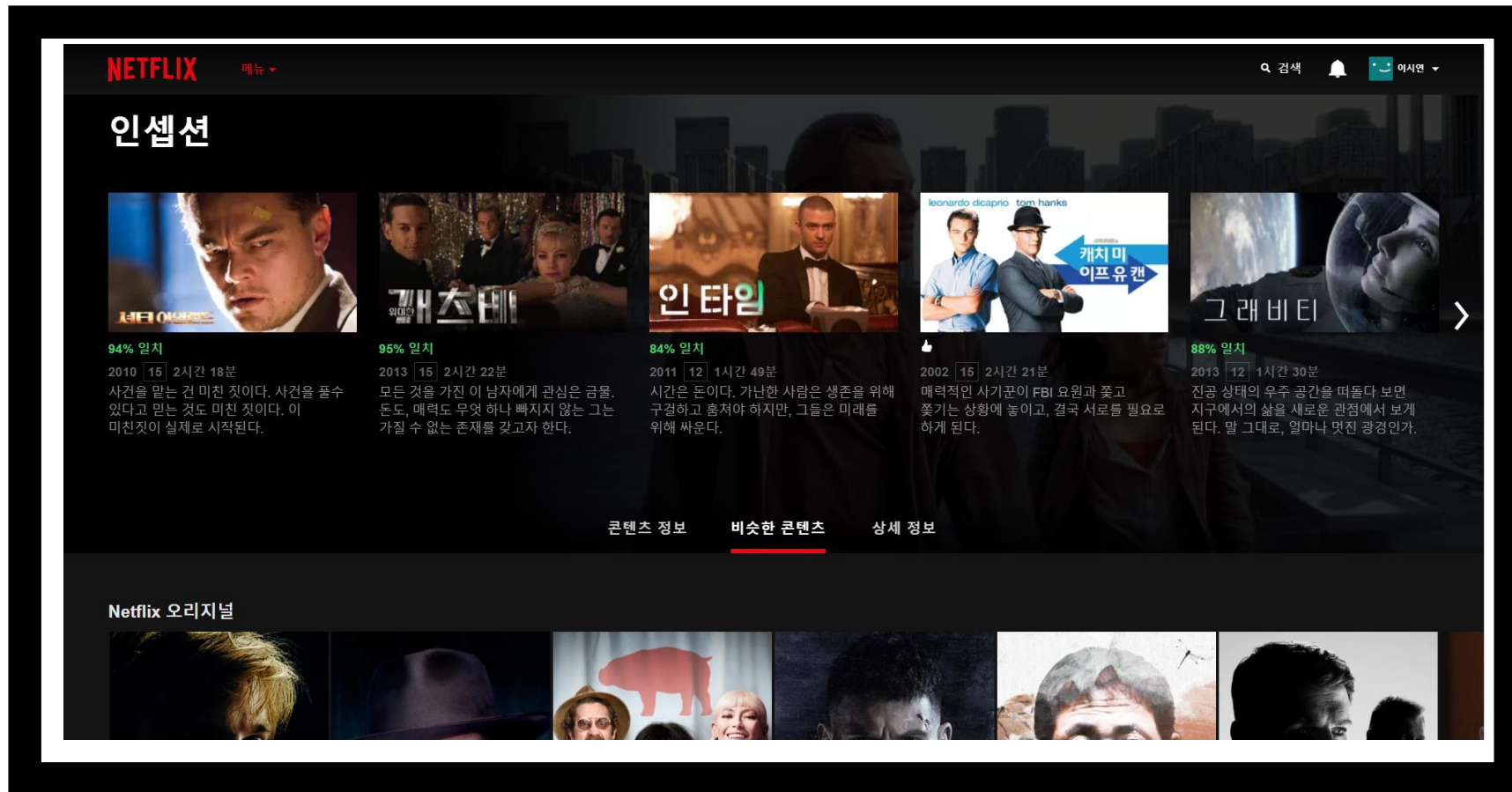
별집을 넘어서

- 개인화 시스템
- 보기패턴이 비슷한 사람들 묶기
 - 요일, 시간대, 장치, 위치
 - 페이스북 친구와 연동
- A/B 테스트 기법
- 엔딩 크레딧이 뜨면 추천 시스템 작동
- 다음 에피소드 자동 시작

