| 교육 제목 | 데이터 기반 인공지능 시스템 엔지니어 양성 과정\_판다스 |
| --- | --- |
| 교육 일시 | 2021년 9월 30일 |
| 교육 장소 | YGL C-6 학과장 & 자택(디스코드 이용한 온라인) |
| **교육 내용** | |
| 오전 | 파이썬 머신러닝 판다스 데이터 분석 3장 \_ 데이터 살펴보기   1. 데이터프레임의 구조    * 데이터 내용 미리보기 - df.head() 메소드 : 데이터셋의 내용과 구조를 개략적으로 확인 <-> tail()    * 데이터의 요약 정보 확인하기 - df.shape : 행과 열의 개수를 튜플로 반환 - df.info() : 데이터프레임의 기본 정보를 출력 - df.describe() : 산술(숫자) data를 가지는 열의 주요 통계 정보 출력   c.f) 옵션으로 include=’all’ 추가 시 문자열 데이터의 고유값 개수, 최빈값, 빈도수  | 판다스 자료형 : int64, float64, object(파이썬의 str), datetime64/timedalta64 (시간 data)  !. 특정 열의 자료형 확인 : df.열이름.dtypes | | --- |  * + 데이터 개수 확인  - df.count() : 데이터프레임 각 열이 가지고 있는 데이터 개수를 시리즈 객체로 반환   단!!. 유효한 값의 개수를 반환  - df.value\_counts() : 시리즈 객체의 고유값 개수를 세는 데 사용  c.f) dropna=True 옵션을 설정하면 NaN을 제외하고 계산  1. 통계 함수 적용    * 평균값 - df.mean() / df.열이름.mean() : 열의 평균값을 계산하여 시리즈 객체로 반환 - df.median() / df.열이름.median() : 열의 중간값을 계산하여 시리즈 객체로 반환 - df.max() / df.열이름.max() : 열의 최대값을 계산하여 시리즈 객체로 반환 - df.min() / df.열이름.min() : 열의 최소값을 계산하여 시리즈 객체로 반환 - df.std() / df.열이름.std() : 열의 표준편차를 계산하여 시리즈 객체로 반환 - df.corr() / df.열이름.corr() : 열의 상관계수를 계산하여 시리즈 객체로 반환 2. 판다스 내장 그래프 도구 활용 : plot() 메소드 적용  | kind 옵션 : line-선 그래프, bar-수직 막대 그래프, barh-수평 막대 그래프, his-히스토그램, box-박스 플룻, kde-커널 밀도 그래프, area-면적 그래프, pie-파이 그래프, scatter-산점도 그래프, hexbin-고밀도 산점도 그래프 | | --- |  * + 선 그래프 : df.plot()   + 막대 그래프 : df.plot(kind=’bar’)   + kind 옵션 변경으로 상기 박스 모든 그래프 사용 가능   ... |
| 오후 | 파이썬 머신러닝 판다스 데이터 분석 4장 \_ 시각화 도구   1. Matplotlib - 기본 그래프 도구  | 모듈 불러 오기 : import matplotlib.pyplot as plt  NaN 채우기 : fillna()의 옵션 설정 method=’ffill’  차트제목 : plt.title(‘ 제목 ‘)  축 이름 : plt.xlabel(‘ x 이름') plt.ylabel(‘ y 이름')  그래프 꾸미기  스타일 서식 지정 : plt.style.use(‘ggplot’)  그림 사이즈 지정 : plt.figure(figsize = (길이, 높이))  x 축 라벨 회전 : plt.xticks(rotation=회전각)  범례표시 : plt.legend(labels=[‘범례이름', loc=’best’] | | --- | | 화면 분할하여 그래프 여러개 그리기 - axe 객체 활용  하나의 axe 객체에 그래프 여러개 표시도 가능  그림 사이즈 지정 후 fig 객체에 add\_subplot() 메소드를 적용  fig = plt.figure(figsize = (길이, 높이)  ax1 = fig.add\_subplot(분할행수,분할열수, 1)  ax2 = fig.add\_subplot(분할행수,분할열수, 2)  ...  axn = fig.add\_subplot(분할행수,분할열수, n) | | 선 그래프 꾸미기 옵션 : plt.plot( x 축, y 축, 꾸미기 옵션)  마커 지정 : market=’o’ / 마커 크기 : markersize=10 / 마커 색상 : markerfacecolor=’green’ / 선 색 : color = ‘olive’ / 선의 두께 : linewidth = 2 / 라벨 지정 : label = ‘지정하고 싶은대로' |  * + 선 그래프 : plt.plot(df)   + 먼젹 그래프 : df.plot(kind=’area’,...) : stacked=, alpha= 옵션 설정 가능   + 막대 그래프 : df.plot(kind=’bar’,...) : width=, color= 옵션 설정 가능   + 히스토그램 : df.plot(kind=’hist’,...) bins=, 옵션 설정 가능   + 산점도 : df.plot(kind=’scatter’, x=’x축 변수', y=’y축 변수', …) c=(색상), s=(크기) 옵션   + 파이 차트 : df.plot(kind=’pie’, … ) startangle=(시작점), color=[ , ,] 옵션   + 박스 플롯 : df.boxplot(x=[ , , ], …) label= ,vert= 옵션  1. Seaborn 라이브러리 - 고급 그래프 도구  | import seaborn as sns | | --- |  * + 회귀선이 있는 산점도 : sns.regplot(x=x축변수, y=y축변수, data=df)   + 히스토그램/커널 밀도 그래프 : sns.distplot([,,,]) / sns.kdeplot(x=’열이름', data=df)   + 히트맵, 범주형 데이터의 산점도, 막대 그래프, 빈도 그래프, 박스 플롯/바이올린 그래프, 조인트 그래프 등등이 있다.  1. Folium 라이브러리 - 지도 활용    * 지도를 시각화하기 유용한 도구    * 설치 : 아나콘다 배포판 사용 시에도 설치 필요    * 지도 만들기 : Map() 함수 이용 → 지도 화면은 줌과 이동 기능 가능, 웹 환경에서만 지도 확인 가능, 마커 표시, 마커 클릭 시 팝업창 표시 가능  ex) folium.Map(location[위도,경도], zoom\_start=12) |