Springboard ML Course Capstone Project Deployment

Our face recognition model is based on Dlib and KNN. A set of face images were used to train the KNN model. The training program is face_model_train.py. The program can be executed as below:
(face1)$ python face_model_train.py
A trained model face_model_file_frg will be saved in current folder by using pickle

Deployment Process

1 Create a flask app
The load_model() loads a trained ML model
The get_prediction() receives JSON data. A face encoding 128D data is treated as a string in the JSON data. This function will extract the string and convert it to a numpy array. The 128 x 1 array is sent to model.predict() to get a prediction (name)
The program is saved as face_app.py

In [ ]:

```
#ShengpingJiang- Face recognition model as a flask application

import pickle
import numpy as np
from flask import Flask, request

#model = None
app = Flask(__name__)


def load_model():
    global model
    # model variable refers to the global variable
    with open('face_model_file_frg', 'rb') as f:
        model = pickle.load(f)


@app.route('/')
def home_endpoint():
    return 'Hello World!'


@app.route('/predict', methods=['GET','POST'])
def get_prediction():
    dist_threshold = 0.4
    name=''
    # Works only for a single sample
    if request.method == 'POST':
        data = request.get_json()  # Get data posted as a json
        #data[0] means 1st {} in the JSON data [{..},{..}]. data[0]
        #the value of key 'encoding' in data[0]
        #print(type(data[0]['encoding']))
        #print(data[0]['encoding'])
        #The value of the key 'encoding' is a string '[-0.17077433
        str1 = data[0]['encoding']
        # str1[1:-1] from '[-0.17077433  0.086519...]' to '-0.17077
        # np.fromstring changes a string '-0.17077433  0.086519...'
        # [-0.17077433  0.086519...]
        encoding = np.fromstring(str1[1:-1], dtype=float, sep=' ')
        #print("ecoding type:", type(encoding))
        #print(encoding)

        # reshape(1,-1) change [-0.17077433  0.086519...] to [[-0.1
        xt = encoding.reshape(1,-1)
        #print('xt:', xt)
        closest_distance = model.kneighbors(xt, n_neighbors=1, retu
        #print("closest_distance[0][0][0]:",closest_distance[0][0][0
        if closest_distance[0][0][0] <= dist_threshold :
    # model.predict(xt) returns a string list ['name']
    # model.predict(xt)[0] returns 'name'
            name = model.predict(xt)[0]
            print('name:', name)
        else:
            name = "Unknown"
    elif request.method == 'GET':
        print("Shengping")
```

```
57
58      return name
59
60
61  if __name__ == '__main__':
62      load_model()  # load model at the beginning once only
63      app.run(host='0.0.0.0', port=5000)
64
```

2 Test face_app.py in faceprod virtualenv

2.1 Create a virtual env faceprod and install packages
mkvirtualenv faceprod -p python3
(faceprod) pip install numpy
(faceprod) pip install flask
(faceprod) pip install pickle-mixin
(faceprod) pip install sklearn
(faceprod) pip freeze > faceprod_list.txt

2.2 Launch the flask app face_app.py
(faceprod)$ python face_app.py

2.3 Open another terminal. Send test data (dlib face encoding 128D vector) to web 0.0.0.0:5000/predict, and test the model