I love 다큐
I love 외국영화



개인화를 통한 영화 추천



이시연(23) 대학생



나채은(23) 휴학생



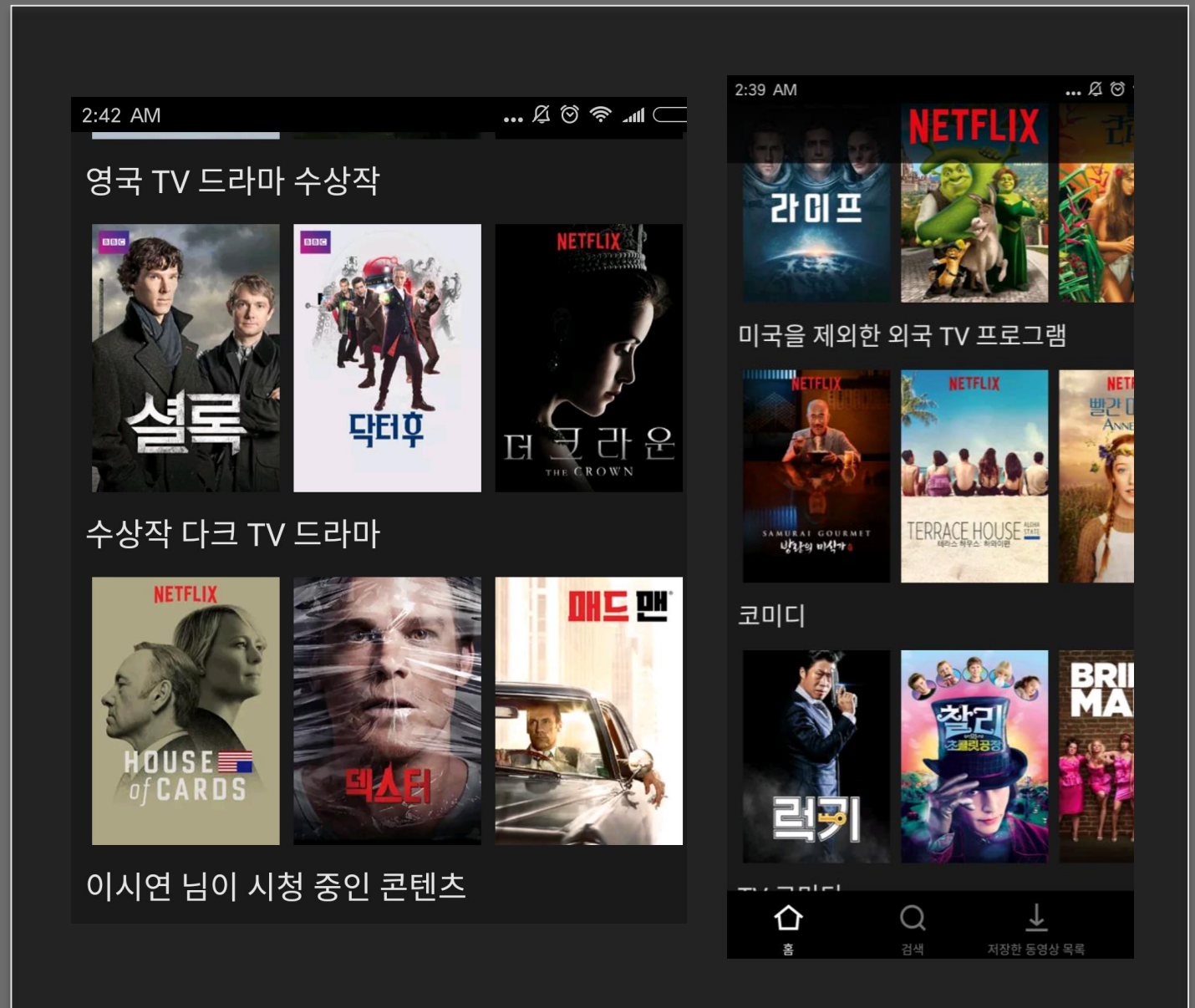
오빛나리(29) 직장인



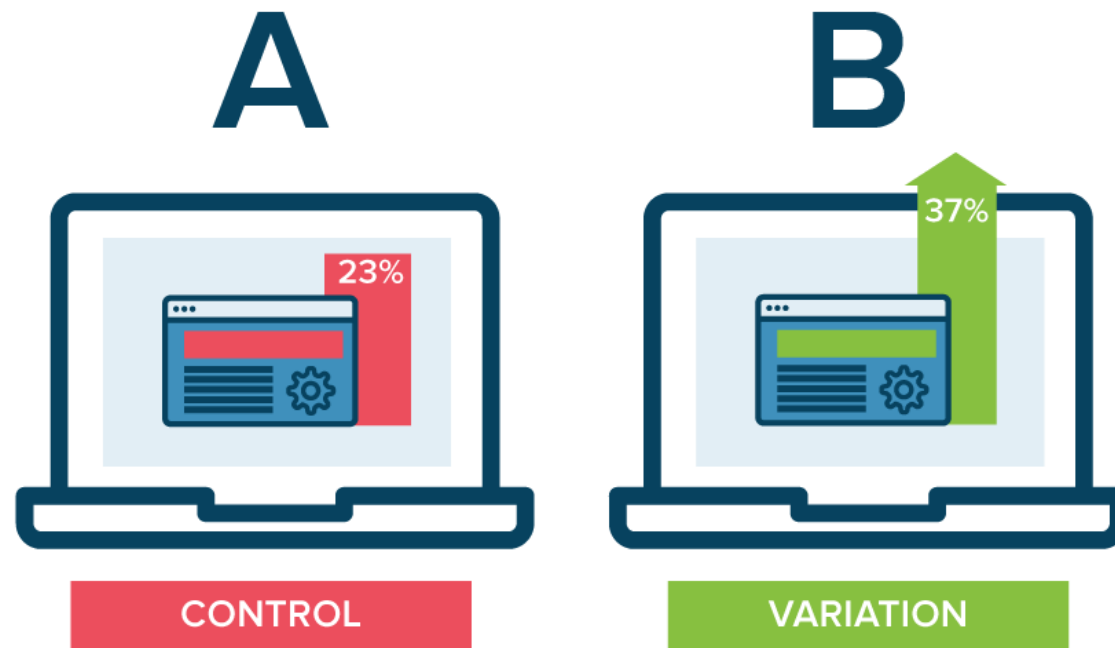
같은 영화라도 사용자의 취향에 따라 다른 포스터가 등장한다.

영화를 구분하는 76,897 방법

- 넷플릭스의 영화장르는 리버스엔지니어링 통해 밝혀진 것만 76897개.
- 단순한 분류 아닌 개인화 추천



A/B 테스트



인과관계 속 상관관계 제거

현황 및 관련 사업



2010년 인구 총조사



**인구성장률
노동인구
유소년 및 노령 인구 변화 예측
시군구 별 인구 이동변화 예측**



**국민기초생활보장
기초노령연금
영유아 복지
노인 일자리 등
사업별 수급자 및 예산 변화 전망**

해외 기업



RateMyDrive



고객 임신여부
- 프로모션

Ship Before They Buy



Amazon.com plans to ship you things before you even buy them. Using predictive analytics, the online retailer will guesstimate your next purchase.

국내 기업



영화 추천 스트리밍 서비스

Macro Trend와 Micro Trend를 조합한 남/녀 각 9개 Trend Code

| M | | | W | | |
|--|---|---|---|--|--|
| Rookie 새롭고 다양한 분야에 관심이 많은 프레스맨 | LOEL 패션 및 명품 브랜드에 관심이 많으며 외모가꾸길 즐기는 센스남 | Friend Daddy 자녀와의 친밀감을 중시하는 친구처럼 가까운 아빠 | it-Girl 최신 유행 및 트렌드를 선도하는 패셔니스타 | Prima Donna 문화와 여가를 즐기는 전문직 싱글 여성 | Trend Setter 럭셔리한 삶을 추구하는 세련된 감각의 여성 |
| Smart Saver 합리적 가격을 중요시하는 계획적인 소비를 즐기는 플랜맨 | Mr. Routine 하루하루 최선을 다하며 소소한 행복을 추구하는 가장 standard 한 직장인 | BOBOS 일과 여가를 즐기며, 독특한 소비감각을 지닌 여유로운 남성 | Rudy 자기계발에 적극적인 젊은 감각의 여성 | Alpha Mom 자녀교육에 매진하는 똑똑한 엄마 | Queen House 경제관이 뚜렷하여 가족을 적극 보살피는 내조의 여왕 |
| Bravo Life 사회적 기여에 관심이 많고, 젊고 댄디한 감성을 잃지 않은 남성 | Realist 건강과 웰빙, 일과 여가의 균형을 추구하는 이성적 남성 | Gray Gentleman 필수적 소비에 집중하며 삶의 질을 중시하는 시니어 남성 | 쥘마렐라 외모와 건강에 관심이 많은 사교적인 여성 | Grace Woman 레저와 여가를 즐기며, 기부활동에 적극적인 여성 | Silver Lady 건강을 유지하며 삶 자체를 즐기는 시니어 여성 |

국내 활용도는 아직 낮은 편

- 중견기업 빅데이터 도입률 9.6%
- 종업원 100명 이상 도입률 4.3%
- 데이터 절대적으로 부족
- 개인정보 보호, 데이터 개방 거부감

사회물리학 전문 기업은 없다

- 하지만 넷플릭스, 구글 포함 다수 기업들
사회물리학을 통한 기업의 발전 보여줌
- 기술과 경영, 마케팅 등 모든 영역을 총괄해서
꼭 필요한 학문

필요 기술

데이터 사이언스 스쿨

머신러닝, 딥러닝 실전 개발 입문

1절: 파이썬 설치와 사용법

2절: 파이썬 기초 문법

데이터 사이언스 스쿨에서 제공하는
커리큘럼으로 파이썬 기초 문법을 익
혔다.

