ECE 1395 Problem Set 1
Bryan Hess: 4259226

ps1-1-a.) A new regression problem that can be solved with machine learning is calculating travel time given traffic on a road. The features (x) used in this problem are how many cars report a slowdown and length of road experiencing traffic (in meters).

ps1-1-b.) The labels (y) in this problem would be the time (in seconds) added to the estimated time of arrival.

ps1-1-c.) Data could be collected using navigation apps (such as google, waze, apple maps, etc.) that update in real time. User's that pull GPS data will in turn report their speed and location. In addition, the actual arrival time at the destination can be used to judge the accuracy of the prediction.

ps1-1-d.) Challenges facing this problem are: collecting user data in real time via the apps, accounting for different travel apps, and calculating the added time from the user data.

ps1-2-a.) A new classification problem that can be solved with machine learning is classifying targeted ads toward users based on internet search quarries. The features (x) used in this problem are keywords in searches, and what given links the user clicks.

ps1-2-b.) The labels (y) in this problem would be the types of products the user may be interested in.

ps1-2-c.) Data could be collected using major search engines (google, bing, duck duck go, etc.). User and device IP could be recorded and every search inquiry will be separated into key words. Then what links clicked by the user will be recorded. From that, ads will be generated. Accuracy of classifications and interest will be judged by if the user interacts with the ads.

ps1-2-d.) Challenges facing this problem are: accounting for different search engines, some users may never click on ads regardless of usefulness, and multiple users on a single device.

ps1-3-a.)

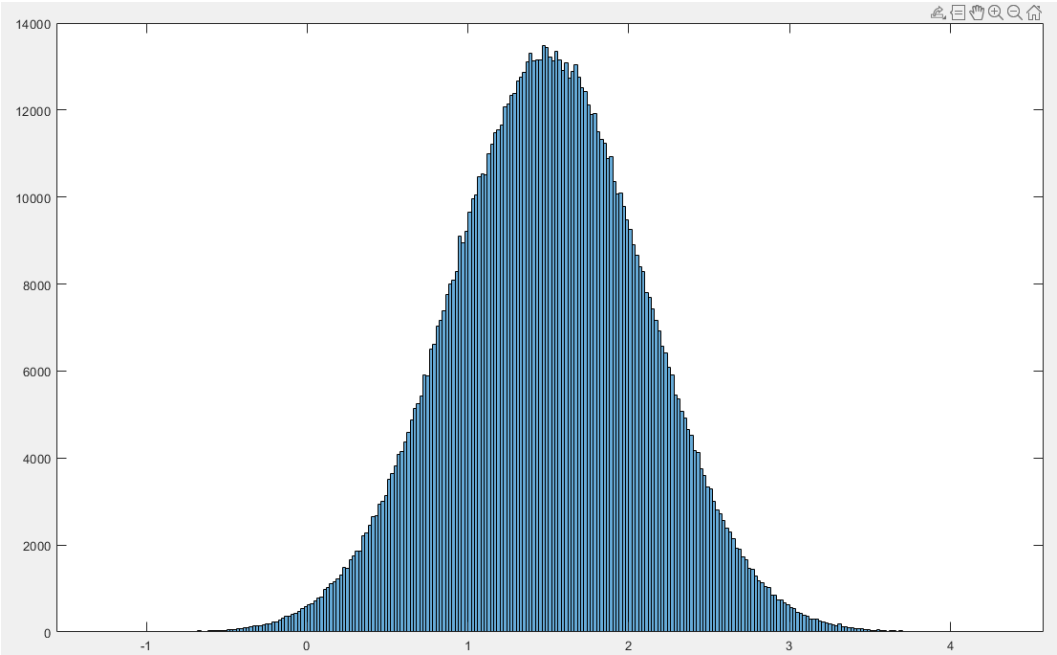
```
%%ps1-3-a.)
x = 1.5 + 0.6.*randn(1000000,1,'double');
```

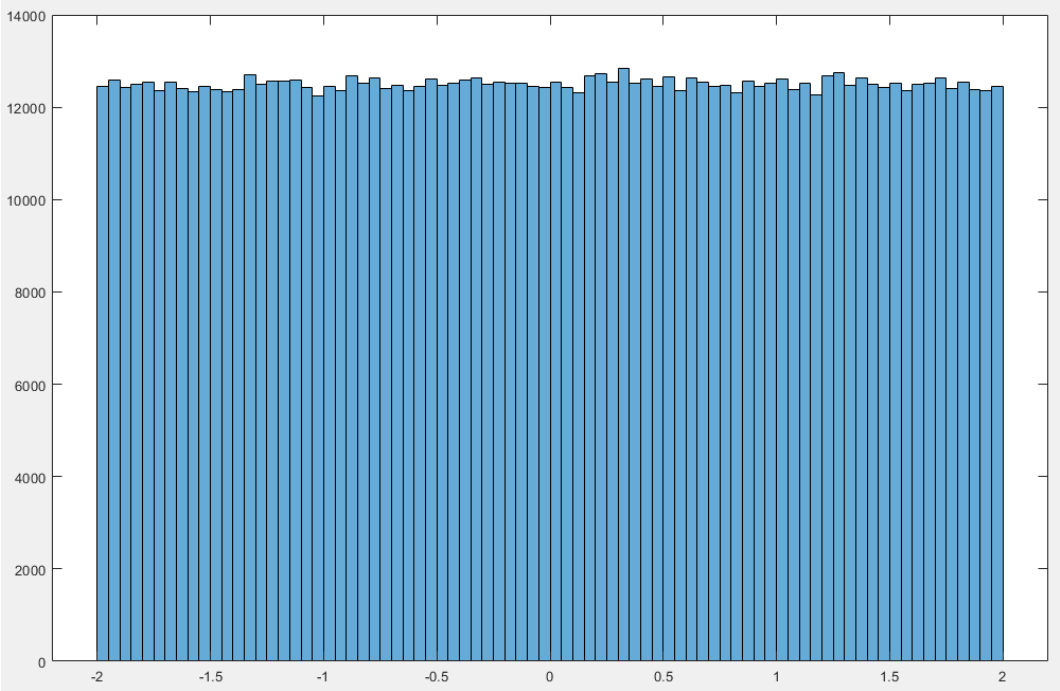
ps1-3-b.)

```
%%ps1-3-b.)
z = (2-(-2)).*rand(1000000,1) - 2;
```