$ curl -X POST 0.0.0.0:5000/predict -H 'Content-Type: application/json' -d '[{"encoding": "
[-0.17077433 0.086519 0.04608656 0.02226515 -0.10071052 0.0246949 -0.09879136
-0.08271502 0.15330137 -0.1101086 0.2084657 0.0172283 -0.18812549 0.00964276
-0.06756912 0.11148367 -0.11918792 -0.07723383 -0.05200598 -0.01760992 0.0567386
0.04599836 0.03339319 0.04884979 -0.10915887 -0.33869374 -0.10735007 -0.11223182
0.08643846 -0.07478593 -0.05546422 -0.08678006 -0.11504613 0.01475477 0.01169325
0.15265159 -0.02465688 -0.06824835 0.21678171 -0.03042633 -0.19874264 -0.01212559
-0.02762683 0.26414317 0.13703299 0.0334272 0.01637992 -0.10932572 0.09580361
-0.21135658 0.11234938 0.1291863 0.0340074 0.03284376 0.09014399 -0.17272305
0.01153929 0.14709072 -0.14064969 0.02695761 0.03161349 0.01307983 -0.0100578
-0.05213601 0.20376676 0.14580815 -0.11039062 -0.15493403 0.11541102 -0.2119666
0.0013991 0.08922509 -0.11429761 -0.22043382 -0.28854343 0.04549009 0.44805536
0.20364918 -0.16662233 0.02062135 -0.00946902 -0.02268174 0.16432424 0.10247331
-0.08463222 0.0589206 -0.11151487 0.04075154 0.17744561 0.00353054 -0.0321093
0.19991624 0.01635876 0.06169297 0.05581587 0.04786064 -0.07188784 -0.04009981
-0.1177263 -0.01570286 0.08082893 -0.0241716 0.03095182 0.11278267 -0.16012146
0.1034444 -0.01475013 -0.01811141 0.03154366 0.02885633 -0.14979976 -0.0449345
0.21942021 -0.22967488 0.15503235 0.15902625 0.02446658 0.15540583 0.12920454
0.0752509 -0.01832712 -0.00534262 -0.19305748 -0.00229457 0.01291393 -0.05213701
0.07341617 0.01301993]"}]'

Note: above command is one line. No return is in the line

3 Create a Dockerfile
Use text editor to create Dockerfile and put in lines below:
FROM python:3.6-slim

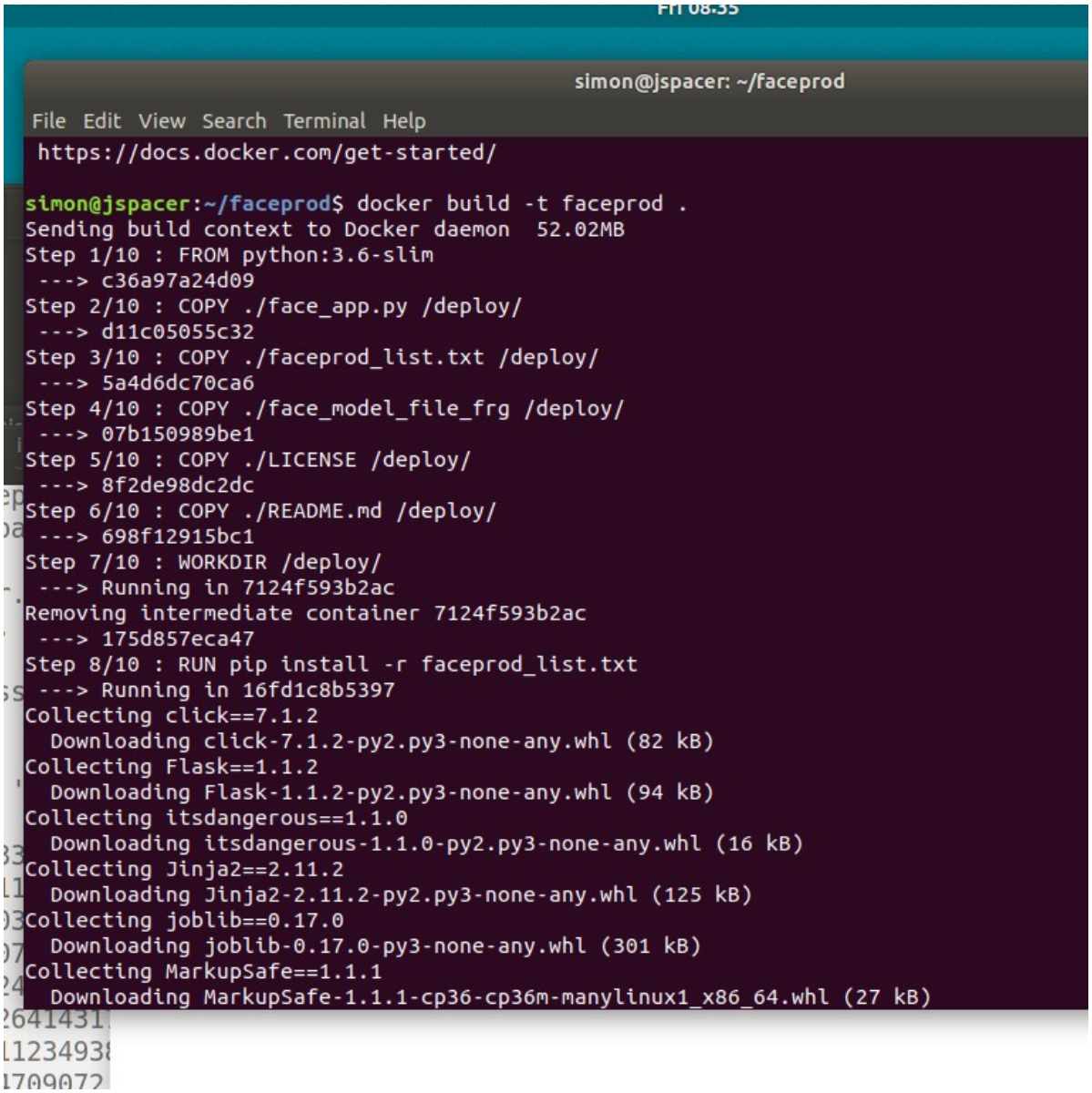
COPY ./face_app.py /deploy/
COPY ./faceprod_list.txt /deploy/
COPY ./face_model_file_frg /deploy/
COPY ./LICENSE /deploy/
COPY ./README.md /deploy/

