

인공지능경제연구

AD20202002 이민지

1. 주차별 학습내용

<Overview>

1. 인공지능 개념 익히기	(3주차) 인공지능 개념	
	(2주차, 3주차) 인공지능 활용분야	
	(5주차) 분석이해하기(회귀와 분류)	
2. 인공지능 활용 실습	업무활용tool	(1주차) 에버노트, 단축키, 매크로, 구글스프레드시트 (11주차) github
	오렌지	(7주차, 8주차) 다양한 알고리즘 활용 분석 실습
	파이썬	(4주차, 6주차, 9주차) 코드작성 기초, 간단한 함수만 들기(반복작업구문)
	API활용	(10주차, 12주차) API로 데이터 다운로드

■ 1주차(2022.3.4.): 효율적인 업무수행

1) 에버노트 활용법

- 웹페이지 등에서 글과 그림을 손쉽게 복사하고 정리 할 수 있는 툴인 에버노트를 사용해봄.
- (유용성) 웹페이지에서 찾은 자료들은 손쉽게 스크랩할 수 있음.

2) 컴퓨터를 활용하여 업무 또는 학습을 할 때 도구와 유용한 윈도우/크롬 단축키 등을 학습함.

- 윈도우 단축키들을 활용하여 키보드만 가지고 빠르게 작업들을 수행 할 수 있음.
- 다양한 단축키를 실습해 보았는데 가장 유용하게 쓰는 것은 가상데스크탑을 생성 [win+ctrl+d], 이동[win+ctrl+좌우 방향키]로 여러 업무들을 진행 할 때 업무별로 필요한 프로그램과 창만 띄울 수 있고, 개인적인 업무와 공적인 업무를 할 때 분리할 수 있음.

<유용한 크롬 단축키>

단축키	기능
ctrl+n	새 창 열기
ctrl+shift+n	시크릿 창 열기
ctrl+t	새 탭열기
ctrl+w	탭 종료
ctrl+숫자키	해당 숫자 탭으로 넘어감, 9는 마지막 탭
Ctrl + Shift +t	최근에 닫은 탭 다시 열기

3) 매크로

- 단순 반복작업을 할 때, 매크로 프로그램을 활용하여 자동으로 실행하게 할 수 있음.
- 웹상의 아무 매크로 파일 다운 받으면 안됨.
- 보통 클릭 커서 위치 등을 바탕으로 반복적으로 움직이게 됨.

4) 구글 스프레드시트 등을 가지고 데이터 작업 쉽게 하기

- 구글 스프레드시트에 매크로 프로그램을 가지고 단순한 반복작업이 가능함.
- 웹에서 다양한 매크로 프로그램을 찾을 수 있는데 유명하거나 많이 쓰는 것을 활용하는 것이 보안 측면에서 중요함.
- 수업때 써본 매크로는 Pulover's 매크로로 활동을 좌표값과 입력값으로 기억하여 손쉽게 매크로를 만들 수 있음.
- 구글 스프레드시트는 일반 엑셀과 함수식 등은 유사하지만, 작업자가 수행하고 있는 작업을 학습하여 함수식을 똑같이 쓰지 않아도 추천해주거나, 미리 작업자가 하려는 것을 예측해 입력값을 제안해주는 등 일반 엑셀보다 효율적으로 업무를 할 수 있게 해줌.

(5) 구글 스프레드시트로 데이터 전처리

- 데이터 전처리를 할 때 도움이 되는, 특히 데이터 필드를 쉽게 조인할 수 있는 vlookup 함수를 배움.
- 예제로 pnu를 만들어보면서, vlookup 함수를 익히고 데이터 전처리 과정을 경험해봄. 데이터들을 살펴보고 데이터들을 규칙성을 갖게 하거나 같은 형태로 만들어주는 것이 중요함.
- 이상한 데이터들안에도 규칙들이 있을 수 있으며 오류를 만든 규칙을 확인하고 오류 그룹별로 문제를 차례로 해결해 나가야함.
- 데이터를 소실하지 않도록 작업시 늘 연번을 붙여서 진행하는 것이 좋음.

