This page is for final practice for DSE230/CSE255 Spring 2016

Warm-up Questions:

1.) Print the number of elements in the RDD B. Also, print the first five elements of RDD B

```
n=10000;
B=sc.parallelize(range(n))
## Your answer here
print 'Number of elements= %d'%(B.count())
print 'First 5 elements = %d'%(B.take(5))
```

2.) Given an RDD of words, find and output one of the longest words

```
words=['this','is','the','best','mac','ever','jupyter']
wordRDD=sc.parallelize(words)
## Your answer here
wordRDD.reduce(lambda w,v: w if len(w)>len(v) else v)
```

3.) Remove duplicate elements in RDD

```
DuplicateRDD = sc.parallelize([1,1,2,2,3,3])
## Your answer here
print DuplicateRDD.distinct().collect()
```

4.) Given an RDD, create a new RDD where each element appears twice

```
A=sc.parallelize(range(5))
### Your answer here
A.flatMap(lambda a: [a,a]).collect()
```

5.) Count how many positive numbers are there in the RDD?

```
B=sc.parallelize(range(-10,20))
## Your answer here
B.filter(lambda n: n > 0).count()
```

Short answer questions

Q-1 What is lazy evaluation and why is it more efficient?

The transformations are only computed when an action requires a result to be returned to the driver program. This design enables Spark to run more efficiently – for example, we can realize that a dataset created through map will be used in a reduce and return only the result of the reduce to the driver, rather than the larger mapped dataset.

Q-2 Consider the following methods of estimating pi. Which one is more efficient and why?(sc refers to SparkContext object)

Method 1

```
from random import random
def sampleAndSum(p):
    points = [(random(),random()) for i in xrange(p)]
    return sum([1 for (a,b) in points if a*a + b*b < 1])

def calculate_pi(sc, NUM_SAMPLES):
    tasks = sc.defaultParallelism
    count = sc.parallelize([NUM_SAMPLES/tasks]*tasks) \
            .map(sampleAndSum) \
            .reduce(lambda a, b: a + b)
    return count


NUM_SAMPLES=10000000
count = calculate_pi(sc, n)
print "Pi is roughly %f" % (4.0 * count / NUM_SAMPLES)
```

Method 2

```
def sample(p):
    x, y = random(), random()
    return 1 if x*x + y*y < 1 else 0

count = sc.parallelize(xrange(0, NUM_SAMPLES)).map(sample) \
            .reduce(lambda a, b: a + b)
print "Pi is roughly %f" % (4.0 * count / NUM_SAMPLES)
```

Method 1 is more efficient because it is generating random numbers in parallel at worker nodes while method 2 is generating NUM_SAMPLES numbers at driver and then distributing them to workers.

Pair RDD Questions

6.) Compute and print the largest value for each key in this pair RDD

```
PairRDD = sc.parallelize([(1,2), (2,4), (2,6)])
## Your answer here
print PairRDD.reduceByKey(lambda a,b: max(a,b)).collect()
```

7.) Sort a pair RDD by key and print the result

```
PairRDD = sc.parallelize([(2,2),(1,4),(3,6),(2,1)])
## Your answer here
print PairRDD.rdd.sortByKey().collect()
```

8.) Perform the following transformation:

```
PairRDD = sc.parallelize([(1, 2), (2, 4), (2, 6)])
# After transformation :  [(2, [4, 6]), (1, [2])]
```

```
### Your answer here
print  PairRDD.groupByKey().mapValues(lambda x:[a for a in x]).collect()
```

9.) Given two pair RDDs A and B, create the following RDD

```
[('adam', ('kalai', None)),
 ('vaclav', (None, 'M')),
 ('john', ('dow', 'M')),
 ('beth', ('simon', 'F'))]
```

```
A=sc.parallelize([('john','dow'),('adam','kalai'),('beth','simon')])
B=sc.parallelize([('beth','F'),('john','M'),('vaclav','M')])
## Your answer here
A.fullOuterJoin(B).collect()
```

Spark for Statistics Questions

10.) Suppose X is an RDD where each element is a floating point value. Write code to **efficiently** compute a good **approximation** of the median value?

```
from numpy.random import rand
X=sc.parallelize(rand(10000000)/2)
## Your answer here
L=X.sample(False,0.001).collect()
L=sorted(L)
L[len(L)/2]
```

