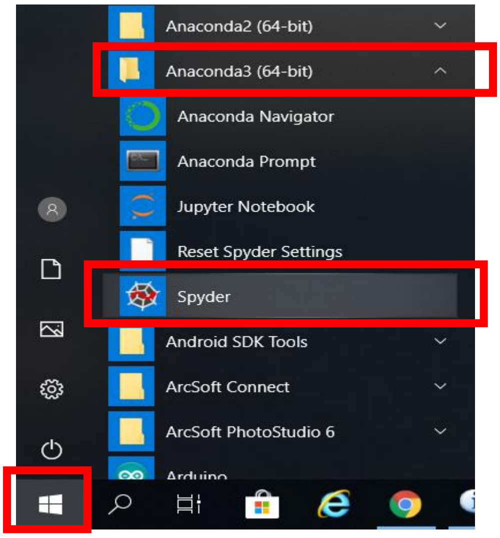
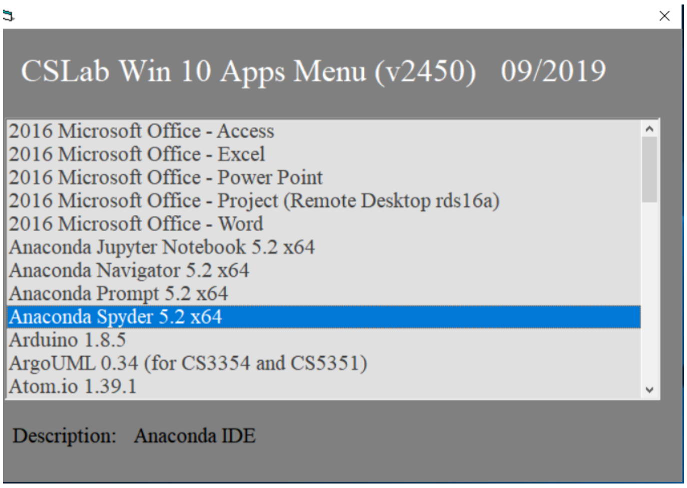
Guideline for Lab 2

Step 1. Open IDE





CSLab Win 10 Apps Menu (v2450) 09/2019

2016 Microsoft Office - Access
2016 Microsoft Office - Excel
2016 Microsoft Office - Power Point
2016 Microsoft Office - Project (Remote Desktop rds16a)
2016 Microsoft Office - Word
Anaconda Jupyter Notebook 5.2 x64
Anaconda Navigator 5.2 x64
Anaconda Prompt 5.2 x64
Anaconda Spyder 5.2 x64
Arduino 1.8.5
ArgoUML 0.34 (for CS3354 and CS5351)
Atom.io 1.39.1

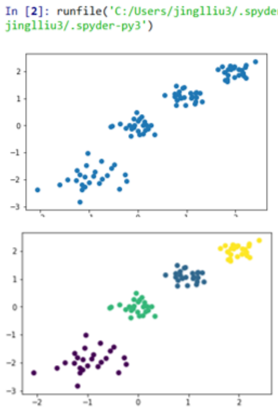
Description: Anaconda IDE

(a) Start->Anaconda3->Spyder

(b) Directly double click Anaconda Spyder5.2x64

Step 2. Example for K-means in Spyder

```
import matplotlib.pyplot as plt
from sklearn.datasets.samples_generator import make_blobs
X, y = make_blobs(n_samples=100, n_features=2,centers=[[-1,-2], [0,0], [1,1], [2,2]],cluster_std=[0.4, 0.2, 0.2, 0.2],
random_state =9)
plt.scatter(X[:, 0], X[:, 1])

from sklearn.cluster import KMeans
noclusters=4
y_pred = KMeans(n_clusters=noclusters).fit_predict(X)

plt.figure()
cValue = ['r','y','g','b','r','y','g','b','r']
cValue=cValue[0:noclusters]
plt.scatter(X[:, 0], X[:, 1], c=y_pred)
plt.show()
```

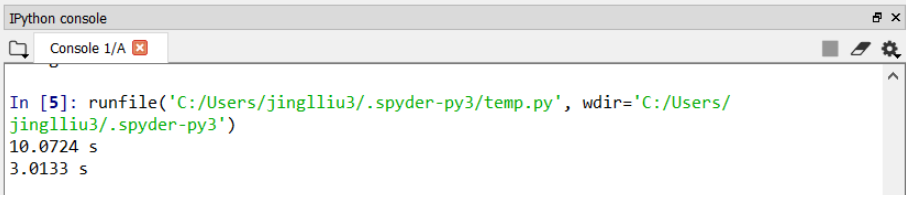


```
In [2]: runfile('C:/Users/jinglliu3/.spyder-py3/temp.py', wdir='C:/Users/
jinglliu3/.spyder-py3')
```

(c) Output of Example for K-means

Step 3. Example for Joblib in Spyder

```
from joblib import Parallel, delayed
import time, math

def my_fun(i):
    """ We define a simple function here.
    """
    time.sleep(1)
    return math.sqrt(i**2)

num = 10
start = time.time()
for i in range(num):
    my_fun(i)
```



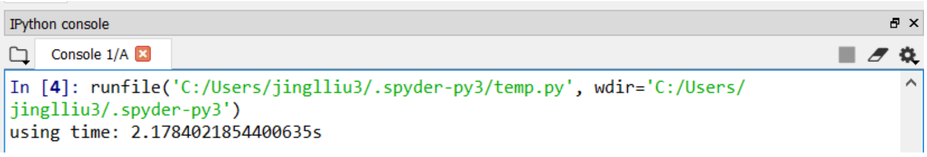
```
In [5]: runfile('C:/Users/jinglliu3/.spyder-py3/temp.py', wdir='C:/Users/
jinglliu3/.spyder-py3')
10.0724 s
3.0133 s
```

(d) Output of Example for Joblib

```
end = time.time()
print('{:.4f} s'.format(end - start))

start = time.time()
# n_jobs is the number of parallel jobs
Parallel(n_jobs=4)(delayed(my_fun)(i) for i in range(num))
end = time.time()
print('{:.4f} s'.format(end-start))
```

Step 4. Example for Multiprocessing in Spyder

```
import multiprocessing
import time

def hello(num):
  i = 0
  while i < num:
    i += 1

if __name__ == '__main__':
  ts = time.time()
  i = 3
  while i >= 1:
    p = multiprocessing.Process(target=hello,args=(20000000, )) # target=name of your function,  args=parameters of your function
    p.start() # start the process
    i-=1
  p.join()
  te = time.time()
  print("using time: "+str(te - ts)+"s")
```



```
In [4]: runfile('C:/Users/jinglliu3/.spyder-py3/temp.py', wdir='C:/Users/
jinglliu3/.spyder-py3')
using time: 2.1784021854400635s
```

(e) Output of Example for Multiprocessing

PS:

(1) You can code for Lab2 based on the above three examples.

(2) In Python, the operation of the program is usually based on some packages. Therefore, for Lab2, scikit-learn, joblib, multiprocessing and so on should be installed at first.

(3) On your own computer, I recommend using Pycharm instead of Spyder.