**AprendizajeAutomáticoTC3020**

**Reporte Práctica 5**

**Dr. Jesús Guillermo Falcón Cardona**

**Introducción**

La practica 5 consistió en tres partes, en la primera se puso en practica lo visto sobre optimización de enjambre de partículas en la cual el profesor proporciono un código en C el cual tuvo que se modificado y los resultados que se obtenían se debían animar.

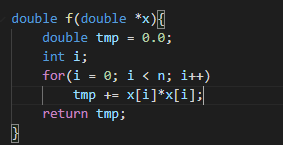
La segunda parte fue poner en práctica la teoría vista sobre Naïve Bayes en donde se tenían que obtener la probabilidad de sucesos de un dataset dado.

La última parte consistió en hacer un programa usando la librería de Naïve Bayes de skicit learn para clasificar el dataset de Iris, obtener su precisión y después graficar las instancias clasificadas.

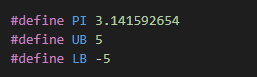
**Reporte**

**Parte 1: Particle Swarm Optimization**

1. La primera parte solo cambie dentro de la función f() la formula dentro del ciclo tal y como se indicaba en las instrucciones, se borro también el aumento que se hizo en la variable temp. El código quedo de la siguiente manera:



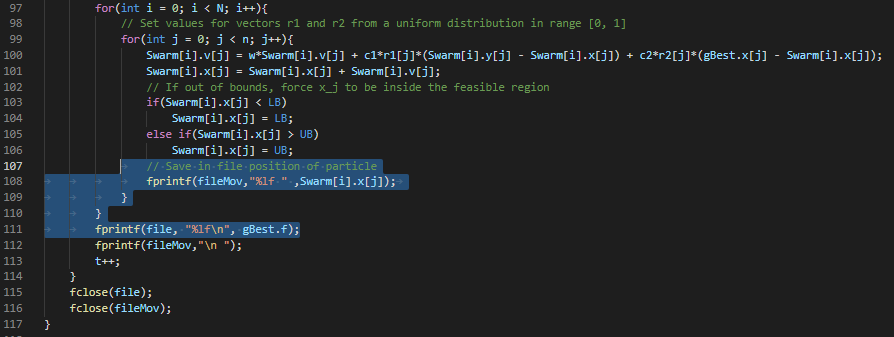
Mientras que los limites puestos en los DEFINE se les cambio el valor a -5 y 5.



1. Se creo un nuevo archivo llamado answer.txt en donde se guardó la posición de las partículas por cada iteración del algoritmo.

Para obtener resultados exitosos se usaron los siguientes parámetros al ejecutar el programa:

    ./PSO 2 100 100 3.5 2.3 0.9 1



1. Se genero una animación para mostrar el desplazamiento de las partículas en el espacio dado, debido a que por obvias razones la animación no la puedo mostrar en el reporte, esta se encontrara disponible en la carpeta donde se entrega el trabajo bajo el formato de MP4, adjunto fotografía de como se ve un frame de la animación: Diagrama

   Descripción generada automáticamente

**Parte 2: Ejercicio Manual de Naïves Bayes**

La tabla para hacer los operaciones fue obtenida de las instrucciones del reporte, en la parte de abajo se dejan los cálculos y pasos que seguir para llegar al resultado de la clasificación.

P(Beach = Yes) = 4/10

P(Beach = No) = 6/10

Outlook

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Yes | No | P(Yes) | P(No) |
| Sunny | 3 | 1 | ¾ | 1/6 |
| Rain | 0 | 3 | 0 | 3/6 |
| Cloudly | 1 | 2 | ¼ | 2/6 |
| Total | 4 | 6 | 100% | 100% |

Temp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Yes | No | P(Yes) | P(No) |
| High | 3 | 2 | ¾ | 2/6 |
| Mild | 1 | 2 | ¼ | 2/6 |
| Low | 0 | 2 | 0 | 2/6 |
| Total | 4 | 6 | 100% | 100% |

Humidity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Yes | No | P(Yes) | P(No) |
| High | 2 | 2 | 2/4 | 2/6 |
| Normal | 2 | 4 | 2/4 | 4/6 |
| Total | 4 | 6 | 100% | 100% |

P(X|C)P(C=Yes) = ¼ \* ¼ \* 2/4 \* 4/10 = 0.0125

(C=No) = 1/6 \* 2/6 \* 2/6 \* 6/10 = 0.0111

Se divide por ambos lados P(x) para normalizar

P(x) = 2/10 \* 3/10 \* 4/10 = 0.024

Se dividen los resultados

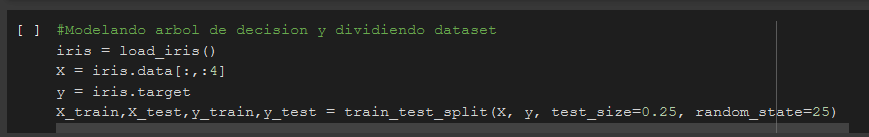
P(C=Yes|X) = 0.0125/0.024 = 0.5208

P(C=No|X) = 0.0111/0.024 = 0.4625

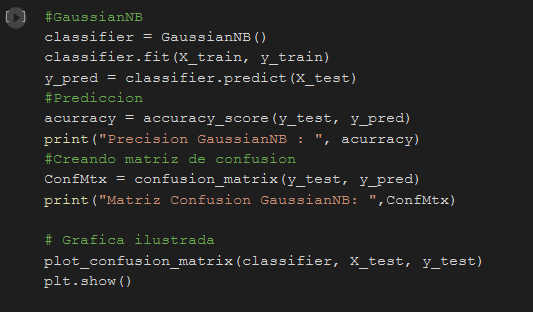
**R-. No, no se podrá ir a la playa**

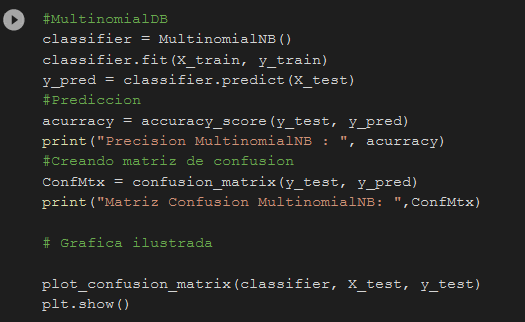
**Parte 3: Uso de Naïves Bayes con Scikit-Learn**

1-2) Se utilizo la base de datos de Iris para crear el programa, este dataset se importo de sklearn. Para la división del training set y testing set se utilizó la misma técnica de las otras prácticas, pero en este caso se usó un 75% para el training y un 25% para el test set, en el siguiente código se muestra como se hizo este paso:



3) Para hacer la clasificación se usaron dos diferentes funciones que maneja scikit learn, GaussianNB y Multinomial. Ambas se manejaron de la misma manera pues recibían los mismos parámetros, en el siguiente código se muestra como se implementaron y como se obtuvo la precisión del modelo usando ambas funciones y por último de obtuvo las matrices de confusión de cada caso:



NB 

1. Los resultados obtenidos se mostrarán en la parte inferior, pero se puedo concluir que la función de GaussianNB tiene mejor precisión que la multinomial, por lo tanto, para categorizar este dataset es mejor usar la función de GaussianNB.