4장: 파이썬을 이용한 데이터 분석

- 01.03.02.01 파이썬을 계산기로 사용하기
- 01.03.02.02 부동소수점 실수 자료형
- 01.03.02.03 파이썬으로 글자를 출력하기
- 01.03.02.04 파이썬의 문자열 형식화
- 01.03.02.05 파이썬 조건문 기초
- 01.03.02.06 파이썬 함수
- 01.03.02.07 파이썬 for 반복문
- 01.03.02.08 파이썬에서 여러 개의 자료를 한 변수
에 담기
- 01.03.02.09 파이썬에서 리스트 자료형 다루기
- 01.03.02.10 리스트와 반복문을 사용하여 계산하기
- 01.03.02.11 파이썬에서 딕셔너리 자료형 다루기
- 01.03.02.12 파이썬 객체지향 프로그래밍
- 01.03.02.13 파이썬 패키지 사용하기
- 01.03.02.14 파이썬의 자료형
- 01.03.02.15 파이썬의 문자열 인코딩
- 01.03.02.16 파이썬에서 날짜와 시간 다루기

파이썬을 이용한 머신러닝, 딥러닝 실전개발 입문

웹 크롤링과
스크레이핑부터
머신러닝·딥러닝까지
체계적으로 배우기

주지환 히크루즈연구소 지음
/
윤인성 옮김

DS 데이터 사이언스 시리즈 .003

이 책의 커리큘럼을 따라
머신 러닝을 공부했다.



히크루즈

1



머신러닝, 딥러닝 실전 개발 입문 0강 - 책 소개와 Docker 설치

윤인성

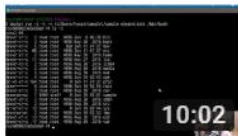
2



머신러닝, 딥러닝 실전 개발 입문 1강 - Docker 환경 구성

윤인성

3



머신러닝, 딥러닝 실전 개발 입문 2강 - 웹에서 데이터 가져오기

윤인성

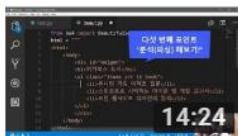
4



머신러닝, 딥러닝 실전 개발 입문 3강 - GET 요청 기본

윤인성

5



머신러닝, 딥러닝 실전 개발 입문 4강 - BeautifulSoup 기본

윤인성

머신러닝, 딥러닝 실전 개발 입문 5강 - BeautifulSoup 활용

목차

▣ 00장: 머신러닝을 위한 데이터 처리

0-1. 크롤링, 스크레이핑, 머신러닝

▣ 01장: 크롤링과 스크레이핑

1-1. 데이터 다운로드하기

1-2. BeautifulSoup로 스크레이핑하기

1-3. CSS 선택자

1-4. 링크에 있는 것을 한꺼번에 내려받기

▣ 02장: 고급 스크레이핑

2-1. 로그인에 필요한 사이트에서 다운받기

2-2. 웹 브라우저를 이용한 스크레이핑

2-3. 웹 API로 데이터 추출하기

2-4. cron을 이용한 정기적인 크롤링

▣ 03장: 데이터 소스의 서식과 가공

3-1. 웹의 다양한 데이터 형식

3-2. 데이터베이스

▣ 04장: 머신러닝

4-1. 머신러닝이란?

4-2. 머신러닝 첫걸음

4-3. 이미지 내부의 문자 인식

4-4. 외국어 문장 판별하기

4-5. 서포트 벡터 머신(SVM)

4-6. 랜덤 포레스트

4-7. 데이터를 검증하는 방법

▣ 05장: 딥러닝

5-1. 딥러닝 개요

5-2. TensorFlow 설치하기

5-3. Jupyter Notebook

5-4. TensorFlow 기본

5-5. TensorBoard로 시각화하기

5-6. TensorBoard로 딥러닝하기

5-7. Keras로 다양한 딥러닝 해보기

5-8. Pandas/NumPy 다루기

▣ 06장: 텍스트 분석과 챗봇 만들기

6-1. 한국어 분석(형태소 분석)

6-2. Word2Vec으로 문장을 벡터로 변환하기

6-3. 베이지 정리로 텍스트 분류하기

6-4. MLP로 텍스트 분류하기

6-5. 문장의 유사도를 N-gram으로 분석하기

6-6. 마르코프 체인과 LSTM으로 문장 생성하기

6-7. 챗봇 만들기

▣ 07장: 이미지와 딥러닝

7-1. 유사 이미지 검출하기

7-2. CNN으로 Caltech 101의 이미지 분류하기

7-3. 규동 메뉴 이미지 판정하기











7-4. OpenCV로 얼굴 인식하기

7-5. 이미지 OCR - 연속된 문자 인식하기

웹페이지에서 정보 추출하기

데이터 스크래핑



| | | | |
|---|-------------|---|-------------|
| · '유가 상승세 언제까지?'  | 10,28 15:32 | · '유가 상승세 언제까지?'  | 10,28 15:32 |
| · '유가 상승세 언제까지?'  | 10,28 15:32 | · '유가 상승세 언제까지?'  | 10,28 15:32 |
| · '유가 상승세 언제까지?'  | 10,28 15:32 | · '유가 상승세 언제까지?'  | 10,28 15:32 |
| · "국제유가 변동성, 당분간 이어질 것...살하  | 10,28 12:37 | · 한국은행 "국제유가 높은 변동성 당분간 지  | 10,28 12:01 |
| · 한은 "국제유가 상승·하락 요인 혼재...변동  | 10,28 12:00 | · 한은 "국제유가 변동성 당분간 높을 전망"  | 10,28 12:00 |

| 국내 시장 금리 | | |
|-------------|---------|----------|
| CD (91일) | 1.70 | - 0.00 |
| 콜 금리 | 1.54 | ▼ 0.02 |
| 국고채 (3년) | 1.97 | ▼ 0.01 |
| 회사채 (3년) | 2.41 | ▼ 0.02 |
| COFIF 잔액 | 1.90 | ▲ 0.01 |
| COFIF 신규취급액 | 1.83 | ▲ 0.03 |
| <hr/> | | |
| 달러 인덱스 | 96.1200 | ▼ 0.3300 |
| <hr/> | | |

The screenshot displays the Chrome DevTools interface, focusing on the Elements, Console, and Properties panels. The Elements panel shows the DOM tree, with a red box highlighting a specific HTML element: a `div` with class `head_info point_up` containing a `span` with class `value` and the text `1,142.50`. The Console panel shows the `element.style` object, and the Properties panel shows the `margin` and `border` properties. The bottom of the image shows a 'Highlights from the Chrome 68 update' section with three items: 'Eager evaluation', 'Argument hints', and 'Function autocompletion'.

Eager evaluation
Preview return values in the Console without explicitly executing expressions.

Argument hints
View a function's expected arguments in the Console.

Function autocompletion
View a function's expected arguments in the Console.

