CENTRO DE ENSEÑANZA TÉCNICA Y SUPERIOR

Escuela de Ingeniería

Ciencias Computacionales

Materia: Aprendizaje De Máquina

Presenta:

Pablo Díaz 030343

Kevin Huerta 30502

Gerardo Hernández 29902

Diego Garibay 30046

Profesor: Ulises Orozco Rosas

Practica #1: Acondicionamiento

Tijuana, B.C., 26 de agosto de 2020

Aprendizaje de Máquina
Practica 01

Pablo Díaz, *30343,* Kevin Huerta, *30502,* Gerardo Hernández, *29902,* Diego Garibay, *30046*

Abstract—En este reporte mencionaremos como fue el proceso que tomó hacer la práctica 1 de la materia Aprendizaje de máquina. Los conocimientos que debíamos tener para realizar dichas asignaciones, después mostraremos resultdos y daremos a conocer respectivas conclusiones.

Index Terms—Arreglos, Ciclos, Python, R

1 Introducción

Este reporte tiene como objetivo explicar las metodologías aplicadas en los programas resueltos a través de dos lenguajes de programación: Python y R. Para comenzar Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad. [1].En la actualidad es uno de los lenguajes más utilizados, debido a su simplicidad y manejo, además de que tiene bastante cantidad de librerías que te ayudan al manejo de Machine Learning y Big Data, en ocasiones se usa para la creación de videojuegos aunque no es muy eficiente como lo son otros lenguajes de programación, entre ellos Java o C++. Con este lenguaje no tenemos problema, ya que lo hemos usado en la escuela en materias pasadas, aparte lo hemos usado en proyectos personales. Tenemos conocimientos básicos e intermedios del lenguaje, y en el curso pensamos que aumentaremos el nivel de programación en este lenguaje.

Por otra parte, tenemos el lenguaje de R. R es lenguaje de programación interpretado, es decir, ejecuta las instrucciones directamente, sin una previa compilación del programa a instrucciones en lenguaje máquina. El término entorno, en R, se refiere a un sistema totalmente planificado y coherente, en lugar de una acumulación de herramientas específicas e inflexibles, como suele ser el caso en otros softwares de análisis de datos. [2] Se suele utilizar para la computación estadística, debido a que tiene varias herramientas útiles para el manejo de gran cantidad de datos. Está enfocado en la manipulación, procesamiento y visualización gráfica de los datos. El propósito de R fue hacer estadística, y está disponible para LINUX, MAC OS, Y Windows. Al principio sentíamos que iba a ser complicado aprender de nuevo un nuevo lenguaje de programación, pero al momento de probarlo nos dimos cuenta que era muy parecido a python y nos gustó. El manejo de los datos era muy parecido, la sencillez, flexibilidad y entendimiento del código se nos hizo una ventaja muy grande de este lenguaje.

En el contenido del reporte se explicarán las metodologías utilizadas en los problemas encargados por el profesor, así mismo mostraremos los fundamentos necesarios para lograr solucionarlos, además mostraremos los resultados obtenidos y diremos las conclusiones de la práctica.

2 Fundamentos

Con base en nuestra experiencia al leer vectores, y al ver las figuras que consistían en un conjunto de patrones (asteriscos), determinamos que se usarían ciclos, vectores, declaración de variables y por último If-Else. Por lo tanto, nos dimos a la tarea de investigar esto para R, por que era la primera vez que lo usábamos. Y también buscamos estos términos para Python, para refrescar los conocimientos que tenemos.

