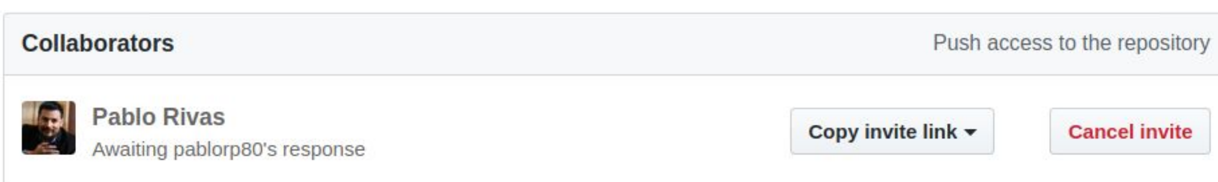
2/8/2018

Homework 0
MSIS 689 | DATA 440

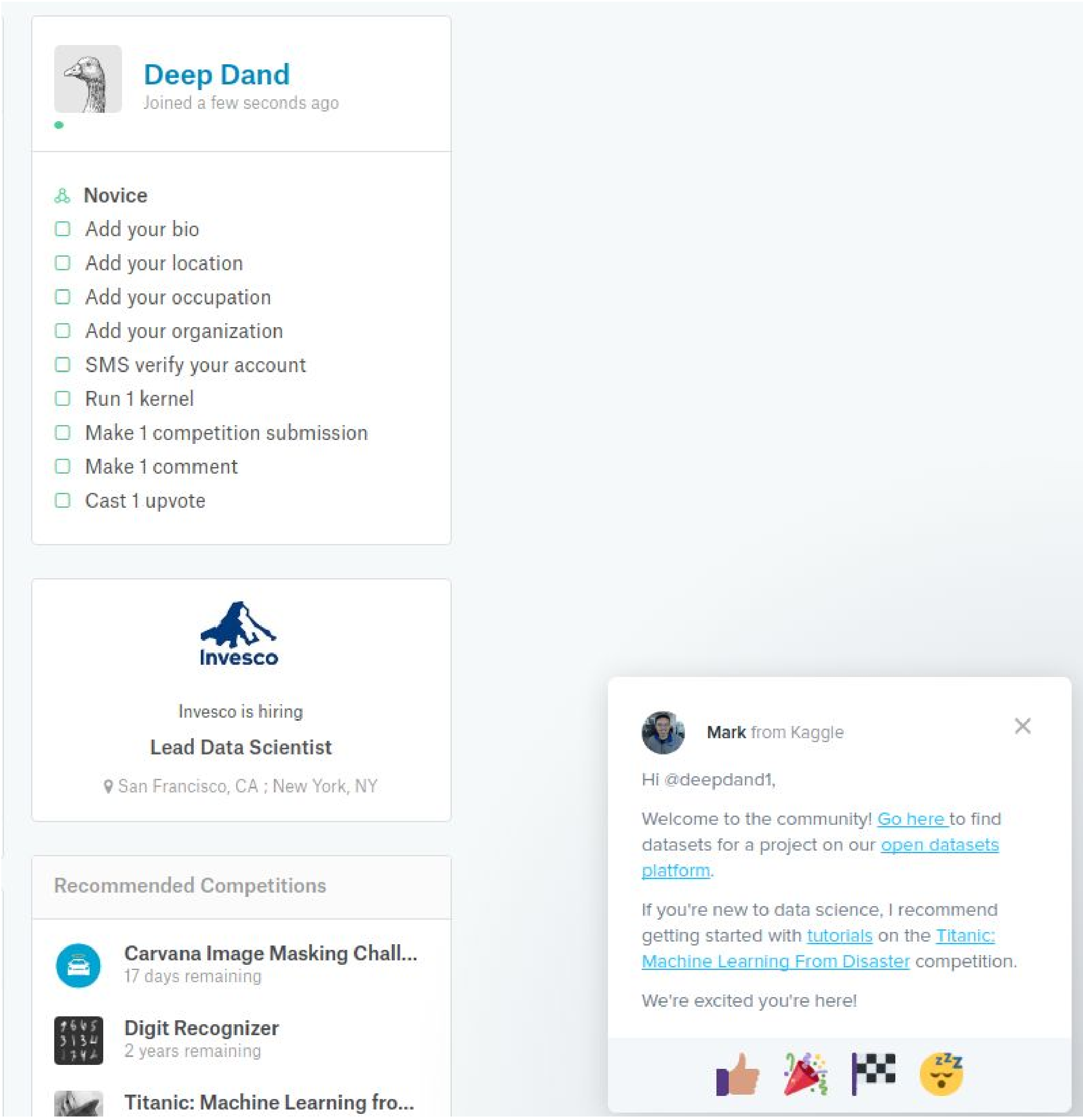
1. Packages
   a.

```
deep@acergpu940:~$ python
Python 2.7.12 (default, Nov 20 2017, 18:23:56)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys
>>> import numpy
>>> import scipy
>>> import sklearn
>>> import matplotlib
>>> import pandas
>>> print(sys.version)
2.7.12 (default, Nov 20 2017, 18:23:56)
[GCC 5.4.0 20160609]
>>> print (numpy.__version__)
1.13.3
>>> print (scipy.__version__)
0.17.0
>>> print (sklearn.__version__)
0.18.1
>>> print (matplotlib.__version__)
2.0.0
>>> print (pandas.__version__)
0.19.2
```