BeautifulSoup이용해서 html데이터 접근 css선택자 이용하여 데이터 추출

```
In [27]: import urllib.request
from bs4 import BeautifulSoup

url = "https://finance.naver.com/marketindex/"
response = urllib.request.urlopen(url)

soup = BeautifulSoup(response, "html.parser")
results = soup.select("span.value")

print("달러 : ", results[0].string)
print("엔 : ", results[1].string)
print("유로 : ", results[2].string)
```

```
달러 : 1,142.50
엔 : 1,018.86
유로 : 1,296.17
```

[고든 정의 TECH+] 진흙으로 짓는 1000달러 3D 프린터 주택 서울신문 | 10+

[고든 정의 TECH+] 금속 3D 프린터로 항공모함 보급 문제 해결한다 서울신문 | 30+

[고든 정의 TECH+] 인텔 9세대 프로세서 출시 - 인텔 8코어 CPU의 대중화 서울신문 | 100+

아하! 우주 연재보기

[아하! 우주] 세기의 '우주 중계방송' 시작...놓치지 마세요!

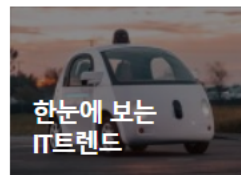
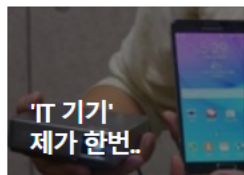
NASA의 인사이트 화성 착륙선이 화성 표면에서 촬영하는 사상급 (출처: NASA / JPL-Caltech) 화성 지진탐사선인
사이트 화성에 착지한다오늘부터 막
서울신문

[아하! 우주] 우주서 가장 흔한

[아하! 우주] 가방만한 꼬마위성

[아하! 우주] 수성가는데 왜 7년...
서울신문

이슈별 보기



쌀쌀한데 차 한잔 어떠세요?



몸도 마음도 따뜻하게
향기로운 티타임~

AD 네이버 쇼핑

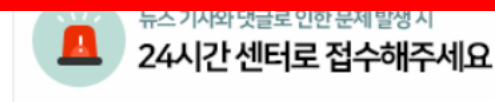
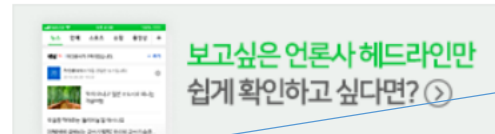
더 알아보기



박정호 "SKT, 5G 상용서비스 한달 앞당긴 내년 2월"

통신3사중 장비 구축 선두 서비스 조기 출시 자신감 SK텔레콤이 5G 상용서비스를 ...

디지털타임스 | 1시간전



뉴스토픽

| 뉴스 | 연예/스포츠 |
|----------------------------------|--------|
| 1 사이판 관광객 | |
| 2 비건 美대북대표 | |
| 3 금산둘레 300리 길 | |
| 4 이재명 촛불정부의 경찰 맞나 | |
| 5 창원 어린이집 원장 | |
| 6 비무장화 공동검증 완료 | |
| 7 석문산단에 600억 투자 | |
| 8 논산시 우수 정책 | |
| 9 당첨자는 2명 | |
| 10 슈뢰더 전 독일총리 | |
| ① 2018.10.28. 17:30 ~ 20:30 기준 > | |

Elements Console Sources

"nclicks(rig.newstopic)" target="_blank" title="사이판 관광객">

<strong class="title">사이판 관광객 == \$0

...

<strong class="title">비건 美대북대표

container #newstopic_news li a strong.title

24 of 40 Cancel

Styles Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls +

element.style {

}

.newstopic_common.css:1106

list li .title {

font-weight: normal;

vertical-align: top;

}

a strong { common.css:25

letter-spacing: -1px;

}

strong user agent stylesheet

g, b {

font-weight: bold;

}

Inherited from a.nclicks(rig...

newstopic_common.css:1001

Console What's New X

Highlights from the Chrome 68 update

Eager evaluation

Preview return values in the Console without explicitly executing expressions.

Argument hints

View a function's expected arguments in the Console.

Function autocompletion

margin -

border -

padding -

auto x auto

Filter Show all

border-collap... separate

color rgb(3...

cursor pointer

display inline

font-family "Helvet...

```
In [31]: import urllib.request
from bs4 import BeautifulSoup

url = "http://news.naver.com/main/main.nhn?mode=LSD&mid=shm&sid1=105"
response = urllib.request.urlopen(url)

soup = BeautifulSoup(response, "html.parser")
results = soup.select("strong")
#results = soup.select_one()
for result in results:
    print(result.string)
```

10

28

자동 추출

사이판 관광객

비건 美대북대표

금산둘레 300리 길

이재명 촛불정부의 경찰 만나

창원 어린이집 원장

비무장화 공동검증 완료

석문산단에 600억 투자

논산시 우수 정책

당첨자는 2명

슈뢰더 전 독일총리

1박2일 故 김주혁

집사부일체 이문세

북면가왕 프랑켄슈타인 쇼리

북면가왕 조현영

주말사용설명서 이세영

1박2일 차태현

한정수 1박2일

궁민남편 차인표

따로 또 같이 김한길

하나뿐인 내편 이장우

개인정보처리방침

**마찬가지 방법으로
네이버 실시간 인기 뉴스 헤드라인을
추출했다.**

In [21]: # 뉴스목록

```
In [34]: import urllib.request
from bs4 import BeautifulSoup

url = "http://news.naver.com/main/main.nhn?mode=LSD&mid=shm&sid1=105"
response = urllib.request.urlopen(url)

soup = BeautifulSoup(response, "html.parser")
results = soup.select("#newstopic_news strong")
#results = soup.select_one()
for result in results:
    print(result.string)
```

사이판 관광객
비건 美대북대표
금산둘레 300리 길
이재명 촛불정부의 경찰 맞나
창원 어린이집 원장
비무장화 공동검증 완료
석문산단에 600억 투자
논산시 우수 정책
당첨자는 2명
슈뢰더 전 독일총리

In []:

```

import urllib.request
from bs4 import BeautifulSoup

url = "http://news.naver.com/main/main.nhn?mode=LSD&mid=shm&sid1=105"
response = urllib.request.urlopen(url)

soup = BeautifulSoup(response, "html.parser")
results = soup.select("#newstopic_news a")
#results = soup.select_one()
for result in results :
    print(result.attrs["title"])
    print(result.attrs["href"])

```

사이판 관광객

https://search.naver.com/search.naver?where=nexearch&query=%EC%82%A%EC%9D%B4%ED%8C%90+%EA%B4%80%EA%B4%91%EA%B0%9D&ie=utf8&sm=nws_htk.nws

비건 美대북대표

https://search.naver.com/search.naver?where=nexearch&query=%EB%B9%B4%EA%B1%B4+%E7%BE%8E%EB%8C%80%EB%B6%B1%EB%8C%80%ED%91%9C&ie=utf8&sm=nws_htk.nws

금산둘레 300리 길

https://search.naver.com/search.naver?where=nexearch&query=%EA%B8%B8%EC%82%B0%EB%91%98%EB%A0%B8+300%EB%A6%AC+%EA%B8%B8&ie=utf8&sm=nws_htk.nws

이재명 촛불정부의 경찰 맞나

https://search.naver.com/search.naver?where=nexearch&query=%EC%9D%B4%EC%9E%A%EB%AA%B5+%EC%B4%9B%EB%B6%B8%EC%A0%95%EB%B6%80%EC%9D%98+%EA%B2%BD%EC%B0%B0+%EB%A7%9E%EB%B2%98&ie=utf8&sm=nws_htk.nws