ps1-3-c.)

```
%%ps1-3-c.)
figure(1);
histogram(x)
figure(2);
histogram(z)
```





ps1-3-d.)

```
%%psl-3-d.)
tic
for i = 1:size(x)
        x(i)=x(i)+1;
end
toc
                    Elapsed time is 0.010906 seconds.
```

ps1-3-e.)

```
%%psl-3-e.)
tic
    x = x+1;
elapsedTime = toc;
fprintf('Elapsed time for second opperation is %d seconds', elapsedTime);
%%Adding to each element wihout a loop is much faster than using a loop

Elapsed time is 0.012160 seconds.
Elapsed time for second opperation is 4.002000e-04 seconds
```

Using loops is much slower than simply performing the operation to the entire vector

ps1-3-f.)

```
%%psl-3-f.)
y=[];
for i = 1:size(z)
        if (0<=z(i))&&(z(i)<0.5)
            y(end+1)=z(i);
        end
end
y=y';

%%y comes out to have ≈1/8 the number of elements of z, which makes sense
%%as all positive numbers below 0.5 is 1/8 of the range of the normal
%%distibution z. The number retrieved is slighlty different each run as z
%%isn't a perfect normal distibution
```

ps1-4-a.)

```
%minimum of each row
fprintf('\nMinimum of each column: ');
min(A,[],1)
%minimum of each column
fprintf('\nMaximum of each row: ');
max(A,[],2)
%minimum of whole array
fprintf('\nMinimum of whole array: ');
min(A,[],'all')
%sum of each row
fprintf('\nSum of each row: ');
sum(A,2)
%sum of whole array
fprintf('\nSum of whole array: ');
sum(A,'all')
%square array elements
fprintf('\nSquare of array elements: ');
B = A.^2
```

```
Minimum of each column:
ans =

     2     1     3

Maximum of each row:
ans =

     3
     8
    18

Minimum of whole array:
ans =

     1
```

```
Sum of each row:
ans =

     6
    16
    32

Sum of whole array:
ans =

    54

Square of array elements:
B =

     4     1     9
     4    36    64
    36    64   324
```

ps1-4-b.)

```
%%ps1-4-b.)
syms b n m;
eqn1 = 2*b + n + 3*m == 1;
eqn2 = 2*b + 6*n + 8*m == 3;
eqn3 = 6*b + 8*n + 18*m == 5;
[O,P] = equationsToMatrix([eqn1, eqn2, eqn3], [b, n, m]);
X = linsolve(O,P)
```

```
X =

 3/10
  2/5
    0
```

ps1-4-c.)

L1 Norm x1: $||X_1||_1 = |0.5|+|0|+|-1.5| = 2$

L1 Norm x2: $||X_2||_1 = |1|+|-1|+|0| = 2$

L2 Norm x1: $||X_1|| = |0.5|^2+|0|^2+|-1.5|^2) = \sqrt{(0.25+2.25)} = 1.58113883008$

L2 Norm x2: $||X_2|| = (|1|^2+|-1|^2+|0|^2) = \sqrt{(1+1)} = 1.41421356237$

```
%%ps1-4-c.)
x1 = [0.5 0 -1.5];
x2 = [1 -1 0];

%%L1 norm
fprintf('\nL1 norm for x1: ');
norm(x1,1)
fprintf('\nL1 norm for x2: ');
norm(x2,1)
%%L2 norm
fprintf('\nL2 norm for x1: ');
norm(x1)
fprintf('\nL2 norm for x2: ');
norm(x2)
```

```
L1 norm for x1:
ans =

     2


L1 norm for x2:
ans =

     2


L2 norm for x1:
ans =

    1.5811


L2 norm for x2:
ans =

    1.4142
```

ps1-5.)

```
%%ps1-5.)
mat1 = [1,2,3;4,5,6];
mat2 = [1,2;3,4;5,6];

normal1 = normalize_col(mat1)
normal2 = normalize_col(mat2)

function [B] = normalize_col(A)
    [NumRows, NumCols]=size(A);
    for i = 1:NumCols
       B(:,i) = A(:,i)./sqrt(sum(A(:,i).^2));
    end
end
```

```
normal1 =

    0.2425    0.3714    0.4472
    0.9701    0.9285    0.8944


normal2 =

    0.1690    0.2673
    0.5071    0.5345
    0.8452    0.8018
```