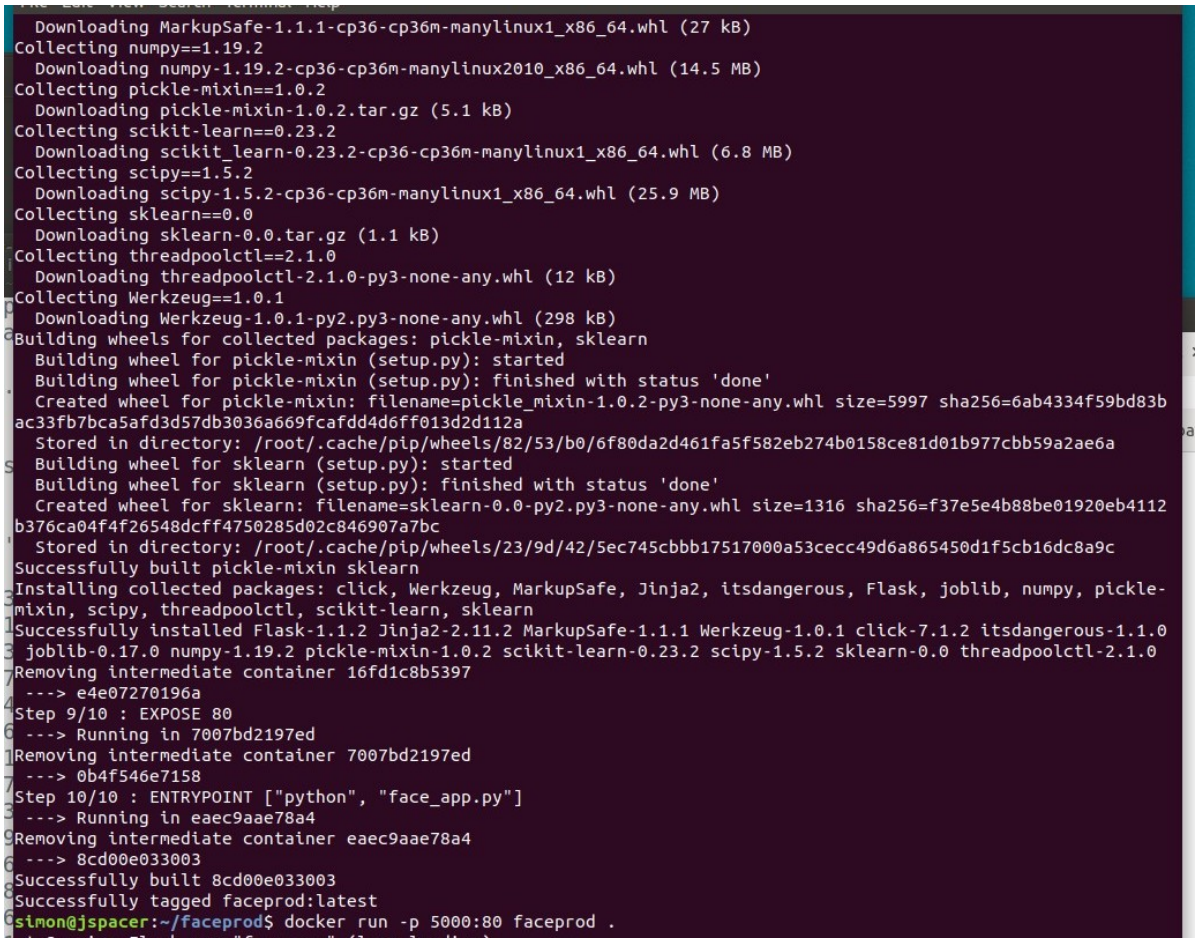
WORKDIR /deploy/
RUN pip install -r faceprod_list.txt
EXPOSE 5000
ENTRYPOINT ["python", "face_app.py"]

```
4 Create Docker image<br>
Get out the virtual env faceprod. Check docker is running<br>
$ docker run hello-world<br>
Got permission denied...<br>
$ sudo chmod 666 /var/run/docker.sock #this command fix above error<br>
$ docker run hello-world<br>
Hello from Docker!<br>

Create docker image<br>
~/faceprod$ docker build -t faceprod .