창원 어린이집 원장

https://search.naver.com/search.naver?where=nexearch&query=%EC%B0%BD%EC%9B%90+%EC%96%B4%EB%A6%B0%EC%9D%B4%EC%A7%91+%EC%9B%90%EC%9E%A5&ie=utf8&sm=nws_htk.nws

비무장화 공동검증 완료

https://search.naver.com/search.naver?where=nexearch&query=%EB%B9%B4%EB%AC%B4%EC%9E%A5%ED%99%94+%EA%B3%B5%EB%8F%99%EA%B2%80%EC%A6%9D+%EC%99%84%EB%A3%8C&ie=utf8&sm=nws_htk.nws

석문산단에 600억 투자

https://search.naver.com/search.naver?where=nexearch&query=%EC%84%9D%EB%AC%B8%EC%82%B0%EB%B8%A8%EC%97%9D+600%EC%96%B5+%ED%88%A%EC%9E%90&ie=utf8&sm=nws_htk.nws

논산시 우수 정책

https://search.naver.com/search.naver?where=nexearch&query=%EB%B5%BC%EC%82%B0%EC%88%9C+%EC%9A%B0%EC%88%98+%EC%A0%95%EC%B1%B5&ie=utf8&sm=nws_htk.nws

당첨자는 2명

https://search.naver.com/search.naver?where=nexearch&query=%EB%B8%B9%EC%B2%A8%EC%9E%90%EB%8A%94+2%EB%AA%B5&ie=utf8&sm=nws_htk.nws

슈뢰더 전 독일총리

https://search.naver.com/search.naver?where=nexearch&query=%EC%8A%B8%EB%A2%B0%EB%8D%94+%EC%A0%B4+%EB%8F%B5%EC%9D%BC%EC%B4%9D%EB%A6%AC&ie=utf8&sm=nws_htk.nws

```

13]: #모듈 추출
import urllib.request
from bs4 import BeautifulSoup
#기사목록을 가져옵니다

url = "https://news.naver.com/main/home.nhn"
response = urllib.request.urlopen(url)

soup = BeautifulSoup(response, "html.parser")

results = soup.select("#section_politics strong")
results2 = soup.select("#section_politics a")

for i in range(len(results)):
    print(results[i].string)
    print(results2[i+6].attrs["href"])
|

```

[단독] 政-靑 일자리質 개선 큰소리 친 고용보험 통계 '과대포장'됐다

<https://news.naver.com/main/read.nhn?mode=LSD&mid=shm&sid1=100&oid=014&aid=0004116919>

“군대는 선교의 가두리 양식장” ··· 종교 강요금지 요구도 [박성진의 군이야기]

<https://news.naver.com/main/read.nhn?mode=LSD&mid=shm&sid1=100&oid=032&aid=0002902193>

[팩트체크] '하늘의 별 따기' 법관 탄핵…일본은 국민도 탄핵 청구

<https://news.naver.com/main/read.nhn?mode=LSD&mid=shm&sid1=100&oid=001&aid=0010432877>

이언주 “처방 완전 거꾸로…文정부, 한국경제 자살로 몰아가는 중”

<https://news.naver.com/main/read.nhn?mode=LSD&mid=shm&sid1=100&oid=020&aid=0003177600>

[국감] "경제위기 아니다"…정부, '정책 신뢰' 호소(종합)

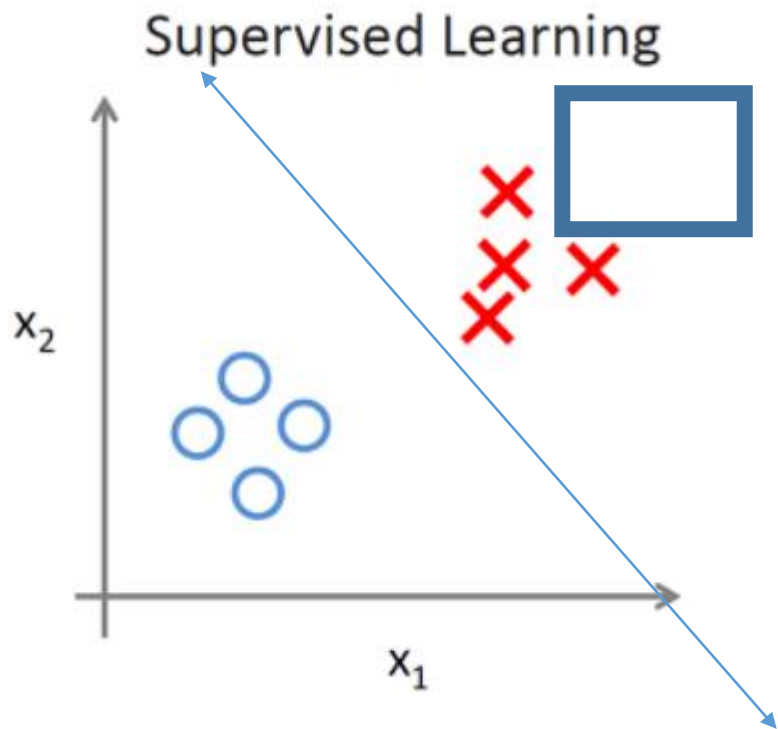
<https://news.naver.com/main/read.nhn?mode=LSD&mid=shm&sid1=100&oid=008&aid=0004124792>

데이터 스크레이핑을 배웠어요.

머신러닝

수많은 데이터를 학습시켜 거기에 있는 패턴을 찾아내는 것

특징량



1. 0, X가 모여 있는 위치관계
“특징량” 확인

2. 특징량을 기준으로 구분선 긋기

머신러닝은 계산을 통해 구분선을 찾아내는 것

특징 추출

- 붓꽃의 종류를 구별하는 프로그램을 만든다고 할 때
- 어떤 특징을 사용하여 데이터를 구별할지 찾고 이를 벡터로 변환해야 함.
- 어떤 특징을 추출할지는 프로그래머가 정해야 함

학습의 종류 - 교사학습

- 교사학습

- 명시적인 답이 주어지면서 컴퓨터를 학습

- (데이터, 레이블)의 형태

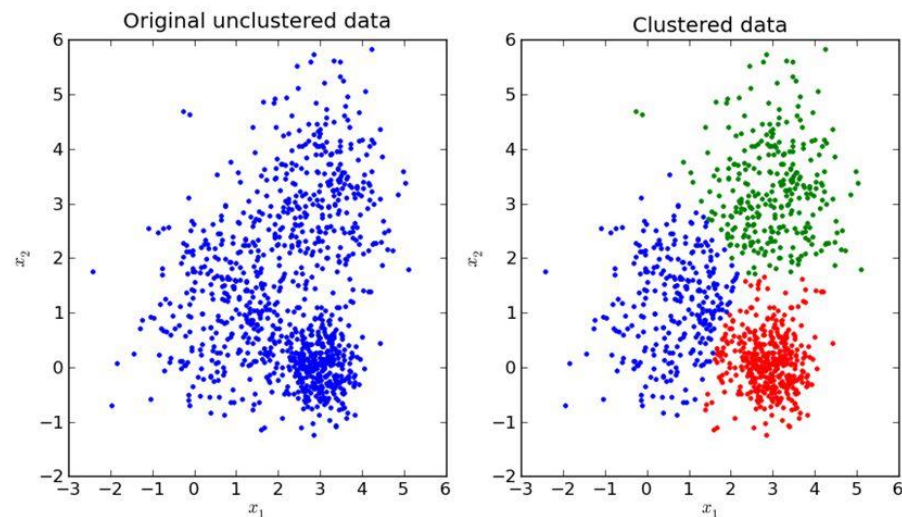


학습의 종류 - 비교사학습

- (데이터)의 형태
- Ex. 데이터를 유사한 특징을 가진 세 가지로 묶는
"클러스터링"