11.) For the same RDD in Q-10, compute the mean and the standard deviation in one pass.

```
### Your answer here
from numpy import sqrt
(N,S,S2)=X.map(lambda x: (1,x,x*x)).reduce(lambda a,b:(a[0]+b[0],a[1]+b[1],a[2]+b[2]))
E=S/N
Var=S2/N-E**2
print 'mean=%f, std=%f'%(E,sqrt(Var))
```

12.) Suppose R is an RDD of tuples, each tuple containing two floating point numbers **(x,y)**. Compute the covariance of **x** and **y** using a single pass over the RDD.

```
n=10000
a=rand(n); b=rand(n)
R=sc.parallelize(zip(5*a+b,5*a-b))
## Your answer here
(N,X,Y,XY)=R.map(lambda x:np.array([1,x[0],x[1],x[0]*x[1]])).reduce(lambda a,b:a+b)
print 'cov(x,y)=',XY/N-(X/N)*(Y/N)
```

13.) Suppose R is an RDD that contains integer numbers in the range 0 to 3. Write code to efficiently compute and plot an approximate histogram.

```
X=([0]*10000+[1]*23000+[2]*15532+[3]*10000)
keys=rand(len(X))
R=sc.parallelize(zip(keys,X)).cache()
R=R.repartitionAndSortWithinPartitions(2).map(lambda x:x[1])
## Your answer here
H=R.sample(False,0.01).collect()
hist(H);
```

ML / Math questions

12) Suppose xR4 and that F(x)=(x1-3)2+(x2-2)2+(x3+1)2+(x4-3)2

What is the gradient of F(x) at the point x=(0,0,0,0) ?

Ans: F'(x)=(-6,-4,2,-6)

13) Suppose $\vec{x}$ is a random vector in R^{100} and suppose that the PCA analysis yields the mean vector $\vec{\mu}$,the eigenvectors (of length one) $\vec{v}_1,\ldots,\vec{v}_{100}$ and the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{100}$

a. Write an expression the vector $\vec{x}$ shifted so that the mean of the new RV is zero. Call the shifted vector $\vec{z}$

$\vec{z} = \vec{x} - E(\vec{x})$

b. Suppose $\vec{z}$ is expressed in terms of the eigenvectors of the PCA. Write an expression for the coefficient in front of the eigenvector $\vec{v}_i$ call this coefficient $c_i = \vec{z} \cdot \vec{v}_i$

c. What is the variance of ci ? λ_i

d. What is the covariance of c_5 and c_{10} ? 0

e. Write the expression for the approximation of x using the first two eigenvectors (you can use z and c1,c2,c3, in the expression) . Call this approximation $\mu_2 = (z.v_1)v_1 + (z.v_2)v_2$

f. What is the expected residual error? $E[||x - \mu_2||^2] = \Sigma^{100}{}_{i=3}\lambda_i$

14) Suppose that the domain X consists of 4 points: {1,2,3,4} whose probabilities are P(1)=P(2)=P(4)=⅕, and P(3) = ⅖

The conditional probabilities are

	Best prediction	Prediction error	P(X=x)
$P(Y =+1\|X = 1) = 0.9$	1	0.1	1/5
$P(Y =+1\|X = 2) = 0.2$	0	0.2	1/5
$P(Y =+1\|X = 3) = 1.0$	1	0	2/5
$P(Y =+1\|X = 4) = 0.8$	1	0.2	1/5

a. What is the Bayes error?
0.1*⅕ + 0.2*⅕ + 0*⅖+0.2*⅕ = 0.5/5=0.1

b. Consider rules of the form $f_\theta(x) =+1 \text{ } if \text{ } x \geq \theta, \quad -1 \text{ } otherwise$
i)What is the maximal absolute value of the correlation achievable by a rule with this form?
ii)What is the values of θ that achieves that highest correlation?

