

**Universidad San Francisco Xavier de Chuquisaca**

**Facultad de Tecnología**



**Informe laboratorio 4 – Sis420**

**Nombre:** Arancibia Aguilar Daniel Andree

**Carrera:** Ingeniería en Ciencias de la Computación

## Introducción

En este laboratorio entrenamos nuevamente los modelos de regresión lineal multivariable, regresión polinómica, ecuación de la normal, estos 3 con el dataset "Maintenance Naval Vessel"

Para la regresión logística se entrenó con el dataset "Heart disease indicator"

Para la regresión OneVsAll se entrenó con el dataset "Letter Recognition"

Para el laboratorio use el código de los anteriores laboratorios, pero esta vez agregando la regularización en las funciones y luego haremos una comparativa entre los resultados haciendo uso de la regularización para ver cuál es más eficiente

Todos los datasets fueron preprocesados con PANDAS con el cual se pudo ver si habían valores nulos o si debíamos hacer modificaciones

## Regresión Linear Multivariable y regresión polinómica

En este modelo usamos las funciones

`featureNormalize()`: Con esta función normalizamos las características del dataset y además se agregó  $\epsilon$  al valor de  $\sigma$  para evitar la división entre 0

`computeCostMulti()`: Calcula la función de costo para regresión lineal con múltiples variables, se agregó la función `computeCostMultireg()` para la regularización

`gradientDescentMulti()`: Implementa el algoritmo de descenso de gradiente para optimizar los parámetros de la regresión lineal, se agregó la función `gradientDescentMultireg()` para la regularización

## Ecuación de la normal

`normalEqn()`: Calcula los parámetros de la regresión lineal mediante la ecuación normal, se agregó la función `normalEqnreg()` para la regularización

## Regresión logística

`Sigmoid()`: Calcula la función sigmoide, utilizada en la regresión logística para transformar las predicciones en probabilidades

`calcularCosto()`: Calcula la función de costo y el gradiente para la regresión logística, se agregó la función `calcularCostoreg()` para la regularización

`descensoGradiente()`: Implementa el algoritmo de descenso de gradiente para optimizar los parámetros en el contexto de la regresión logística, se agregó la función `descensoGradientereg()` para la regularización

`predict()`: Realiza predicciones utilizando los parámetros aprendidos en un modelo de regresión logística o clasificación binaria.

## Regresion OneVsAll

calcularSigmoide(): Calcula la función sigmoide, utilizada en la regresión logística para transformar las predicciones en probabilidades.

calcularCostoReg(): Calcula la función de costo y el gradiente para la regresión logística con regularización se agregó la función calcularCosto() sin regularización.

oneVsAllReg(): Implementa el entrenamiento de un clasificador de regresión logística one-vs-all con regularización, se agregó la función oneVsAll() sin regularización.

## Comparación modelos con y sin regularización

### Regresión linear multivariable

```
theta calculado sin regularizacion: [ 9.88460581e-01  1.24028926e-04  1.15193327e-04  1.11327789e-03
 4.69503530e-04  1.11199432e-03  1.04326868e-03  1.04326868e-03
-2.99617415e-03  0.00000000e+00 -1.16049287e-03  5.37584866e-04
 0.00000000e+00 -1.27748354e-03  6.10743708e-04 -2.00773952e-04
-5.49793398e-04 -1.71453123e-04]
theta calculado con regularizacion: [ 9.88459799e-01  9.67392300e-05  3.61868299e-05  2.26583925e-03
 8.71182579e-04  1.71488211e-03  1.81339472e-03  1.81339472e-03
-4.86927796e-03  0.00000000e+00 -1.94881547e-03  1.16345780e-03
 0.00000000e+00 -2.31943730e-03  4.51711094e-04 -4.27804384e-04
-6.71358886e-04 -2.81632930e-04]
El TDSC sin regularización: 0.9884
El TDSC con regularización: 0.9883
```