Unsupervised Learning

- 데이터의 숨겨진 특징이나 구조



학습의 종류 - 강화학습

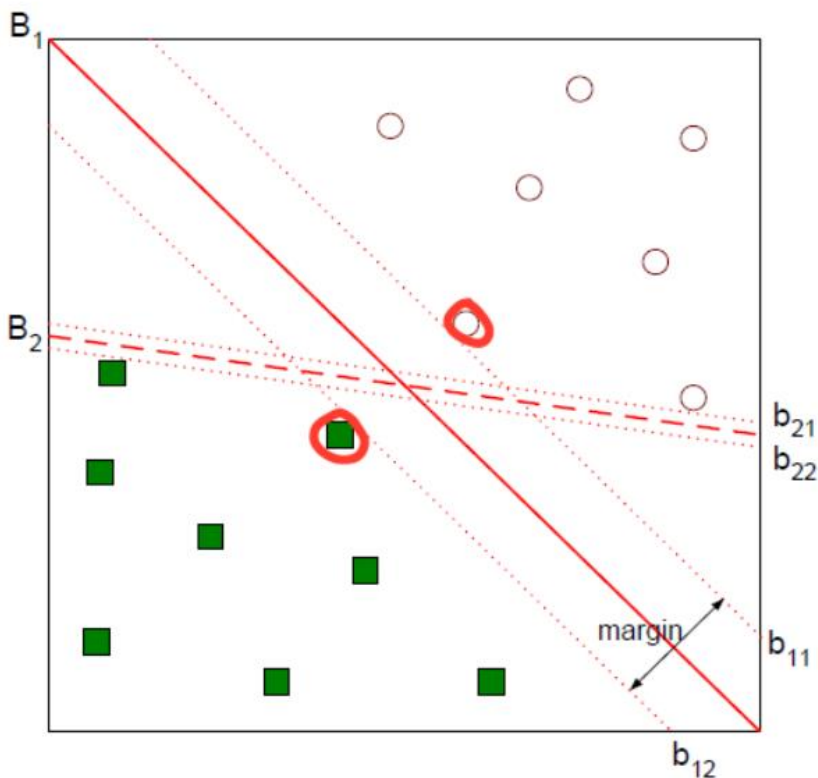
- 에이전트가 주어진 환경에 대해 행동을 취하고 보상을 얻으면서 진행



Reinforcement Learning Setup

SVM 알고리즘

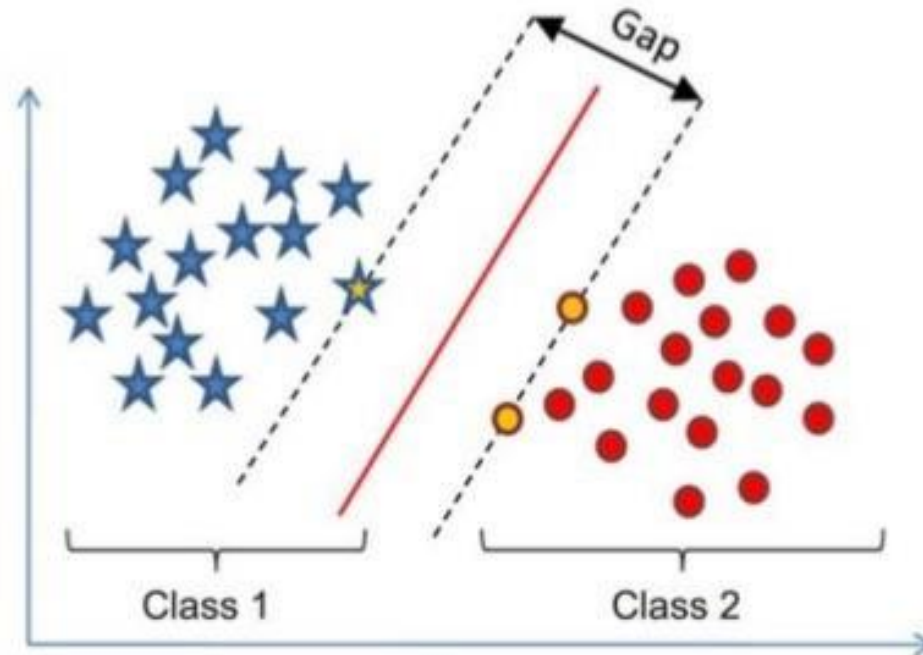
Support Vector Machine



SVM의 특징

- ✓ 선을 구성하는 매개변수를 조정해서 요소들을 구분하는 선을 찾고 패턴을 인식하는 방법
- ✓ “식별 평면” – 패턴의 경계
- ✓ 마진 최대화
- ✓ “일반화 능력“

Basic concept of SVM



Find a linear decision surface ("hyperplane") that can separate classes and has the largest distance (i.e., largest "gap" or "margin") between border-line patients (i.e., "support vectors")

머신러닝의 3단계

학습하기

예측하기

평가하기

붓꽃의 품종 구별

Scikit-learn 라이브러리 이용

| A | B | C | D | E |
|-------------|------------|-------------|------------|-------------|
| SepalLength | SepalWidth | PetalLength | PetalWidth | Name |
| 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| 4.9 | 3 | 1.4 | 0.2 | Iris-setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| 5 | 3.6 | 1.4 | 0.2 | Iris-setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | Iris-setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | Iris-setosa |
| 5 | 3.4 | 1.5 | 0.2 | Iris-setosa |

Iris.csv

150개의 데이터

꽃받침의 길이, 넓이
꽃잎의 길이 넓이

꽃의 품종과 연관이 있을 것!

