assignment #3-1

0540170 伏勁松

language：python 2.7

環境：ipython notebook

module：graphlab, math

1、將數據放入excel中，轉換成csv的格式，文件名為data1.csv。

	A	B	C	D	E	F	G	H
	ID	SS-IN	SED-IN	COND-IN	SS-OUT	SED-OUT	COND-OUT	STATUS
	1	168	3	1814	15	0.001	1879	ok
	2	156	3	1358	14	0.01	1425	ok
	3	176	3.5	2200	16	0.005	2140	ok
	4	256	3	2070	27	0.2	2700	ok
	5	230	5	1410	131	3.5	1575	settler
	6	116	3	1238	104	0.06	1221	settler
	7	242	7	1315	104	0.01	1434	settler
	8	242	4.5	1183	78	0.02	1374	settler
	9	174	2.5	1110	73	1.5	1256	settler
	10	1004	35	1218	81	1172	33.3	solids
	11	1228	46	1889	82.4	1932	43.1	solids
	12	964	17	2120	20	1030	1966	solids
	13	2008	32	1257	13	1038	1289	solids

2、導入csv文件

```
data_qow = graphlab.SFrame.read_csv('data1.csv')
data_qow
```

3、先將data_qw按照status分組

```
status_ok = data_qow[data_qow['STATUS'] == 'ok']
status_settler = data_qow[data_qow['STATUS'] == 'settler']
status_solids = data_qow[data_qow['STATUS'] == 'solids']
```

```
status_ok
```

ID	SS-IN	SED-IN	COND-IN	SS-OUT	SED-OUT	COND-OUT	STATUS
1	168	3.0	1814	15.0	0.001	1879.0	ok
2	156	3.0	1358	14.0	0.01	1425.0	ok
3	176	3.5	2200	16.0	0.005	2140.0	ok
4	256	3.0	2070	27.0	0.2	2700.0	ok

[? rows x 8 columns]
Note: Only the head of the SFrame is printed. This SFrame is lazily evaluated.
You can use sf.materialize() to force materialization.

```
status_settler
```

ID	SS-IN	SED-IN	COND-IN	SS-OUT	SED-OUT	COND-OUT	STATUS
5	230	5.0	1410	131.0	3.5	1575.0	settler
6	116	3.0	1238	104.0	0.06	1221.0	settler
7	242	7.0	1315	104.0	0.01	1434.0	settler
8	242	4.5	1183	78.0	0.02	1374.0	settler
9	174	2.5	1110	73.0	1.5	1256.0	settler

[? rows x 8 columns]
Note: Only the head of the SFrame is printed. This SFrame is lazily evaluated.
You can use sf.materialize() to force materialization.

```
status_solids
```

ID	SS-IN	SED-IN	COND-IN	SS-OUT	SED-OUT	COND-OUT	STATUS
10	1004	35.0	1218	81.0	1172.0	33.3	solids
11	1228	46.0	1889	82.4	1932.0	43.1	solids
12	964	17.0	2120	20.0	1030.0	1966.0	solids
13	2008	32.0	1257	13.0	1038.0	1289.0	solids

[? rows x 8 columns]

4、使用normal distribution, 分別計算各個分組的各個feature的mean和std值

```
#计算status==ok的avg和std
ok_ssin_avg = status_ok['SS-IN'].mean()
ok_ssin_std = status_ok['SS-IN'].std()
ok_sedin_avg = status_ok['SED-IN'].mean()
ok_sedin_std = status_ok['SED-IN'].std()
ok_condin_avg = status_ok['COND-IN'].mean()
ok_condin_std = status_ok['COND-IN'].std()
ok_ssout_avg = status_ok['SS-OUT'].mean()
ok_ssout_std = status_ok['SS-OUT'].std()
ok_sedout_avg = status_ok['SED-OUT'].mean()
ok_sedout_std = status_ok['SED-OUT'].std()
ok_condout_avg = status_ok['COND-OUT'].mean()
ok_condout_std = status_ok['COND-OUT'].std()
```

```
#计算status==settler的avg和std
settler_ssin_avg = status_settler['SS-IN'].mean()
settler_ssin_std = status_settler['SS-IN'].std()
settler_sedin_avg = status_settler['SED-IN'].mean()
settler_sedin_std = status_settler['SED-IN'].std()
settler_condin_avg = status_settler['COND-IN'].mean()
settler_condin_std = status_settler['COND-IN'].std()
settler_ssout_avg = status_settler['SS-OUT'].mean()
settler_ssout_std = status_settler['SS-OUT'].std()
settler_sedout_avg = status_settler['SED-OUT'].mean()
settler_sedout_std = status_settler['SED-OUT'].std()
settler_condout_avg = status_settler['COND-OUT'].mean()
settler_condout_std = status_settler['COND-OUT'].std()

#计算status==solids的avg和std
solids_ssin_avg = status_solids['SS-IN'].mean()
solids_ssin_std = status_solids['SS-IN'].std()
solids_sedin_avg = status_solids['SED-IN'].mean()
solids_sedin_std = status_solids['SED-IN'].std()
solids_condin_avg = status_solids['COND-IN'].mean()
solids_condin_std = status_solids['COND-IN'].std()
solids_ssout_avg = status_solids['SS-OUT'].mean()
solids_ssout_std = status_solids['SS-OUT'].std()
solids_sedout_avg = status_solids['SED-OUT'].mean()
solids_sedout_std = status_solids['SED-OUT'].std()
solids_condout_avg = status_solids['COND-OUT'].mean()
solids_condout_std = status_solids['COND-OUT'].std()
```