2.1 Declaración de variables

En Python es sencillo, basta con escribir un nombre (identificador) seguido del "=" y el valor de la variable. Ejemplos:

- a = 3

- numero = 6

Es importante resaltar que el identificador no puede estar separado por espacios en blanco.
En R la declaración de variables tiene un pequeño cambio en sintaxis, aunque prácticamente tiene la misma funcionalidad: nombre (identificador) seguido de los caracteres ← y el valor de la variable. Pero también se pueden declarar variables de forma invertida y trabaja igual, observemos los ejemplos para que nos quede más claro. Ejemplos: [3]

- a ← 3
- 3 → n

2.2 Declaración de Arreglos

Para Python necesitaremos importar una librería llamada Numpy que sirve para gestionar de mejor manera vectores y matrices en caso de necesitarlas. Esta librería nos servirá mucho para ahorrar líneas de código y que nuestro archivo sea más compacto. Entonces, para declarar arreglos de la librería numpy se hace: nombre de la variable seguido de "=" np.array([elemento1, elemento2, elemento3, etc]). Ejemplos:

- x1 = np.array([43,44,45,46,47])
- y1 = np.array([41,45,49,47,44])

En R se facilita la declaración de arreglos. Para declarar arreglos se usa la función c, que viene de combine, y los elementos dentro del paréntesis deben estar separados por comas. Los valores de los arreglos pueden ser booleanos, numéricos o cadenas. Estos son algunos ejemplos: [4]

- vectornumerico ← c(1, 3, 5, 7) #vector numérico
- vectortexto ← c("a", "b", "c", "d") #vector de texto
- vectorlogico ← c(TRUE, FALSE, TRUE) #vector lógico / booleano

2.3 Ciclos

Los ciclos en Python tienen una sintaxis un tanto diferente de lenguajes como C, Java,etc. La sintaxis es la siguiente:

```
for i in range(1,10):
    Code a ejecutar
```

Donde se utiliza la función range para recorrer una serie de números, el cual puede recibir 1 o 2 parámetros. Cuando se utiliza 1 parámetro, el for se repite desde el 0 hasta el parámetro recibido. Cuando son 2 parámetros, recorre desde el primer parámetro hasta el segundo.
Ahora, los ciclos en R tienen una sintaxis muy parecida a python,y su sintaxis es la siguiente:

```
for(<variable> in <objeto iterable>) <codigo>
```

Cuando se tiene un vector, se recorrería de la siguiente manera:

```
x <- c("a", "b", "c", "d")
for(i in seq_along(x)) {
    print(x[i])
}
```

Se usa la función seq_along la cual genera una secuencia numérica que usaremos para indicar los índices de los elementos de los objetos que queremos recorrer. [5]

2.4 If-Else

Los if-else sirven para verificar una condición. Si la condición dentro del if es verdadera entonces se ejecutará el código dentro de la sentencia If, en caso contrario, se ejecutará el código de la sentence else.

Sintaxis en Python:

```
if  x > y:
  print("x mayor que y ")
Else:
  print("y mayor que x ")
```

Sintaxis en R: [6]

```
if(condicion) {
  operaciones_si_la_condici n_es_TRUE
}
else {
  operaciones_si_la_condici n_es_FALSE
}
```

3 METODOLOGÍA

La metodología a utilizar fue una implementación bastante simple, debido a dos principales razones. La primera es que la naturaleza de la actividad como tal, es una actividad sencilla de nivel elemental para programadores, junto al factor de que se programa en lenguajes de alto nivel. Esto nos ayuda a manejar una comprensión más humana y no tanto computacional. Además, por el hecho que la actividad en sí misma, es una actividad de acondicionamiento, donde no se maneja teoría referente al tema, por lo cual solo se requiere un nivel elemental para la solución de la misma.

Ahora en cuanto a como realizamos las actividades, primero como equipo establecimos la forma en la que íbamos a resolver cada una de las actividades, independientemente de en que lenguaje. Esto quedó de la siguiente forma:

En el 1.1 solamente usamos print() para mostrar un mensaje consola que en este caso era "Hello World !".

En el 1.2 creamos una función con nombre reconocible (para que nos sirva en futuras actividades) la cual es una función de segundo grado. Después solo se evalúa en x=1.

En el 1.3 creamos el vector X=-2,-1,0,1,2. que se nos especifica y lo evaluamos en la función del 1.2. Imprimimos los resultados.