```

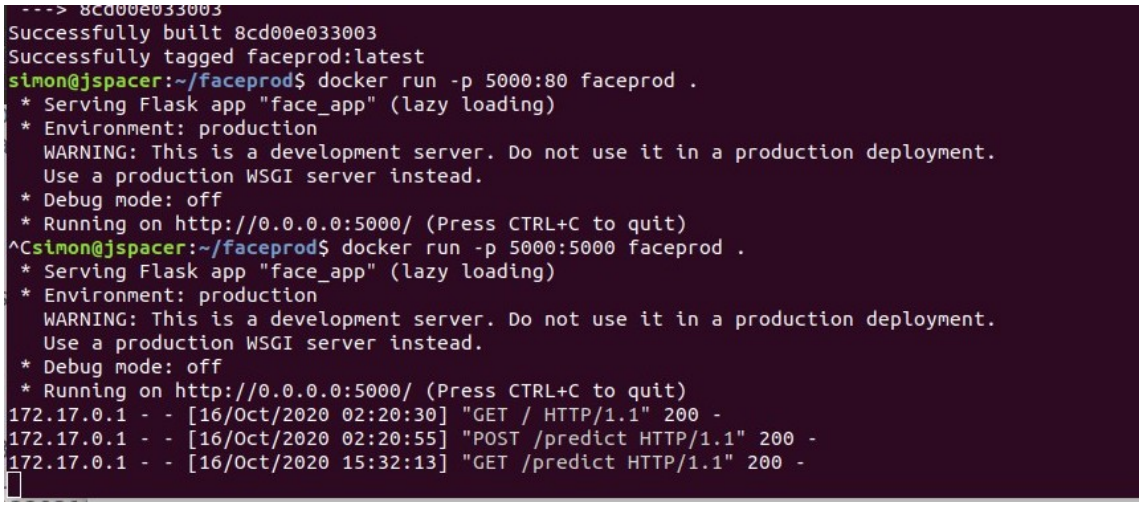

Fri 08:35
simon@jspacer: ~/faceprod
File Edit View Search Terminal Help
https://docs.docker.com/get-started/
simon@jspacer:~/faceprod$ docker build -t faceprod .
Sending build context to Docker daemon 52.02MB
Step 1/10 : FROM python:3.6-slim
---> c36a97a24d09
Step 2/10 : COPY ./face_app.py /deploy/
---> d11c05055c32
Step 3/10 : COPY ./faceprod_list.txt /deploy/
---> 5a4d6dc70ca6
Step 4/10 : COPY ./face_model_file_frg /deploy/
---> 07b150989be1
Step 5/10 : COPY ./LICENSE /deploy/
---> 8f2de98dc2dc
Step 6/10 : COPY ./README.md /deploy/
---> 698f12915bc1
Step 7/10 : WORKDIR /deploy/
---> Running in 7124f593b2ac
Removing intermediate container 7124f593b2ac
---> 175d857eca47
Step 8/10 : RUN pip install -r faceprod_list.txt
---> Running in 16fd1c8b5397
Collecting click==7.1.2
Downloading click-7.1.2-py2.py3-none-any.whl (82 kB)
Collecting Flask==1.1.2
Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
Collecting itsdangerous==1.1.0
Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2==2.11.2
Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
Collecting joblib==0.17.0
Downloading joblib-0.17.0-py3-none-any.whl (301 kB)
Collecting MarkupSafe==1.1.1
Downloading MarkupSafe-1.1.1-cp36-cp36m-manylinux1_x86_64.whl (27 kB)