4.1、定義function計算各個feature在不同分組的normal distribution

```
def getNormalDis(x,avg,std):
    return (1 / (std * math.sqrt(2 * math.pi))) * math.exp(-math.pow(x - avg,2) / (2 * math.pow(std,2)))
```

4.2、計算給定query的probaility

```
#对SS-IN = 222, SED-IN = 4.5, COND-IN = 1,518, SS-OUT = 74, SED-OUT = 0.25, COND-OUT = 1,642计算probability
p_ok_nd = (float(4) / 13) * getNormalDis(222,ok_ssin_avg,ok_ssin_std) * getNormalDis(1518,ok_condin_avg,ok_condin_std) * getNormalDis(74,ok_ssout_avg,ok_ssout_std) * getNormal
p_settler_nd = (float(5) / 13) * getNormalDis(222,settler_ssin_avg,settler_ssin_std) * getNormalDis(1518,settler_condin_avg,settler_condin_std) * getNormalDis(74,settler_ssout
p_solids_nd = (float(4) / 13) * getNormalDis(222,solids_ssin_avg,solids_ssin_std) * getNormalDis(1518,solids_condin_avg,solids_condin_std) * getNormalDis(74,solids_ssout_avg,s
print p_ok_nd, p_settler_nd,p_solids_nd
```

```
1.2460466248e-44 8.27916285841e-14 7.67538782086e-23
```

因為probability of settler 最大，所以預測為settler

4.3 、因為probability of settler 最大，所以预测为settler

5、使用exponential

5.1、計算各個feature的rate，為1/avg

```
ok_ssin_rate = float(1) / ok_ssin_avg
ok_sedin_rate = float(1) / ok_sedin_avg
ok_condin_rate = float(1) / ok_condin_avg
ok_ssout_rate = float(1) / ok_ssout_avg
ok_sedout_rate = float(1) / ok_sedout_avg
ok_condout_rate = float(1) / ok_condout_avg

settler_ssin_rate = float(1) / settler_ssin_avg
settler_sedin_rate = float(1) / settler_sedin_avg
settler_condin_rate = float(1) / settler_condin_avg
settler_ssout_rate = float(1) / settler_ssout_avg
settler_sedout_rate = float(1) / settler_sedout_avg
settler_condout_rate = float(1) / settler_condout_avg

solids_ssin_rate = float(1) / solids_ssin_avg
solids_sedin_rate = float(1) / solids_sedin_avg
solids_condin_rate = float(1) / solids_condin_avg
solids_ssout_rate = float(1) / solids_ssout_avg
solids_sedout_rate = float(1) / solids_sedout_avg
solids_condout_rate = float(1) / solids_condout_avg
```

5.2 、定義function，計算exponential distribution p

```
## calculate the exponential distribution p
def getExpDis(x,rate):
    return rate * math.exp(-rate * x)
```

5.3、計算給定query的probability

```
p_ok_exp = (float(4) / 13) * getExpDis(222,ok_ssin_rate) * getExpDis(1518,ok_condin_rate) * getExpDis(74,ok_ssout_rate) * getExpDis(4.5,ok_sedin_rate) * getExpDis(0.25
p_settler_exp = (float(5) / 13) * getExpDis(222,settler_ssin_rate) * getExpDis(1518,settler_condin_rate) * getExpDis(74,settler_ssout_rate) * getExpDis(4.5,settler_sed
p_solids_exp = (float(4) / 13) * getExpDis(222,solids_ssin_rate) * getExpDis(1518,solids_condin_rate) * getExpDis(74,solids_ssout_rate) * getExpDis(4.5,solids_sedin_ra
print p_ok_exp,p_settler_exp,p_solids_exp
```