■ 2주차(2022.3.11.): 인공지능과 비즈니스

- 비즈니스 모델 구상 시 유의점: 비즈니스를 위한 수익성 고려가 필요함. 인공지능을 비즈니스 모델로서 인공지능을 활용하기 위해서는 좋은 아이디어와 기술도 중요하지만 결국 수익성을 고려하지 않을 수 없음.

■ 3주차(2022.3.18.): 인공지능의 활용분야

- 인공지능은 다양한 분야에서 활용되고 있으며, 매우 높은 수준으로 발달되어 있음.

1) 흥미로운 인공지능 기반 웹사이트/프로그램

(1) 구글 아트앤 컬처

- 전세계 미술관, 박물관의 작품들을 볼 수 있으며 단순 그림파일이 실제 작품들의 디테

일한 부분까지도 확대해서 볼 수 있음.

(2) BLOB OPERA

- 사용자가 음악적인 지식이 없더라도 제공되는 캐릭터를 가지고 화음을 만들 수 있음.

(3) 아키네이터

- 사용자가 특정 개념에 대해 응답한 결과들을 가지고 어떠한 개념을 생각했는지를 맞추는 프로그램인데, 분류의 개념을 이해하기 쉬웠음.
- 가장 크게 분류할 수 있는 질문들로 시작해 그 범주를 좁혀나가면서 특정 개념을 찾아내는 것임.

(4) 구글렌즈

- 과거에는 문자만을 입력하여 검색을 했지만 구글렌즈는 꼭 문자로 입력하지 않아도 이미지, 소리 등을 인식하여 검색, 기록을 가능하게 해줌.

(5) TEACHABLE MACHINE

- 인공지능을 간단하게 개발해볼 수 있는 웹사이트로 직접 이미지, 음성을 입력만 하면 머신러닝으로 인공지능을 구현해냄.

2) AI ML DL의 차이

- 인공지능은 머신러닝과 딥러닝을 모두 포함하는 개념으로 데이터를 학습하여 의사결정을 내리는 개념임.
- 이러한 인공지능을 실현하는 기능은 머신러닝과 딥러닝 간 차이가 있음. 머신러닝은 어떠한 특정 로직에 따라 데이터들을 분류하는 작업에 유용한데 어떠한 정답이나 규칙을 알려주고 데이터들을 학습시켜 분류나 회귀를 하는데 유용함.
- 또는 많은 데이터를 주고 머신러닝을 통해 규칙성을 찾아내는 작업도 가능함.
- 딥러닝은 머신러닝의 많은 알고리즘들을 조합하여 의사결정을 하는 어떠한 구조를 만들어내는 것 까지 가능함.
- 딥러닝은 우리가 알수 없는 어떠한 규칙성을 함수로 추정하고 근사치를 낼 수 있음.

예) 강아지나 고양이나 딥러닝은 픽셀정보로, 머신러닝은 수염길이 등으로

■ 4주차(2022.3.25.): 마크다운 문법 및 파이썬 기초 익히기

- 마크다운 문법은 특정 규칙을 가진 코드 작성법으로 코드를 입력하면 이를 반영하여 입력한 텍스트를 해당 코드에 맞게 시각화 해줌.
- 궁극적으로 파이썬도 여러 문법들 중 하나이며, 해당 문법에 따라 일련의 작업 프로세스를 작성하고 작동시킬 수 있음.
- 코드 작성시엔 온라인에 오픈되어 있는 코드들을 적극 활용할 필요가 있음.