En el 1.4 hacemos lo mismo que en el 1.3 solo que con un vector que creamos que va desde -2 a 2 con una separación de 0.1. Esto es fácil de hacer con NumPy arange() o en el caso de R con seq()

En el 1.5 evaluamos el vector creado en 1.4 en tres funciones:

$$sen(x), log_{10}(x), e^x$$

En Python lo hicimos con lambda y en R aplicamos directamente la función predeterminada al vector. Al final mostramos el resultado y lo graficamos. Se debe mencionar que como log10 no se pude calcular para valores menores o iguales a 10, hicimos algunos ajustes.

En el 1.6 creamos dos vectores que se nos especifica y efectuamos operaciones con matrices, multiplicación, división, potencia, suma y resta. Todo esto es sencillo y directo con NumPy y con R.

En el 1.7 creamos dos matrices y efectuamos un producto punto entre dos matrices. Todo esto es sencillo y directo con NumPy y con R.

En el 1.8 elaboramos funciones que funcionan a base de ciclos e imprimen las figuras que se nos solicita. En el 1.9, al igual que en el 1.8, creamos funciones que imprimen el patrón solicitado. En Python creamos dos versiones, una en la que van creciendo los triángulos y otra en la que no.

En el 1.10 nos basamos en la idea de como se forma la parte superior de la flecha y creamos un triángulo. Antes de imprimirse le solicita al usuario escribir la altura.

4 Resultados

4.1 Python:

1) Los resultados obtenidos fueron la impresión de la frase "Hello, World!"

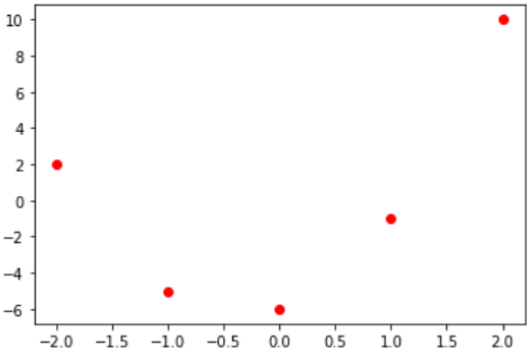
```
Hello, World!
```

2) El resultado obtenido al evaluar la función $3x^2 + 2x - 6$ donde x=1 da un resultado de -1

```
-1
```

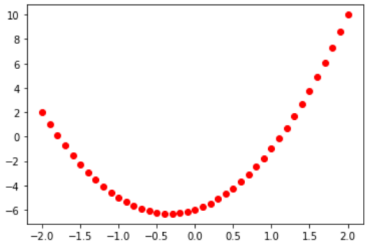
3) Al evaluar el vector dado nos da de resultado la siguiente gráfica

```
[ 2 -5 -6 -1 10]
```



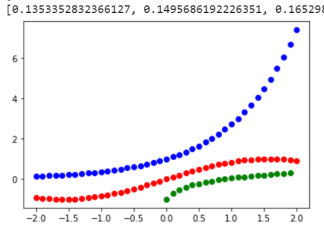
4) Al evaluar el vector dado nos da de resultado la siguiente gráfica

```
[ 2.    1.03  0.12 -0.73 -1.52 -2.25 -2.92 -3.53 -4.08 -4.57 -5.   -5.37
 -5.68 -5.93 -6.12 -6.25 -6.32 -6.33 -6.28 -6.17 -6.   -5.77 -5.48 -5.13
 -4.72 -4.25 -3.72 -3.13 -2.48 -1.77 -1.   -0.17  0.72  1.67  2.68  3.75
  4.88  6.07  7.32  8.63 10.  ]
```