붓꽃의 품종 구별

Scikit-learn 라이브러리 import

```
1  from sklearn import svm, metrics
2  import random, re
```


붓꽃의 품종 구별

```
4 csv = []
5 with open('iris.csv', 'r', encoding='utf-8') as fp:
6     # 한 줄씩 읽어 들이기
7     for line in fp:
8         line = line.strip() # 줄바꿈 제거
9         cols = line.split(',') # 쉼표로 자르기
10        # 문자열 데이터를 숫자로 변환하기
11        fn = lambda n : float(n) if re.match(r'^[0-9\.]+'$, n) else n
12        cols = list(map(fn, cols))
13        csv.append(cols)
14    # 가장 앞 줄의 헤더 제거
15    del csv[0]
16    # 데이터 셔플하기(섞기) --- (※2)
17    random.shuffle(csv)
```

정규표현식을 이용해서 셀 내용이 숫자인지 확인

리스트에 집어 넣기

붓꽃의 품종 구별

```
19 total_len = len(csv)
20 train_len = int(total_len * 2 / 3)
21 train_data = []
22 train_label = []
23 test_data = []
24 test_label = []
25 for i in range(total_len):
26     data = csv[i][0:4]
27     label = csv[i][4]
28     if i < train_len:
29         train_data.append(data)
30         train_label.append(label)
31     else:
32         test_data.append(data)
33         test_label.append(label)
```

전체 2/3 (100개) 학습 전용 데이터로 설정

나머지는 테스트 전용 데이터로 설정

붓꽃의 품종 구별

| | | |
|----|---|-------|
| 35 | <code>clf = svm.SVC()</code> | 학습하기 |
| 36 | <code>clf.fit(train_data, train_label)</code> | |
| 37 | <code>pre = clf.predict(test_data)</code> | |
| 38 | <code># 정답률 구하기 --- (※5)</code> | 예측하기 |
| 39 | <code>ac_score = metrics.accuracy_score(test_label, pre)</code> | |
| 40 | <code>print("정답률 =", ac_score)</code> | 채점 매기 |

붓꽃의 품종 구별 결과

```
(test label, pre) : ( virginica , virginica )
(test label, pre) : ( "Versicolor" , "Versicolor" )
(test label, pre) : ( "Versicolor" , "Versicolor" )
(test label, pre) : ( "Versicolor" , "Versicolor" )
(test label, pre) : ( "Virginica" , "Virginica" )
(test label, pre) : ( "Setosa" , "Setosa" )
(test label, pre) : ( "Setosa" , "Setosa" )
(test label, pre) : ( "Virginica" , "Virginica" )
(test label, pre) : ( "Setosa" , "Setosa" )
(test label, pre) : ( "Versicolor" , "Versicolor" )
(test label, pre) : ( "Versicolor" , "Versicolor" )
(test label, pre) : ( "Versicolor" , "Versicolor" )
(test label, pre) : ( "Versicolor" , "Versicolor" )
(test label, pre) : ( "Versicolor" , "Versicolor" )
(test label, pre) : ( "Setosa" , "Setosa" )
(test label, pre) : ( "Versicolor" , "Virginica" )
(test label, pre) : ( "Virginica" , "Virginica" )
(test label, pre) : ( "Versicolor" , "Versicolor" )
(test label, pre) : ( "Virginica" , "Virginica" )
(test label, pre) : ( "Versicolor" , "Versicolor" )
(test label, pre) : ( "Setosa" , "Setosa" )
(test label, pre) : ( "Versicolor" , "Versicolor" )
(test label, pre) : ( "Setosa" , "Setosa" )
(test label, pre) : ( "Setosa" , "Setosa" )
(test label, pre) : ( "Virginica" , "Virginica" )
(test label, pre) : ( "Setosa" , "Setosa" )
```

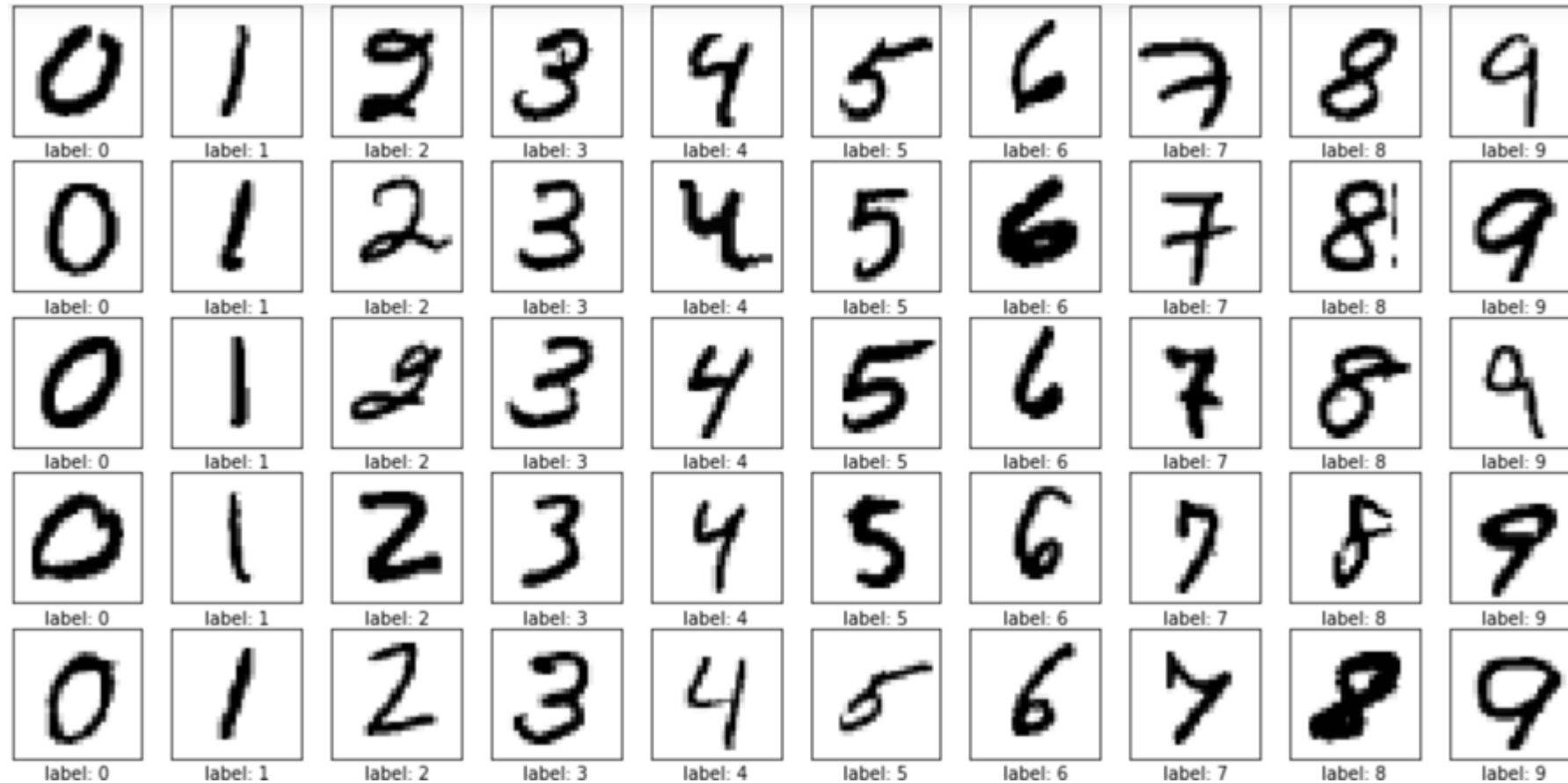
정답률 = 0.96

시뮬레이션을 돌려본 결과

정답률 96%

이미지 내부의 문자 인식

MNIST - 손글씨 인식하기



[illegible]

```

from sklearn import model_selection, svm, metrics
# CSV 파일을 읽어 들이고 가공하기 --- (※1)
def load_csv(fname):
    labels = []
    images = []
    with open(fname, "r") as f:
        for line in f:
            cols = line.split(",")
            if len(cols) < 2: continue
            labels.append(int(cols.pop(0)))
            vals = list(map(lambda n: int(n) / 256, cols))
            images.append(vals)
    return {"labels":labels, "images":images}
data = load_csv("./mnist/train.csv")
test = load_csv("./mnist/t10k.csv")

# 학습하기 --- (※2)
clf = svm.SVC()
clf.fit(data["images"], data["labels"])

# 예측하기 --- (※3)
predict = clf.predict(test["images"])

# 결과 확인하기 --- (※4)
ac_score = metrics.accuracy_score(test["labels"], predict)
cl_report = metrics.classification_report(test["labels"], predict)
print("정답률 =", ac_score)
print("리포트 =")
print(cl_report)