- 코드를 어떻게 하면 간단하게 표현 할 수 있을지, 사고체계에 따라 작성해보면서 정리해 나가는 것이 중요함.
- 처음부터 완벽하게 코드를 짜려하지 말고 길어져도 하나씩 작성해보면서 조정하는 반복적인 과정임.
- * 사고체계: 하고자 하는 의도를 먼저 나열 한 후 어떻게 하면 해당 의도를 실현시킬 수 있을지 고민하기

■ 5주차(2022.4.1.): 손으로 회귀 풀기

1) 엑셀로 각 통계값들을 가지고 R^2 가 높게끔 값들을 예측 조정해봄

- 회귀모형의 일반적인 개념들을 익힘
- 해당 회귀모형의 모형적합도(얼마나 제대로 예측했는지)를 확인하는 지표로는 R^2 , SE(Squared Error), MSE(Mean of Squared Error)가 있음.
- R^2 식은 $1 - (\text{오차의 제곱합} / \text{편차의 제곱합})$ 임. 실제 데이터의 편차(데이터값-평균값) 대비 예측한 값들의 오차의 제곱합이 어느정도 수준인가를 의미함. 편차의 제곱합에서 오차의 제곱합을 뺀 개념이니, 모형이 얼마나 잘 설명해졌는지를 의미하는 것임.
- 통계의 기본적인 개념인 평균, 중위값, 최빈값등의 개념을 익히고,
- 회귀모형은 궁극적으로 함수식이며 어떠한 값들을 통계
- 회귀모형의 산식을 이해하고 데이터를 가지고 회귀모형을 만들어 봄.
- 중요한 것은 target에 영향을 주는 feature 변수들의 특성들을 파악하고 있어야 결측값 등 더 좋은 추정을 위해 값조정을 할 수 있음.
- 레몬에이드 판매 데이터를 가지고 대략적인 계수값들을 조정해보면서 R^2 를 높이는 실습을함. 이를 통해 R^2 산식을 보다 더 잘 이해하고 target에 각 feature들이 어떤 영향을 주는지까지 살펴 볼 수 있음.

2) 예측의 지표-confusion matrix

- 예측이 잘 되었는지를 살펴볼 수 있는 여러 지표가 있음.

accuracy, 정확도	$(TP + TN) / \text{total}$	전체중에 제대로 맞춘것
sensitivity, TPR, recall, 민감도	$TP / (TP + FN)$	진짜*인것들 중 제대로 맞춘 것 *진짜: 실제 true, 예측 positive + 실제 false, 예측 negative
precision, PPV	$TP / (TP + FP)$	positive라고 예측한 것 중에 진짜인 것

- 각 지표의 차이를 이해하고 분석 목표와 데이터 특성을 고려해 적절한 예측 지표를 갖고 해당 예측모형의 설명력을 논해야함.

■ 6주차(2022.4.8.): 로또 만들기

- 랜덤으로 숫자 여섯 개 출력해주는 프로그램을 만들어봄.

- 1) random으로 숫자 뽑는 코드 import random, 실행(random.random())
- 2) 숫자범위 정해주기: list로 정해줌(1~46을 다 열거하지 않으려면: range(1,46))
- 3) 중복이 있으면 안되므로 뽑은 것 중에 있으면 다시 continue하게끔.
- 4) 6개 숫자 뽑을 때까지 반복, 6개 되면 append해서 결과 출력

```
[11] lst = []
    while len(lst)<6:
        li = list(range(1, 46))
        e = random.choice(li)
        if e in lst:
            continue
        else:
            lst.append(e)
    print(lst)
```

5) 하나의 함수로 정의해서 활용할수도 있음.

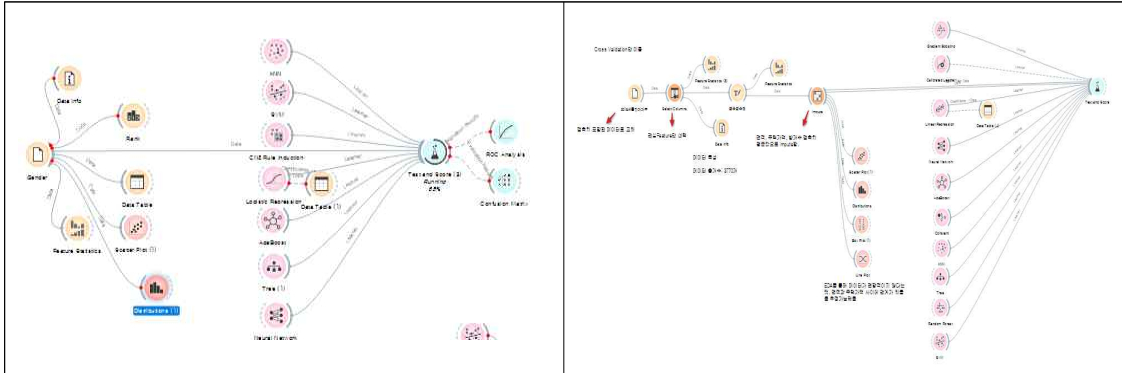
```
def lotto(n, m):
    lst = []
    while len(lst)<m:
        li = list(range(1, n))
        e = random.choice(li)
        if e in lst:
            continue
        else:
            lst.append(e)
    lst.sort()
    return lst
```

```
lotto(100000, 10)
```