En ambos casos el resultado fue muy similar pero sin la regularización se consiguió un valor mas alto

### Regresión Polinomial

```
Sin regularización 1.842013860637511
Con regularización 0.9856825900763946
```

En este caso el valor sin regularizar fue mas alto, pero el mas cercano según el dataset es el valor regularizado

### Ecuación de la normal

```
Theta calculado a partir de la ecuación de la normal: [-6.10383809e+01  2.02850122e+01 -3.85501192e-04 -6.05609360e-03
 2.87329935e-03  3.98100046e+10 -3.98100046e+10 -1.09945703e-03
 6.83967251e-04 -2.39854863e-04  1.35439487e-01 -2.50223257e-01
-4.86981147e-02  2.46734875e-01 -2.44382907e-04  2.90570311e-01
-1.44971018e-01]
El TDSC sin regularización: 1.8420
```

```
Theta regularizada [ 1.47025375e-02  4.47547156e-04  1.27240376e-05  4.77319272e-05
 2.42065647e-05 -7.01361272e-04 -7.00606823e-04 -5.29790726e-04
 4.50420101e-03 -1.72999480e-04  1.33564403e-01  1.56396613e-05
-4.84325118e-02  2.87932492e-02 -1.61902058e-04  2.33783880e-01
-1.19313327e-01]
El TDSC con regularización: 0.9857
```

Se consiguió una diferencia en los resultados siendo el resultado sin regularizar el mayor, pero el que mas se asemeja a los datos seria el resultado regularizado, se vio los mismos resultados que en la regresion polinomial

## Regresión logística

```
Precisión con regularizacion: 90.58 %  
Precisión sin regularizacion: 90.58 %
```

En ambos casos se obtuvo la misma precisión

```
Predicciones de muerte por ataque al corazon sin regularización:0.1454  
Predicciones de muerte por ataque al corazon con regularización:0.1469
```

Las predicciones variaron solo un poco

## Regresión OneVsAll

```
Precisión del conjunto de entrenamiento con regularización: 73.12%  
Precisión del conjunto de entrenamiento sin regularización: 73.12%  
[[ 1.  3.  8. ...  8.  7.  7.]  
 [ 1.  2.  7. ...  6.  0.  8.]  
 [ 1.  2.  4. ... 10.  2.  5.]  
 ...  
 [ 1.  1.  3. ...  8.  1.  8.]  
 [ 1.  2.  0. ...  8.  0.  8.]  
 [ 1.  3.  6. ...  7.  7.  8.]]  
Predicciones en el conjunto de prueba con regularización:  
[ 5 10 20 22 12 15 14  6  7 10  4 11  2 19  3 23 21  9  7 14  5 22 24  7  
 25 21 12  8 16 13 13  6 11  8 17 14 19  1 20  3 16  3  2 25  7  9 17  2  
  3  3 20  7 20  3 26 21 11 18 13 20  3 17  8 24  4 13 21  8  3 11  9 18  
 15 11 26  1  3 21  9 20  4  3  1 10 12 23 22  3 25 25 17 13 26 23 21 23  
 11  9 12 18]  
Predicciones en el conjunto de prueba sin regularización:  
[ 5 10 20 22 12 15 14  6  7 10  4 11  2 19  3 23 21  9  7 14  5 22 24  7  
 25 21 12  8 16 13 13  6 11  8 17 14 19  1 20  3 16  3  2 25  7  9 17  2  
  3  3 20  7 20  3 26 21 11 18 13 20  3 17  8 24  4 13 21  8  3 11  9 18  
 15 11 26  1  3 21  9 20  4  3  1 10 12 23 22  3 25 25 17 13 26 23 21 23  
 11  9 12 18]
```

En este caso se consiguió la misma precisión y mismos resultados en ambos casos

## Conclusiones:

En los modelos entrenados se pudo ver solo una pequeña variación en los resultados teniendo incluso la misma precisión