5) Utilizando el vector creado en la actividad 1.4 y evaluando con las funciones dadas obtenemos estos resultados

```
[-0.9092974268256817, -0.9463000876874145, -0.9738476308781953, -0.9916648104524687,
[-0.9999999999999919, -0.6989700043360145, -0.5228787452803346, -0.3979400086720353,
[0.1353352832366127, 0.1495686192226351, 0.1652988882215866, 0.1826835240527347, 0.1
```



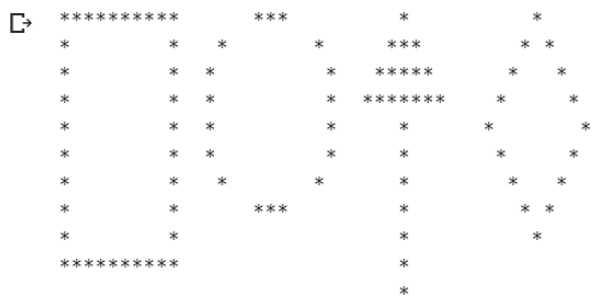
6) Realizando las operaciones a los dos vectores creados nos otorga los siguiente resultados

```
Resultado v1*v2: [ 2  8 18 32]
Resultado v1/v2: [0.5 0.5 0.5 0.5]
Resultado v1^2:  [ 1  4  9 16]
Resultado v1+v2: [ 3  6  9 12]
Resultado v1-v2: [-1 -2 -3 -4]
```

7) Realizando la multiplicación de matrices nos queda de la siguiente manera

```
[[ 0  7]
 [ 7 -1]
 [-5 -3]]
```

8) Con el uso de ciclos se pudo obtener los siguientes resultados



9) Como la actividad anterior con el uso de ciclos se obtuvo este resultado

Versión 1:

Versión 2:

10) Creando nuestra propia función obtuvimos el siguiente resultado

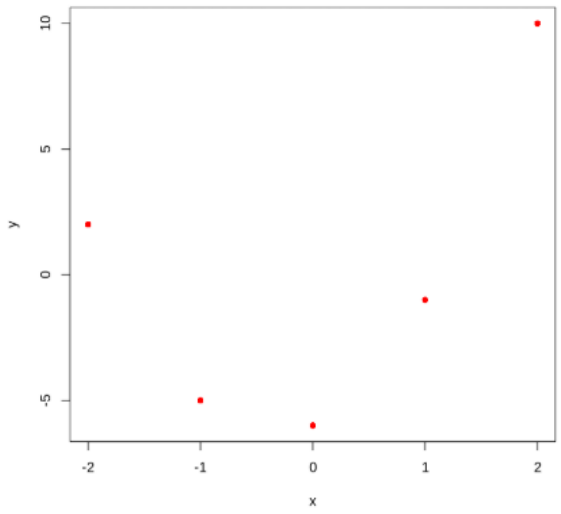
4.2 R:

1) Los resultados obtenidos fueron la impresión de la frase “Hello, World!”

2) El resultado obtenido al evaluar la función $3x^2 + 2x - 6$ donde x=1 da un resultado de -1

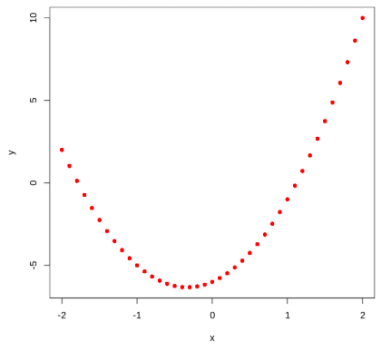
3) Al evaluar el vector dado nos da de resultado la siguiente gráfica

```
[1]  2 -5 -6 -1 10
```