NOTE: In this context (boosting a weak learner) I define the correlation of a rule $h(x)$ with the labels to be:

$$Corr(h) \; = \; E(h(X)Y) = \sum_x P(x)h(x)E(Y|X=x)$$

To compute the contributions due to each $X = x$ we use the equation
$E(Y|X=x) \; = \; P(Y=+1|X=x) - P(Y=-1|X=x) = 2P(Y=+1|X=x) - 1$

And then multiply them by $h(x)$

	x=1	x=2	x=3	x=4
$P(X=x)$	⅕	⅕	⅖	⅕
Contribution if h(x)=1	0.8	-0.6	1.0	0.6
Contribution if h(x)=-1	-0.8	0.6	-1.0	-0.6
Contribution if h(x)=0	0	0	0	0

Using these values we can compute the correlation for each threshold:

	$\theta = 0.5$	$\theta = 1.5$
correlation	⅕*(0.8-0.6+2*1.0+0.6)=2.8/5 = 0.56	⅕*(-0.8-0.6+2*1.0+0.6)=1.2/5 = 0.24

$\theta = 2.5$	$\theta = 3.5$	$\theta = 4.5$
⅕*(-0.8+0.6+2*1.0+0.6)=2.4/5 = 0.48	⅕*(-0.8+0.6-2*1.0+0.6)=-1.6/5 = -0.32	⅕*(-0.8+0.6-2*1.0-0.6)=-2.8/5 = -0.56

The thresholds 0.5 and 4.5 have a correlation of 0.56 and -0.56 respectively. (This makes sense because the first always predicts +1, and the second always predicts -1)

c. Consider rules of the form $f_\theta(x) = +1 \text{ } if \; x \geq \theta, \quad 0 \; otherwise$
   i)What is the maximal correlation achievable by a rule with this form?
   ii)What is the value of θ that achieves this minimum?

We first compute the contribution of each value of X to the correlation

	x=1	x=2	x=3	x=4
Contribution if h(x)=1	0.8	-0.6	1.0	0.6
Contribution if h(x)=0	0	0	0	0

Using these values we can compute the correlation for each threshold:

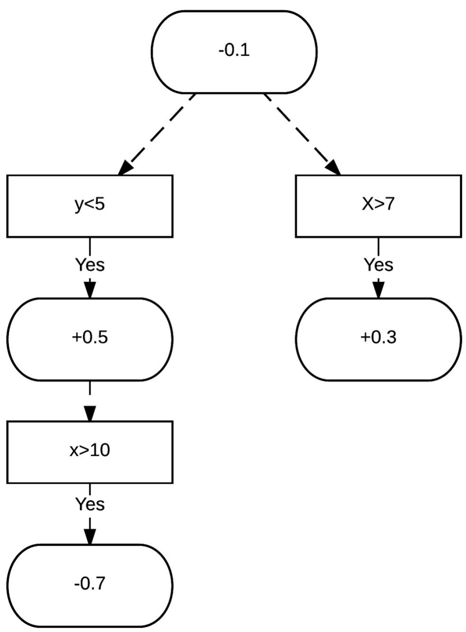
	$\theta = 0.5$	$\theta = 1.5$
correlation	⅕*(0.8-0.6+2*1.0+0.6) = .56	⅕*(-0.6+2*1.0+0.6) = 0.4

$\theta = 2.5$	$\theta = 3.5$	$\theta = 4.5$
⅕*(2*1.0+0.6) = 0.52	⅕*(0.6)= 0.12	0

The correlation is maximum i.e. 0.56 when $\theta = 0.5$

Correlation is minimum when $\theta = 4.5$

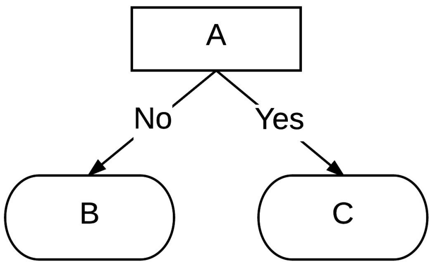
15) Consider the following AD-tree



What would be value of the score function for x=8,y=3 ?

Score = -0.1 + 0.3 + 0.5 = 0.7

16) Consider a decision stump:



Suppose that the label is binary $y \in \{-1,+1\}$ and that
$P(y=+1|B) = 0.7, \quad P(y=+1|A) = 0.8, \quad P(y=+1|C) = 0.9$

Consider a tree learning algorithm that is comparing the performance of the root A with that of the leaves B,C.

Mark all the true statements:

1. The conditional entropy of the partition B,C is lower than that of the root A.
2. The training error of the leaves B,C is lower than that of the root A.

1 is correct because the conditional entropy always decreases if the conditional probabilities of the children are different from that of the root.

2 is incorrect. Whether using the root or the two children, the prediction is always +1, and so the error rate is the same.