2. Github class repo - TheMachineLearningS18.



3. Kaggle account - deepdand1



4. Problems
   a. Taking derivative -
      $g(x) = -6x + 24$
      Equating the $g(x)$ to 0
      $-6x+24 = 0$
      Therefore, $x = 4$

For x = 4, the value of g(x) maximizes.

b. $f(x) = 3x_0^3 - 2x_0x_1^2 + 4x_1 + 8$

$fx_0^1(x) = 9x_0^2 - 2x_1^2$

$fx_1^1(x) = -4x_0x_1 + 4$

c. Matrix multiplication

i. We can not multiply A and B here since the dimensions of both matrices doesn't match the condition of matrix multiplication which is - In order to multiply two matrices, A and B, the number of columns in A must equal the number of rows in B

ii. $A^T.B$ =

```
deep@acergpu940:~$ python
Python 2.7.12 (default, Nov 20 2017, 18:23:56)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> a = np.array([[1, 4, -3],[2, -1, 3]])
>>> b = np.array([[-2, 0, 5],[0, -1, 4]])
>>> np.dot(a.T,b)
array([[-2, -2, 13],
       [-8,  1, 16],
       [ 6, -3, -3]])
```

iii. $A.B^T + C^{-1} =$

```
deep@acergpu940:~$ python
Python 2.7.12 (default, Nov 20 2017, 18:23:56)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> a = np.array([[1, 4, -3],[2, -1, 3]])
>>> b = np.array([[-2, 0, 5],[0, -1, 4]])
>>> np.dot(a.T,b)
array([[-2, -2, 13],
       [-8,  1, 16],
       [ 6, -3, -3]])
>>> c = np.array([[1, 0],[0, 2]])
>>> np.dot(a,b.T) + cinv
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'cinv' is not defined
>>> np.dot(a,b.T) + inv(c)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'inv' is not defined
>>> from numpy.linalg import inv
>>> np.dot(a,b.T) + inv(c)
array([[-16. , -16. ],
       [ 11. ,  13.5]])
```

d. **Mathematical definitions of -**

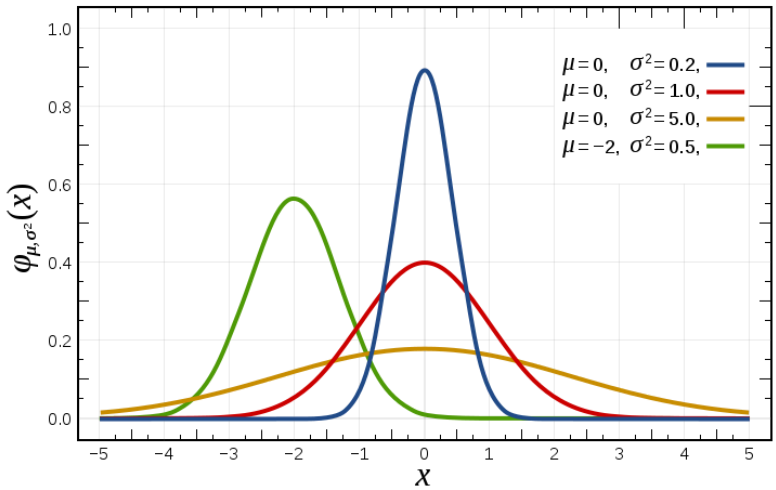
i. Simple Gaussian - Gaussian function is of a form,

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}}$$

Where a, b, c are real constants. The graph of a gaussian function is a symmetric "bell" curve shape and parameter a is height of the curve's peak. B is position of the center of the peak and c is standard deviation which controls the width of the "bell".