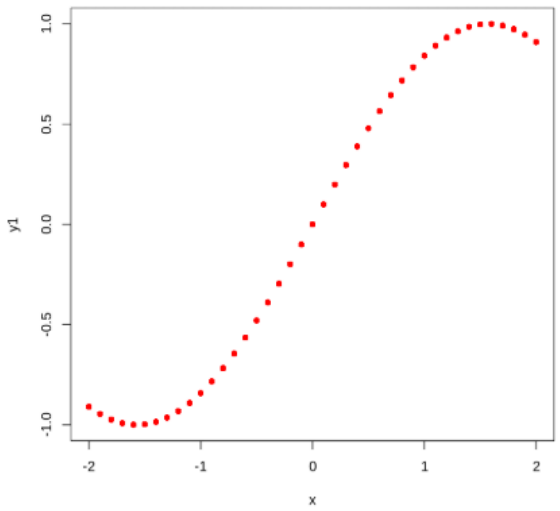
4) Al evaluar el vector dado nos da de resultado la siguiente gráfica

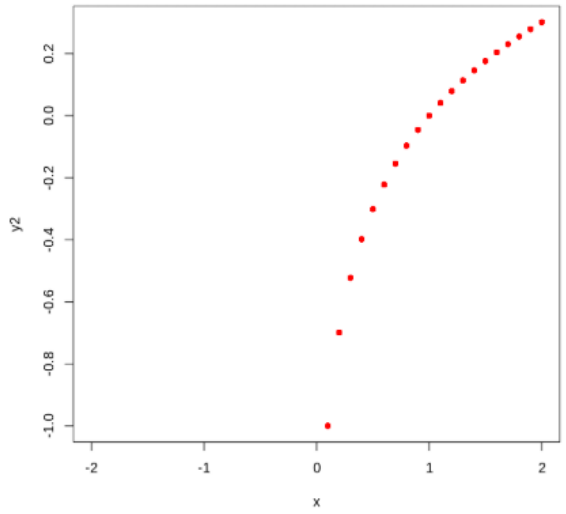
```
 [1]  2.00  1.03  0.12 -0.73 -1.52 -2.25 -2.92 -3.53 -4.08 -4.57 -5.00 -5.37
[13] -5.68 -5.93 -6.12 -6.25 -6.32 -6.33 -6.28 -6.17 -6.00 -5.77 -5.48 -5.13
[25] -4.72 -4.25 -3.72 -3.13 -2.48 -1.77 -1.00 -0.17  0.72  1.67  2.68  3.75
[37]  4.88  6.07  7.32  8.63 10.00
```