5.4、因為probability of settler 最大，所以预测为settler

6、使用smoothing and equal-frequency-binning k=3

6.1、对各个feature分三个bin

```
#对ssin分成三个bin
#bin1<=203 ,  203<bin2<=610,  bin3>610
p_ok_ssin_bin1 = 0.4615
p_ok_ssin_bin2 = 0.3077
p_ok_ssin_bin3 = 0.2308

p_settler_ssin_bin1 = 0.3571
p_settler_ssin_bin2 = 0.4286
p_settler_ssin_bin3 = 0.2143

p_solids_ssin_bin1 = 0.2308
p_solids_ssin_bin2 = 0.2308
p_solids_ssin_bin3 = 0.5384
```

```
#对SEDIN 分成三个bin
#bin1<=3.25 ,  3.25<bin2<=12 , bin3>12
p_ok_sedin_bin1 = 0.4615
p_ok_sedin_bin2 = 0.3077
p_ok_sedin_bin3 = 0.2308

p_settler_sedin_bin1 = 0.3571
p_settler_sedin_bin2 = 0.4286
p_settler_sedin_bin3 = 0.2143

p_solids_sedin_bin1 = 0.2308
p_solids_sedin_bin2 = 0.2308
p_solids_sedin_bin3 = 0.5384
```

```
#CONDIN
#bin1<=1286, 1286<bin2<=2758.5, bin3>2758.5
p_ok_condin_bin1 = 0.2308
p_ok_condin_bin2 = 0.3846
p_ok_condin_bin3 = 0.3846

p_settler_condin_bin1 = 0.4286
p_settler_condin_bin2 = 0.3571
p_settler_condin_bin3 = 0.2143

p_solids_condin_bin1 = 0.3846
p_solids_condin_bin2 = 0.2308
p_solids_condin_bin3 = 0.3846
```

```
#ssout bins
#bin1<=23.5, 23.5<bin2<=81.7, bin3>81.7
p_ok_ssout_bin1 = 0.4615
p_ok_ssout_bin2 = 0.3077
p_ok_ssout_bin3 = 0.2308

p_settler_ssout_bin1 = 0.2143
p_settler_ssout_bin2 = 0.3571
p_settler_ssout_bin3 = 0.4286

p_solids_ssout_bin1 = 0.3846
p_solids_ssout_bin2 = 0.3077
p_solids_ssout_bin3 = 0.3077
```

```
#sedout bins
#bin1<=0.04, 0.04<bin2<=516.75, bin3>516.75
p_ok_sedout_bin1 = 0.4615
p_ok_sedout_bin2 = 0.3077
p_ok_sedout_bin3 = 0.2308

p_settler_sedout_bin1 = 0.3571
p_settler_sedout_bin2 = 0.4286
p_settler_sedout_bin3 = 0.2413

p_solids_sedout_bin1 = 0.2308
p_solids_sedout_bin2 = 0.2308
p_solids_sedout_bin3 = 0.5384
```

```
#condout bins
#bin1<=1331.5, 1331.5<bin2<=1727, bin3>1727
p_ok_condout_bin1 = 0.2308
p_ok_condout_bin2 = 0.3077
p_ok_condout_bin3 = 0.4615

p_settler_condout_bin1 = 0.3571
p_settler_condout_bin2 = 0.4286
p_settler_condout_bin3 = 0.2413

p_solids_condout_bin1 = 0.4615
p_solids_condout_bin2 = 0.2308
p_solids_condout_bin3 = 0.3077
```

6.2、对給定query計算probability

```
#对SS-IN = 222, SED-IN = 4.5, COND-IN = 1,518, SS-OUT = 74, SED-OUT = 0.25, COND-OUT = 1,642计算probability
#SSIN bin2, SEDIN bin2, CONDIN bin2, SSOUT bin2, SEDOUT bin2 , CONDOUT bin2
p_ok_binning = (float(4) / 13) * p_ok_ssin_bin2 * p_ok_sedin_bin2 * p_ok_condin_bin2 * p_ok_ssout_bin2 * p_ok_sedout_bin2 * p_ok_condout_bin2
p_settler_binning = (float(5) / 13) * p_settler_ssin_bin2 * p_settler_sedin_bin2 * p_settler_condin_bin2 * p_settler_ssout_bin2 * p_settler_sedout_bin2 * p_settler_condout_bin2
p_solids_binning = (float(4) / 13) * p_solids_ssin_bin2 * p_solids_sedin_bin2 * p_solids_condin_bin2 * p_solids_ssout_bin2 * p_solids_sedout_bin2 * p_solids_condout_bin2
print p_ok_binning,p_settler_binning,p_solids_binning
```

6.3、settler的probability最大，所以選擇settler