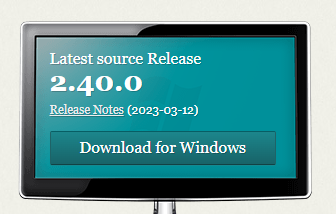
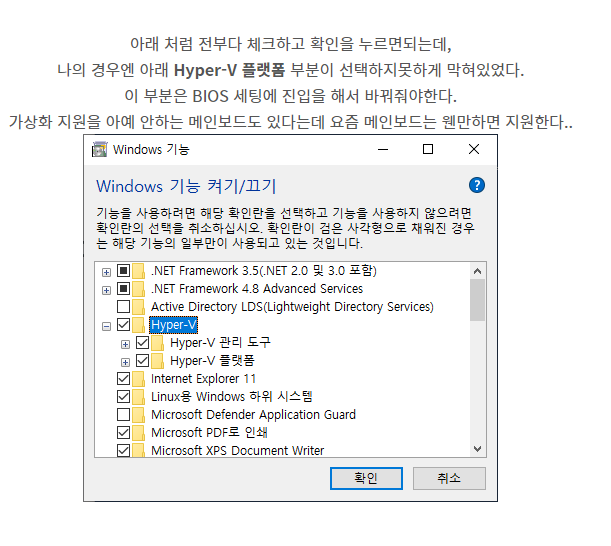
* **Try**
* 머신러닝 라이브러리
  + Numpy
    - import numpy as np *# numpy 패키지 로드하여 np로 사용*
    - a = [[1,2,3], [4,5,6]] *# 리스트에서 행렬생성*
    - b = np.array(a)
    - print(b)
  + Pandas
    - import pandas as pd
    - *# 딕셔너리*
    - data = {
    - 'year':[2016, 2017, 2018],
    - 'GDP rate': [2.8, 3.1, 3.0],
    - 'GDP': ['1.637M', '1.73M', '1.83M' ]
    - }
    - df = pd.DataFrame(data, index=data['year'])
    - *# index추가할 수 있음*
    - print(df)
  + Matplotlib → pyplot
    - **import** matplotlib.pyplot **as** plt
    - plt.**plot**([1, 2, 3, 4])
    - plt.**show**()
    - plt.plot([1, 2, 3, 4], [1, 4, 9, 16])
    - # plt.**axis**([0, 6, 0, 20]) # 스타일
    - plt.show()
    - **import** matplotlib.pyplot **as** plt
    - **import** numpy **as** np
    - # 200ms 간격으로 균일하게 샘플된 시간
    - t = np.arange(0., 5., 0.2)
    - # 빨간 대쉬, 파란 사각형, 녹색 삼각형
    - plt.plot(t, t, 'r--', t, t\*\*2, 'bs', t, t\*\*3, 'g^')
    - plt.show()
  + Tensorflow
    - ###########################
    - # 라이브러리 사용
    - **import** tensorflow as tf
    - **import** pandas as pd
    - ###########################
    - # 데이터를 준비합니다.
    - 파일경로 **=** '<https://raw.githubusercontent.com/blackdew/tensorflow1/master/csv/lemonade.csv>'
    - 레모네이드 **=** pd.read\_csv(파일경로)
    - 레모네이드.head()
    - # 종속변수, 독립변수
    - 독립 **=** 레모네이드[['온도']]
    - 종속 **=** 레모네이드[['판매량']]
    - print(독립.shape, 종속.shape)
    - ###########################
    - # 모델을 만듭니다.
    - X **=** tf.keras.layers.Input(shape**=**[1])
    - Y **=** tf.keras.layers.Dense(1)(X)
    - model **=** tf.keras.models.Model(X, Y)
    - model.compile(loss**=**'mse')
    - ###########################
    - # 모델을 학습시킵니다.
    - model.fit(독립, 종속, epochs**=**1000, verbose**=**0)
    - model.fit(독립, 종속, epochs**=**10)
    - ###########################
    - # 모델을 이용합니다.
    - print(model.predict(독립))
    - print(model.predict([[15]]))
* **Done**
* STEP 1 : 추천 영화를 제공해 주는 방법
  + API-1 : 영화 제목 검색
  + API-2 : 제작년도 검색
  + API-3 : 랜덤 검색
* STEP 2 : 영화 상세 정보 제공
  + API-1-1 : 영화 상세 정보 제공
* **To Do**
* **영화 제목 검색 API 개발 ( ~ )**
* **영화 제작년도 검색 API 개발 ( ~ )**
* 개발 환경
  + 공유 저장 공간 : 소스
    - github ← 계정 레파지토리 생성
    - invite : [stuko92@gmail.com](mailto:stuko92@gmail.com)
    - <https://velog.io/@summer_luna_0/GitHub%EC%B4%88%EB%8C%80-collaboration%ED%95%98%EB%8A%94%EB%B2%95-%EC%B4%88%EB%8C%80%EB%B0%9B%EC%95%98%EC%9D%84%EB%95%8C>
    - git 설치
      * <https://git-scm.com/>
      * 
      * git 사용법 숙지
    - 폴더 구조
      * front
        + Android vs React
      * backend
        + python : API 서버
  + 파이썬 설치
    - 파이썬 아나콘다 설치 : 3.8버전 설치
  + Pycharm 설치
  + Docker 설치
    - <https://www.docker.com/products/docker-desktop/>
    - 
  + pip install flask (API)
  + 코드 작성
  + Console 실행
  + Docker 실행
* **영화 상세 정보 → 음악 취향 선택 알고리즘에 대한 설계**
  + 머신러닝/딥러닝 방법을 활용 함.
* 분석 환경 (개발 환경 완료 후)
  + 머신러닝 환경 (pip install …..)
    - numpy
    - pandas
    - sklearn
  + 데이터 수집
    - API 기반의 데이터 수집 개발
  + 모형 설계 / 개발
    - 알고리즘 선택 및 모형 개발
    - X 변수들… ( 선택한 영화의 요소들 )
    - Y 변수 ( 음악 취향 )
* **Concept**
* 모듈( Module) (응집도:Cohesion , 결합도:Coupling)