■ 7주차(2022.4.15.): 오렌지 사용법

- 오렌지라는 인공지능 데이터 분석 툴을 익힘.
 - 별도의 코딩없이 알고리즘 및 분석방법들이 UI형태로 되어있어 손쉽게 데이터 분석을 할 수 있음.
 - 데이터를 불러오는 법, 시각화 하는 법, 알고리즘 사용법 확인
- <유용한 기능>
- Select Column: 분석하고 싶은 필드만 선택가능
 - Impute로 결측치 값을 평균/최빈값/중위값 등으로 변경 가능

■ 8주차(2022.4.22.): 중간고사: 오렌지를 활용하여 분류, 회귀 작업을 해봄.



■ 9주차(2022.4.29.): 코드맵등 코드를 짜는데 유념해야할 것

-code map을 통해 code 단순화 방법을 이해함.

■ 10주차(2022.5.6.): 중간고사 피드백 및 공공API 활용법 학습

1) 오렌지

- 중간고사인 오렌지를 활용한 분류, 회귀 분석 수행 결과를 발표함.
- 공통적으로 중요한 포인트는,

- (1) 데이터를 이해하는 것: 기초통계 및 시각화 알고리즘을 통해 해당 데이터들의 특성들을 상세히 파악하고 어떠한 알고리즘이 적합할지 등을 고민해야함.
- (2) 데이터를 통해 어떤 것을 알고 싶은지, 그 주제가 어떤 의미를 갖는지등을 고민하고 프로젝트 계획을 세워야 함.
- (3) 오렌지에서 제공하는 각 기능들, 선택들을 이해하고 활용하는 것이 필요함.

예) cross validation 내 Stratified의 의미: stratified는 데이터에 특정 계층의 비율이 치우쳐져있을 때, 각 클래스들의 비율을 고려하여 검증 데이터 set을 구성하는 것을 의미함.

<오렌지 내 알고리즘>

알고리즘	설명
Gradient Boosting	어떠한 예측을 하고 오차를 다시 예측하면서 모형을 만드는 방법
Calibrated Learner	각 (분류)계층의 확률분포를 측정해서 결정기준을 최적화하는 방법.
Linear Regression	투입데이터로부터 선형함수를 구성하여 feature와 target 변수의 관계를 확인하는 방법
Neural Network	선형관계 뿐 아니라 비선형관계에 있는 변수간 관계를 확인할 수 있는 방법으로 데이터로부터 최적화된 함수를 만드는데 함수의 기울기를 구하고 기울기가 낮은 극값으로 이동시키며 최적화를 하는 방법
AdaBoost	관측치에 가중치를 더하면서 분류하기 어려운 데이터에 가중치를 더하고 이미 잘 분류된 데이터에는 가중치를 적게 부여하여 학습을 시키는 방법
Constant	트레이닝 데이터에서 평균값 또는 최빈값을 예측하는 방법

kNN	가장 근사한 트레이닝 데이터를 예측하는 방법
Tree	target 변수로부터 얻은 정보를 통해 얻은 분류값에 따라 데이터를 계속 나눠 분류하는 방법.
Random Forest	의사결정트리를 만들어 분류, 회귀를 하는 알고리즘으로 가장 잘 설명할 수 있는 변수들로 임의의 하위변수 데이터 set을 트리로 구성하는 방법.
SVM	초평면에 따라 속성들을 분류하여 다른 클래스(종류), 분류 값들 간 차이를 극대화하여 분류하는 방법
CN2 rule inducer	어떠한 규칙들에 따라 분류를 하게 하게 하는 방법
Logistic Regression	각 feature변수에 따라 분류해야될 target변수 결정할 확률을 계산하여 함수식으로 구성하여 분류하는 방법.