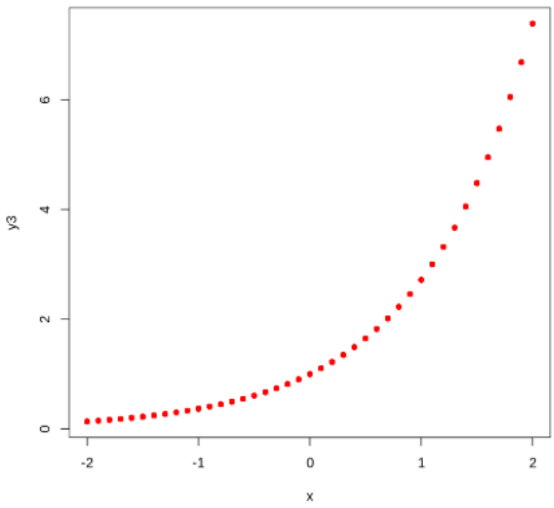


5) Utilizando el vector creado en la actividad 1.4 y evaluando con las funciones dadas obtenemos estos resultados

```
Warning message in eval(expr, envir, enclos):
"NaNs produced"

Seno(x):
 [1] -0.90929743 -0.94630009 -0.97384763 -0.99166481 -0.99957360 -0.99749499
 [7] -0.98544973 -0.96355819 -0.93203909 -0.89120736 -0.84147098 -0.78332691
[13] -0.71735609 -0.64421769 -0.56464247 -0.47942554 -0.38941834 -0.29552021
[19] -0.19866933 -0.09983342  0.00000000  0.09983342  0.19866933  0.29552021
[25]  0.38941834  0.47942554  0.56464247  0.64421769  0.71735609  0.78332691
[31]  0.84147098  0.89120736  0.93203909  0.96355819  0.98544973  0.99749499
[37]  0.99957360  0.99166481  0.97384763  0.94630009  0.90929743


Log10(x):
 [1]         NaN         NaN         NaN         NaN         NaN         NaN
 [7]         NaN         NaN         NaN         NaN         NaN         NaN
[13]         NaN         NaN         NaN         NaN         NaN         NaN
[19]         NaN         NaN        -Inf -1.00000000 -0.69897000 -0.52287875
[25] -0.39794001 -0.30103000 -0.22184875 -0.15490196 -0.09691001 -0.04575749
[31]  0.00000000  0.04139269  0.07918125  0.11394335  0.14612804  0.17609126
[37]  0.20411998  0.23044892  0.25527251  0.27875360  0.30103000


e^x:
 [1] 0.1353353 0.1495686 0.1652989 0.1826835 0.2018965 0.2231302 0.2465970
 [8] 0.2725318 0.3011942 0.3328711 0.3678794 0.4065697 0.4493290 0.4965853
[15] 0.5488116 0.6065307 0.6703200 0.7408182 0.8187308 0.9048374 1.0000000
[22] 1.1051709 1.2214028 1.3498588 1.4918247 1.6487213 1.8221188 2.0137527
[29] 2.2255409 2.4596031 2.7182818 3.0041660 3.3201169 3.6692967 4.0552000
[36] 4.4816891 4.9530324 5.4739474 6.0496475 6.6858944 7.3890561
```







6) Realizando las operaciones a los dos vectores creados nos otorga los siguiente resultados

```
Resultado v1*v2: 2 8 18 32
Resultado v1/v2: 0.5 0.5 0.5 0.5
Resultado v1^2:  1 4 9 16
Resultado v1+v2: 3 6 9 12
Resultado v1-v2: -1 -2 -3 -4
```

7) Realizando la multiplicación de matrices nos queda de la siguiente manera

```
     [,1] [,2]
[1,]    0    7
[2,]    7   -1
[3,]   -5   -3
```

8) Con el uso de ciclos se pudo obtener los siguientes resultados

9) Como la actividad anterior con el uso de ciclos se obtuvo este resultado

10) Creando nuestra propia función obtuvimos el siguiente resultado

```
Ingrese la altura: 10
                   *
                  ***
                 *****
                *******
               *********
              ***********
             *************
            ***************
           *****************
          *******************
```

5 Conclusiones

En conclusión esta actividad claramente nos sirvió para calentar y reforzar aquellas áreas que teníamos débiles sobre el manejo de los lenguajes R y Python. También nos ayuda a utilizar nuevas herramientas como colab las cuales nos permite el uso múltiple de usuarios, en un solo archivo. Lo cual nos facilita en gran medida la solución de dichos ejercicios presentes en esta práctica. Los resultados de esta práctica fueron los esperados, por ende nos sentimos satisfechos con la realización de la misma.

References

[1] M. A. Alvarez. (2003) Qué es python. [Online]. Available: https://desarrolloweb.com/articulos/1325.php

[2] U. Revista. (2019) Lenguaje r, ¿qué es y por qué es tan usado en big data? [Online]. Available: https://www.unir.net/ingenieria/revista/noticias/lenguaje-r-big-data/549204698515/

[3] E. Paradis. (2003) R para principiantes. [Online]. Available: https://cran.r-project.org/doc/contrib/rdebuts$_es.pdf$
[4] V. el software libre. Vectores en r: QuÉ son y cÓmo trabajar con ellos. [Online]. Available: https://cran.r-project.org/doc/contrib/rdebuts$_es.pdf$
[5] Mauricio. (2016) Curso de r — estructuras de control. [Online]. Available: https://mauricioanderson.com/curso-r-estructuras-control/
[6] bookdown. 9.1 if, else. [Online]. Available: https://bookdown.org/jboscomendoza/r-principiantes4/if-else.html