5 Launch and test docker image

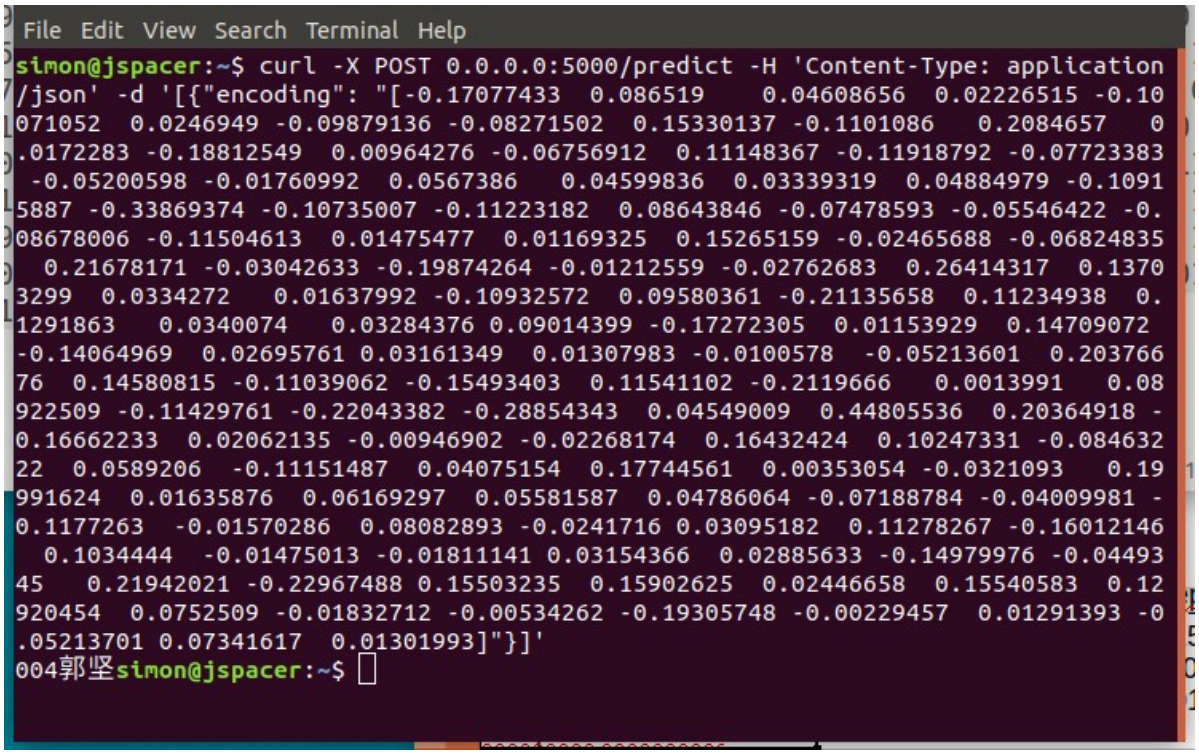
Run docker image. 1st 5000 is local machine port. 2nd 5000 is the port assigned in face_app.py (it is inside docker image)
~/faceprod$ docker run -p 5000:5000 faceprod .

- Serving Flask app "face_app" (lazy loading)
- Environment: production
  WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
- Debug mode: off
- Running on http://0.0.0.0:5000/ (http://0.0.0.0:5000/) (Press CTRL+C to quit)