2) 공공API

- API는 직접 데이터파일을 다운받지 않고 특정 규칙에 따라 필요한 데이터들을 신청하고 제공받는 것임.
- API는 데이터를 허가 받고 사용하는 개념이므로 프로젝트 키 등 보안키 입력 등이 중요하며, 작성방법을 이해해야 활용할 수 있음.
- 데이터를 제공하는 포털에 따라 API주소 형식이 다름.

■ 11주차(2022.5.13.): git 연습게임, 기말고사, Github활용법

- Github에 자료를 올리고 수정하는 방법을 배움.
- 여러 작업자들이 협업을 할 때 파일들을 관리하고 작업 상황을 확인하는데 유용함.
- Anaconda prompt로 github로 업데이트할수있음.
- branch란? 개개인이 작업하는 폴더(경로)로 branch에서 작업물을 만들고 main으로 push하면 다른 사람들과 공유할 수 있음.

코드
cd ~: 경로확인 cd desktop git clone [코드] ls: 현재위치 cd [폴더명] git pull : 최신상태로 업데이트 git status : 변경상태확인 git add . : 변경내용저장 git commit -m "upload files" -로그인필요 git branch: branch 생성 git checkout A: A 브랜치로 전환 git push origin main: origin 폴더를 main으로 업데이트 git checkout sub: 초점을 바꿈. +) branch 병합

git checkout -t origin/minji : 원하는 브랜치로 이동
 cd submit : 제출해야하는 폴더로 경로 지정
 git commit -m "final test commit": 커밋하기
 git checkout main: 다시 메인으로 이동(메인으로 이동한 후에 merge해줘야 됨)
 git merge minji: 'minji' 폴더 병합
 git push: 메인으로 push

```
user@pc2-02 MINGW64 ~/Desktop/2022jejuUniv/submit/minji (main)
$ git branch
main
user@pc2-02 MINGW64 ~/Desktop/2022jejuUniv/submit/minji (main)
$ git branch sub
user@pc2-02 MINGW64 ~/Desktop/2022jejuUniv/submit/minji (main)
$ git branch
main
sub
user@pc2-02 MINGW64 ~/Desktop/2022jejuUniv/submit/minji (main)
$ git branch minji
user@pc2-02 MINGW64 ~/Desktop/2022jejuUniv/submit/minji (main)
$ git branch
minji
sub
user@pc2-02 MINGW64 ~/Desktop/2022jejuUniv/submit/minji (main)
$ git checkout minji
Switched to branch 'minji'
user@pc2-02 MINGW64 ~/Desktop/2022jejuUniv/submit/minji (minji)
$ git status
On branch minji
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  report_park_jinsu - \353\263\265\354\202\254\353\263\270 (3) - \353\263\265\354\202\254\353\263\270.pdf
  report_park_jinsu - \353\263\265\354\202\254\353\263\270 (4) - \353\263\265\354\202\254\353\263\270.pdf
  report_park_jinsu - \353\263\265\354\202\254\353\263\270 (5) - \353\263\265\354\202\254\353\263\270.pdf
nothing added to commit but untracked files present (use "git add" to track)
```

