to see the current environment variables:

>>> $ printenv

# In order to see a specific path name: echo $<variable name>

>>> $ echo $PATH

# setting temporary environment variable:

# export <current variable name> = <new value>:$<current variable name>

>>> export PATH=/Users/hy/Documents/apache-maven-3.5.0/bin:$PATH

# setting a permanent variable:

>>> $ nano .bash\_profile # go to the end and add the path with above “export” command and save with Ctrl+y

## 5.5 Virtual Environment

# diyelim ki bulamadım ve numpy’i yüklemekte icin “python interpreter’dan cikiyorum.

>>> quit() # “ctrl+D” de kullanilabilir.

# bunun yerine bir virtual environment oluşturarak,.

>>>$ python -m venv demo\_env # saniyeler icinde yeni environment oluşturulmuş olacak

>>>$ ls # windows icin “dir” - bu komutla neler oldugunu kontrol edebiliriz

# olusturdugumuz environment’i aktif hale getirelim

>>>$ . demo\_env/bin/activate # “.” Yerine “source” da yazilabilir.

# not: burada “demo\_env” ya da “bin” icin girip activate etmeye calisildiginda yeni oluşturduğumuzu environment’a geçmiyor. hangi folder’in icinde oluşturduysak orada iken activate edilmesi gerekiyor.

# asagidaki komutlarla python version kontrol edilebilir:

>>>$ python -V # ya da “python —version”

# virtual environment’daki tum kurulu paketleri göruntulemekmek icin.

>>>$ pip list

>>>$ pip install numpy # “numpy==1.16.0” ile version ozellestirilebilir.

>>>$ python

>>> import numpy

>>> numpy.version.version #ya da “print(numpy.\_\_version\_\_)”

# virtual environient’dan cikmak çok basit

>>>$ deactivate

# in order to create “requirements” file.

>>> $ pip freeze > requirements.txt

#asagidaki komut ile de daha once oluşturulmuş ya da değiştirilmiş bir dosya daki tum paketlerin (halihazirda yok ise) indirilmesini ya da versiyonun değiştirilmesini sağlıyor.

>>> pip install -r requirements.txt

7. Datayi tekrar train et

>>> $ python packages/regression\_model/regression\_model/train\_pipeline.py

10. 6.7 Versioning

0.1.0 —> ilk hane: major / ikinci hane: minor / ucuncu hane: macro

12. setup.py calistirildi

>>> $ python packages/regression\_model/setup.py sdist bdist\_wheel

# sdist —> source distribution

# bidet —> build distribution (ne anlama geliyor bilemedim)

# flask ile ilgili dosyalara göz atıldı

# bir variablenin setup icin: ml\_api navigate edildi ve sonrasında asagidaki komut kullanıldı:

>>> $ set FLASK\_APP=run.py

>>> $ python run.py # bu komutla da API run edildi

Asagidaki adres girildiğinde de (Crome) local’da sonucu goruyoruz:

>>> <http://127.0.0.1:5000/health>

Git

Notes from Udemy Course - Deployment ML models

>>> git clone <adress>

# go to the repo (deploying-machine-learning-models)

>>> cd deploying-machine-learning-models

# in order checkout commits use “git checkout <commit reference> 1st blue icon

>>> git checkout e000631baa42203b3c02c545a24876b828080dbb

# Open a pull request

# set the origin repo:

>>> git remote set-url origin https://github.com/huseyin01/deploying-machine-learning-models.git

# crate a branch —> git checkout -b <branch name>

>>> git checkout -b test\_branch

# make an empty commit —>

>>> git commit —allow-empty -m “opening pull request”

# push the commit

—> git push origin test\_branch

# go to the repo page and hit the “compare pull request”

# choose the base repository

# hit the “create pull request”