In another terminal, send test data and get response
$ curl -X POST 0.0.0.0:5000/predict -H 'Content-Type: application/json' -d '[{"encoding": "
[-0.17077433 0.086519 0.04608656 0.02226515 -0.10071052 0.0246949 -0.09879136
-0.08271502 0.15330137 -0.1101086 0.2084657 0.0172283 -0.18812549 0.00964276
-0.06756912 0.11148367 -0.11918792 -0.07723383 -0.05200598 -0.01760992 0.0567386
0.04599836 0.03339319 0.04884979 -0.10915887 -0.33869374 -0.10735007 -0.11223182
0.08643846 -0.07478593 -0.05546422 -0.08678006 -0.11504613 0.01475477 0.01169325
0.15265159 -0.02465688 -0.06824835 0.21678171 -0.03042633 -0.19874264 -0.01212559
-0.02762683 0.26414317 0.13703299 0.0334272 0.01637992 -0.10932572 0.09580361
-0.21135658 0.11234938 0.1291863 0.0340074 0.03284376 0.09014399 -0.17272305
0.01153929 0.14709072 -0.14064969 0.02695761 0.03161349 0.01307983 -0.0100578
-0.05213601 0.20376676 0.14580815 -0.11039062 -0.15493403 0.11541102 -0.2119666
0.0013991 0.08922509 -0.11429761 -0.22043382 -0.28854343 0.04549009 0.44805536
0.20364918 -0.16662233 0.02062135 -0.00946902 -0.02268174 0.16432424 0.10247331
-0.08463222 0.0589206 -0.11151487 0.04075154 0.17744561 0.00353054 -0.0321093
0.19991624 0.01635876 0.06169297 0.05581587 0.04786064 -0.07188784 -0.04009981
-0.1177263 -0.01570286 0.08082893 -0.0241716 0.03095182 0.11278267 -0.16012146
0.1034444 -0.01475013 -0.01811141 0.03154366 0.02885633 -0.14979976 -0.0449345
0.21942021 -0.22967488 0.15503235 0.15902625 0.02446658 0.15540583 0.12920454
0.0752509 -0.01832712 -0.00534262 -0.19305748 -0.00229457 0.01291393 -0.05213701
0.07341617 0.01301993]"}]'

An answer from the flask app:
004郭坚
Screenshort for testing docker image and get an answer:


File Edit View Search Terminal Help
simon@jspacer:~$ curl -X POST 0.0.0.0:5000/predict -H 'Content-Type: application
/json' -d '[{"encoding": "[-0.17077433 0.086519 0.04608656 0.02226515 -0.10
071052 0.0246949 -0.09879136 -0.08271502 0.15330137 -0.1101086 0.2084657 0
.0172283 -0.18812549 0.00964276 -0.06756912 0.11148367 -0.11918792 -0.07723383
-0.05200598 -0.01760992 0.0567386 0.04599836 0.03339319 0.04884979 -0.1091
5887 -0.33869374 -0.10735007 -0.11223182 0.08643846 -0.07478593 -0.05546422 -0.
08678006 -0.11504613 0.01475477 0.01169325 0.15265159 -0.02465688 -0.06824835
0.21678171 -0.03042633 -0.19874264 -0.01212559 -0.02762683 0.26414317 0.1370
3299 0.0334272 0.01637992 -0.10932572 0.09580361 -0.21135658 0.11234938 0.
1291863 0.0340074 0.03284376 0.09014399 -0.17272305 0.01153929 0.14709072
-0.14064969 0.02695761 0.03161349 0.01307983 -0.0100578 -0.05213601 0.203766
76 0.14580815 -0.11039062 -0.15493403 0.11541102 -0.2119666 0.0013991 0.08
922509 -0.11429761 -0.22043382 -0.28854343 0.04549009 0.44805536 0.20364918 -
0.16662233 0.02062135 -0.00946902 -0.02268174 0.16432424 0.10247331 -0.084632
22 0.0589206 -0.11151487 0.04075154 0.17744561 0.00353054 -0.0321093 0.19
991624 0.01635876 0.06169297 0.05581587 0.04786064 -0.07188784 -0.04009981 -
0.1177263 -0.01570286 0.08082893 -0.0241716 0.03095182 0.11278267 -0.16012146
0.1034444 -0.01475013 -0.01811141 0.03154366 0.02885633 -0.14979976 -0.04493
45 0.21942021 -0.22967488 0.15503235 0.15902625 0.02446658 0.15540583 0.12
920454 0.0752509 -0.01832712 -0.00534262 -0.19305748 -0.00229457 0.01291393 -0
.05213701 0.07341617 0.01301993]"}]'
004郭坚simon@jspacer:~$