The gaussian function composes an exponential function with a concave quadratic function whose log is concave quadratic function.

The normalized gaussian curve is shown below -



Here,
The corresponding parameters are , b = μ and c = σ.

ii. Multivariate Gaussian - It is generalization of the one-dimensional normal distribution to higher dimension. A random vector is said to be a k-variate normally distributed if every linear combination of its k-variate components has a univariate normal distribution.

For k-dimensional random vector X = $[X_1, X_2, ...,X_K]^T$ can be written as,

$$\mathbf{X} \sim \mathcal{N}_k(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

iii. Bernoulli distribution - It is a discrete distribution having two possible outcomes that can be labelled as n = 0(failure) and n =1(success) and can be expressed as,

$$P(n) = \begin{cases} 1-p & \text{for } n = 0 \\ p & \text{for } n = 1, \end{cases}$$

or

$$P(n) = p^n (1-p)^{1-n}.$$

iv. Binomial distribution - It is discrete probability distribution that gives exactly n successes out of N bernoulli trials. This can be expressed as,

X = # of success trials
$f(x) = p(X=x) = {}^{n}c_x \, p^x \, (1-p)^{n-x}$

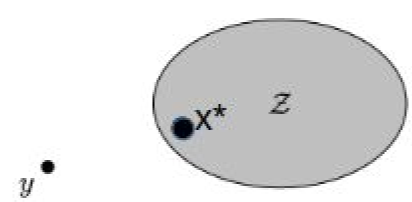
v. Exponential distributions - It is also known as negative exponential distribution which describes the time between events that occur continuously and independently at a constant average time(aka Poisson point process).
It can be expressed as,

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0, \\ 0 & x < 0. \end{cases}$$

e. Relationship between bernoulli and binomial distribution -
A Bernoulli is random variable has two possible outcomes either 1/0. 1 being success and 0 being failure. For e.g. Coin toss which has probability p of result to be heads and probability of 1-p for result to be tails. This is bernoulli's distribution.

A binomial distribution is sum of independent and identically distributed Bernoulli random variable. For e.g. If you toss the coin 100 times, finding the probability of number of heads is binomial distribution.

f. Expected value is 2 since the mean is 2.
g. Answers
   i. x* is 1 since closest natural number from y(1.1) is 1 in N(natural numbers).
   ii. Draw a dot inside the Z plane closest to y.



h. Answers
   i. $^{\infty}\int_{-\infty} p(Y = y)\, dy$
   $= {}^{\infty}\int_{0} e^{-y}\, dy$
   $= [-e^{-y}]^{\infty}$
   $0 - (-1) = 1$

ii. $\mu_Y = E[Y] = {}^{\infty}\int_{y=-\infty} p(Y = y)y\, dy$; - expected value of y?
$\mu_Y = E[Y] = {}^{\infty}\int_{y=-\infty} p(Y = y)y\, dy$
$= {}^{\infty}\int_0 e^{-y}\, y\, dy$
$= [(y+1)\, e^{-y}]^{\infty}_0$
$= (0+1)e^{-0}$
$= 1$

iii. $\sigma\ 2 = Var[Y] = {}^{\infty}\int_{y=-\infty} p(Y = y)(y - \mu\ Y)^2\, dy$
$= {}^{\infty}\int_0 e^{-y}\, (y-1)^2\, dy$
$= [(y^2+1)\, e^{-y}]^{\infty}_0$
$= (0^2+1)e^{-0}$
$= 1$

iv. $E[Y \mid Y \geq 10]$
$= {}^{\infty}\int_{10} e^{-y}\, y\, dy$
$= [(y+1)\, e^{-y}]^{\infty}_{10}$
$= (10+1)e^{-10}$
$= 11e^{-10}$