```

데이터 : 붓글씨 사진을 픽셀로 나눠 해당 픽셀의 검정색 농도를 수치화 한 것

핵심 코드

```
# 학습하기 --- (※2)
clf = svm.SVC()
clf.fit(data["images"], data["labels"])

# 예측하기 --- (※3)
predict = clf.predict(test["images"])

# 결과 확인하기 --- (※4)
ac_score = metrics.accuracy_score(test["labels"], predict)
```

정답률 = 0.7884231536926147

리포트 =

| | precision | recall | f1-score | support |
|--|-----------|--------|----------|---------|
|--|-----------|--------|----------|---------|

| | | | | |
|---|------|------|------|----|
| 0 | 0.87 | 0.93 | 0.90 | 42 |
|---|------|------|------|----|

| | | | | |
|---|------|------|------|----|
| 1 | 0.81 | 1.00 | 0.89 | 67 |
|---|------|------|------|----|

| | | | | |
|---|------|------|------|----|
| 2 | 0.84 | 0.69 | 0.76 | 55 |
|---|------|------|------|----|

| | | | | |
|---|------|------|------|----|
| 3 | 0.87 | 0.57 | 0.68 | 46 |
|---|------|------|------|----|

| | | | | |
|---|------|------|------|----|
| 4 | 0.76 | 0.75 | 0.75 | 55 |
|---|------|------|------|----|

| | | | | |
|---|------|------|------|----|
| 5 | 0.63 | 0.80 | 0.71 | 50 |
|---|------|------|------|----|

| | | | | |
|---|------|------|------|----|
| 6 | 0.97 | 0.67 | 0.79 | 43 |
|---|------|------|------|----|

| | | | | |
|---|------|------|------|----|
| 7 | 0.74 | 0.86 | 0.79 | 49 |
|---|------|------|------|----|

| | | | | |
|---|------|------|------|----|
| 8 | 0.91 | 0.72 | 0.81 | 40 |
|---|------|------|------|----|


| | | | | |
|---|------|------|------|----|
| 9 | 0.71 | 0.81 | 0.76 | 54 |
|---|------|------|------|----|

| | | | | |
|-------------|------|------|------|-----|
| avg / total | 0.80 | 0.79 | 0.79 | 501 |
|-------------|------|------|------|-----|

데이터양을 늘려주면 78%
보다 높은 정확도 얻을 수
있음

Star cluster Simulation

We use cookies on kaggle to deliver our services, analyze web traffic, and improve your experience on the site. By using kaggle, you agree to our use of cookies. [Got it](#) [Learn more](#)

 Dataset

Star Cluster Simulations

Direct N-body simulation of a star cluster: Position and velocities of stars

 Mario Pasquato • updated 2 years ago (Version 1)

75 voters

share

[Data](#) Overview Kernels (20) Discussion (2) Activity Download (41 MB) [New Kernel](#)

Kaggle에서 성운의 별들이 움직이는 데이터 얻음

Data (41 MB)

[API](#) `kaggle datasets download -d mariopasquato/star-c...` [Download All](#)

| Data Sources | About this file | Columns |
|---|--|---|
| <div><div>c_0000.csv</div>8 columns</div> <div><div>c_0100.csv</div>8 columns</div> <div><div>c_0200.csv</div>8 columns</div> <div><div>c_0300.csv</div>8 columns</div> <div><div>c_0400.csv</div>8 columns</div> <div><div>c_0500.csv</div>8 columns</div> <div><div>c_0600.csv</div>8 columns</div> <div><div>c_0700.csv</div>8 columns</div> <div><div>c_0800.csv</div>8 columns</div> | <div>Initial conditions. Col. 1, 2, 3: positions of stars; 4, 5, 6: velocities; 7: masses; 8: ids.</div> | <div># x</div> <div># y</div> <div># z</div> <div># vx</div> <div># vy</div> <div># vz</div> <div># m</div> <div># id</div> |

```
from vpython import*
import re
import os
import random

#file_list = os.listdir('C:\space')
star = {}

#for fhand in file_list :
#hand = open('space/'+fhand)
hand = open('space/c_0000.csv')

for line in hand :
    line = line.rstrip().split(',')
    # print(line[0])
    try :
        x = float(line[0])
        y = float(line[1])
        z = float(line[2])
        m = float(line[6])
        vx = float(line[3])
        vy = float(line[4])
        vz = float(line[5])

        i = int(line[7])
    except :
        continue
    if i > 20000 : break
    star[i] = sphere(pos=vec(x,y,z), radius=m*1000, color=color.white)
    star[i].velocity = vec(vx,vy,vz)
```

결과 ; 시간이 흐르면 별이 흩어짐(중력을 적용하지 않았기 때문)
하지만 성운이 회전하는 모습을 볼 수 있었음.

```
t = 0
#0.01
deltat = 0.1
print("start to move")
for testt in range(0,10000000000000000) :
    #st = random.choice(star)
    #st.pos += st.velocity*deltat
    for st in star :
        #print(star[st])
        rate(5000)
        star[st].pos += star[st].velocity*deltat
        #print("st.pos:"+str(star[st].pos))
        #print("st.vel:",star[st].velocity)
        #print("333333333333333333")

    t = t + deltat
```

참고 문헌 및 출처

참고 문헌 및 출처

- 물리학은 사회현상을 설명·예측할 수 있는가?, 사이언스온, 조향현, 2010.12.09, http://scienceon.hani.co.kr/?document_srl=33737, 2017.09.15
- 사회물리학, 위의 책, 정우성, 2011년 5월 제20권 5호
- 빅데이터 분석의 국내외 활용 현황과 시사점, 최재경, KISSTEP InI 제 14호 (2016.06)
- 모사현실을 통한 미래사회 탐색과 예측, 소아영, 융합연구정책센터 2017 SEPTEMBER vol.86 (2017.09.04)
- 빅데이터와 사회과학하기: 자료기반의 변화와 분석전략의 재구상, 한신갑, 한국사회학 제49집 제2호(2015년), pp.161~192
- [넷플릭스는 어떻게 작동하는가, 네이버레터, 박상현, <http://nter.naver.com/naverletter/110357>, 2017.12.01
- https://www.youtube.com/watch?v=ivf1l85pzw8&list=PLBXuLglnP-5m_vn9ycXHRI7hlsl1huqm5&index=6