그림 1 branch간 이동,

```
user@pc2-02 MINGW64 ~/Desktop/2022jejuUniv/submit/minji (minji)
$ git push origin minji
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 446 bytes | 111.00 KiB/s, done.
Total 5 (delta 1), reused 1 (delta 0), pack-reused 0
Remote: Resolving deltas: 100% (1/1), completed with 1 local object.
Remote: Create a pull request for 'minji' on GitHub by visiting:
Remote: https://github.com/1aftworld/2022jejuUniv/pull/new/minji
Remote:
To https://github.com/1aftworld/2022jejuUniv.git
 * [new branch] minji -> minji

user@pc2-02 MINGW64 ~/Desktop/2022jejuUniv/submit/minji (minji)
$ git status
On branch minji
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   report_park_jinsu - \353\263\265\354\202\254\353\263\270 (3) - \353\263\265\354\202\254\353\263\270.pdf
    new file:   report_park_jinsu - \353\263\265\354\202\254\353\263\270 (4) - \353\263\265\354\202\254\353\263\270.pdf
    new file:   report_park_jinsu - \353\263\265\354\202\254\353\263\270 (5) - \353\263\265\354\202\254\353\263\270.pdf
```

그림 2 main으로 push

■ 12주차(2022.5.20.): API로 데이터 받고 CSV로 저장하기

- API로 받은 데이터들을 CSV로 저장하는 것까지 코딩해봄.
- 구글 Collab을 활용하였으며, 원하는 데이터를 추출할 때 반복작업이 필요한 경우 어떠한 방법으로 해결할지에 대해 고민함.

```
import requests as rq
import json as js
import csv
import pandas as pd
def getCovid(page, df):#함수정의
    urls = url + param + str(page)
    print(urls)
    result= rq.get(urls) #rq.get() 데이터불러오기
    if result.status_code != 200: # 200:OK #상태코드 확인하는 것임
        print(result.status_code)
    return
```



```

r = js.loads(result.text) #데이터 형태 지정 및 데이터 로드
print(r)    # for debug

df2 = pd.DataFrame(r['data'])
df
df = df.append(df2)
df

url = 'https://open.jejudatahub.net/api/proxy/881baattD8ba1a6aaaDaa8111
88b16t8/1tj1c1111j_02r1j_ecro12b96_1b1t_?'
param='limit=100&number=3'

urls = url + param + '1'
#여기서는 31페이지를 가져오게 됨, 위에 number=3으로 지정해서.
print(urls)
result= rq.get(urls)
print(result.status_code) # 200:OK
r = js.loads(result.text)

print(r['hasMore']) # 추가 데이터 여부
print(r['totCnt']) # 전체 행 수
df = pd.DataFrame(r['data'])
#가져올 페이지 범위
for page in range(1,2):
    getCovid(page, df)
# 가져온 데이터 보기
df.describe()

```

2. 학습내용 활용방법

1) 개인적인 성과

(1) 인공지능에 대한 개념 이해

- 인공지능이라는 분야가 와닿지 않고 생소했는데, 직접 프로그램 코딩도 해보면서 실체가 없게 느껴졌던 인공지능 분야가 낯설지 않게 됨.
- 최근에 인공지능을 기반으로 한 많은 흥미로운 활용사례와 프로그램들을 알게되어 흥미로웠음.

(2) 효율적인 업무 수행의 필요성

- 코딩을 하는 과정을 통해서 효율적인 업무수행에 대한 방법을 고민하게 되었음.
- 어떠한 프로젝트든, 연구이던 일련의 흐름을 잘 기록하고 정리하는 과정의 반복인 만큼

각종 업무효율을 위한 소프트웨어나 github같은 시스템을 잘 활용해볼 필요가 있다고 느낌.

2) 향후 활용 계획

(1) 개인적인 연구에 있어 데이터 분석, 전처리 등에 활용 할 수 있을 것.

- 파이썬의 경우 학습시간이 더 필요할 것으로 보여 바로 활용은 어려울 것 같음.
- 연구를 위한 데이터를 수집하고 처리할 때, 반복적인 데이터 전처리나 크롤링을 활용한 데이터 수집이 가능할것으로 기대함.

(2) 활용해본 여러 인공지능 툴로 업무수행

- 학기 초에 배웠던 여러 단축키 등은 매우 유용하게 사용하고 있음.
- 이미 개발되어있는 많은 인공지능 tool에 관심을 갖고 효율적인 업무수행에 도움이 되는 부분들은 적극 활용해 볼 계획임.