

```
clear
clc
tic
```

Práctica 7. Regresión multivariable.

Ian Gabriel Cañas Fernández, 1092228

En el presente laboratorio se estará trabajando una regresión multivariable tal y como se apreció en el laboratorio anterior. Trabajando con la misma base de datos anterior.

Primeramente, se hará una aleatorización y segmentación de los datos.

```
data = readtable('auto-mpg.csv');
dataset = data(randperm(length(data.mpg)), :);
% "Dataset" porque son los datos con los que trabajaremos

dataset.horsepower = fillmissing(dataset.horsepower(:), 'constant', ...
    nanmean(dataset.horsepower));

i = round(length(dataset.mpg)*0.75);

traindata = dataset(1:i, :);
testdata = dataset((i+1):end, :);

i = length(dataset.cylinders);
D1 = ones(i, 1);

X = [traindata.mpg traindata.displacement traindata.horsepower traindata.weight traindata.acceleration];
X2 = [testdata.mpg testdata.displacement testdata.horsepower testdata.weight testdata.acceleration];
r = traindata.cylinders;
r2 = testdata.cylinders;
```

A continuación se computaron los pesos de la regresión lineal multivariable utilizando el algoritmo ya conocido.

```
data2 = readtable('auto-mpg.csv');
dataset2 = data2(randperm(length(data2.mpg)), :);
% "Dataset" porque son los datos con los que trabajaremos

i = round(length(dataset2.mpg)*0.75);

traindata2 = dataset2(1:i, :);
testdata2 = dataset2((i+1):end, :);

i = length(traindata2.cylinders);
D2 = ones(i, 1);

traindata2 = dataset2(1:i, :);
testdata2 = dataset2((i+1):end, :);
```

```

i = length(traindata2.cylinders);
j = length(testdata2.cylinders);

X2 = [ones(j,1) testdata2.cylinders testdata2.displacement testdata2.weight testdata2.acceleration];
X = [ones(i,1) traindata2.cylinders traindata2.displacement traindata2.weight traindata2.acceleration];
r = traindata2.mpg;
r2 = testdata2.mpg;

%for d = 2:length(X(1,:))
    d = 2:length(X(1,:))

```

```

d = 1x6
     2     3     4     5     6     7

```

```

disp('Pesos y error para polinomio de grado ')

```

Pesos y error para polinomio de grado

```

disp(d-1)

```

```

     1     2     3     4     5     6

```

```

wA3 = inv(X(:,1:d)'*X(:,1:d))*X(:,1:d)'*r

```

```

wA3 = 2x1
    42.9090
    -3.5776

```

```

mpgML = X2(:,1:d)*wA3;

```

```

e3 = [mpgML testdata2.mpg (mpgML-testdata2.mpg).^2];
emean1 = mean(e3(:,3))

```

```

emean1 = 26.2523

```

```

%end

```

Luego, se aplica kNN para comparar la eficiencia.

```

k=3;
for i = 1:length(testdata.mpg)
    i;
    d = ((X2(i,:) - X).^2);
    d = sqrt(d(:,1)+d(:,2));

    [dis, pos] = mink(d, k);
    MPG_ML2(i,1) = mode(r(pos));
end

%kNNlda = sum(MPG_ML2==r2(:))/length(testdata.mpg)
emeanNN1 = mean((MPG_ML2-r2(:)).^2)

```

emeankNN1 = 42.0683

Para una segunda permutación

```
data = readtable('auto-mpg.csv');
dataset = data(randperm(length(data.mpg)), :);

dataset.horsepower = fillmissing(dataset.horsepower(:), 'constant', nanmean(dataset.horsepower));

i = round(length(dataset.mpg)*0.75);

traindata = dataset(1:i, :);
testdata = dataset((i+1):end, :);

i = length(dataset.cylinders);
D1 = ones(i, 1);

X = [traindata.mpg traindata.displacement traindata.horsepower traindata.weight traindata.acceleration];
X2 = [testdata.mpg testdata.displacement testdata.horsepower testdata.weight testdata.acceleration];
r = traindata.cylinders;
r2 = testdata.cylinders;
```

A continuación se computaron los pesos de la regresión lineal multivariable utilizando el algoritmo ya conocido.

```
data2 = readtable('auto-mpg.csv');
dataset2 = data2(randperm(length(data2.mpg)), :);
% "Dataset" porque son los datos con los que trabajaremos

i = round(length(dataset2.mpg)*0.75);

traindata2 = dataset2(1:i, :);
testdata2 = dataset2((i+1):end, :);

i = length(traindata2.cylinders);
D2 = ones(i, 1);

traindata2 = dataset2(1:i, :);
testdata2 = dataset2((i+1):end, :);

i = length(traindata2.cylinders);
j = length(testdata2.cylinders);

X2 = [ones(j,1) testdata2.cylinders testdata2.displacement testdata2.weight testdata2.acceleration];
X = [ones(i,1) traindata2.cylinders traindata2.displacement traindata2.weight traindata2.acceleration];
r = traindata2.mpg;
r2 = testdata2.mpg;

%for d = 2:length(X(1,:))
    d = 2:length(X(1,:))
```

```
d = 1×6
    2    3    4    5    6    7
```

```
disp('Pesos y error para polinomio de grado ')
```

Pesos y error para polinomio de grado

```
disp(d-1)
```

```
1    2    3    4    5    6
```

```
wA3 = inv(X(:,1:d)'*X(:,1:d))*X(:,1:d)'*r
```

```
wA3 = 2×1
    42.2457
   -3.4474
```

```
mpgML = X2(:,1:d)*wA3;
```

```
e3 = [mpgML testdata2.mpg (mpgML-testdata2.mpg).^2];
emean2 = mean(e3(:,3))
```

```
emean2 = 20.7041
```

```
%end
```

Luego, se aplica kNN para comparar la eficiencia.

```
k=3;
for i = 1:length(testdata.mpg)
    i;
    d = ((X2(i,:) - X).^2);
    d = sqrt(d(:,1)+d(:,2));

    [dis, pos] = mink(d, k);
    MPG_ML2(i,1) = mode(r(pos));
end

%kNNlda = sum(MPG_ML2==r2(:))/length(testdata.mpg)
emeankNN2 = mean((MPG_ML2-r2(:)).^2)
```

```
emeankNN2 = 38.0756
```

```
toc
```

Se ha podido observar un mejor rendimiento mediante el uso de kNN hasta cierto punto, solo que este se ve limitado a etiquetas discretas o contables. Mientras que por regresión se puede tener mejor certidumbre ante las etiquetas continuas y de valores no vistos antes.