|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T3  **V2** | |  | | --- | | **Minería de Datos** | | #1 Mariana Aguilar Cevada | | #20 Iván Magdiel Texis Manzano | | 8°B | E# |

**Requerimientos:**

**\***Computadora con Windows 7 y el software Python 3.5 y Spyder instalados.

\*Carpeta con imágenes del 0 al 9 segmentadas.

**OCR**

OCR es un proceso dirigido a la digitalización de textos, los cuales identifican automáticamente a partir de una imagen símbolos o caracteres que pertenecen a un determinado alfabeto, para luego almacenarlos en forma de datos. Así podremos interactuar con estos mediante un programa de edición de texto o similar.

**Conjunto de Imágenes**

Las imágenes se encuentran de manera binaria, esto es una imagen digital que tiene únicamente dos valores posibles para cada pixel. Los colores utilizados para su representación son negro y blanco aunque puede usarse cualquier pareja de colores. Uno de los colores representa el fondo y el otro los objetos que aparecen en la imagen.

Conjunto: Se tiene un total de 14914 imágenes, distribuidas en 36 carpetas con 238 imágenes cada una.

Las siguientes imágenes muestran un ejemplo de las dimensiones de cada imagen contenida en las carpetas:

Carpeta 0: 

**Imagen 1.-** Dimensiones estándar de la imagen: 56x88 píxeles.



Carpeta 1**:**

**Imagen 2.-**Dimensiones estándar de la imagen: 32x86 píxeles.

Carpeta 8:



**Imagen 3.-** Dimensiones estándar de la imagen: 56x89 píxeles.

**DataSet**

Un DataSet representa un conjunto completo de datos, incluyendo las tablas que contienen, ordenan y restringen los datos, así como las relaciones entre las tablas.

El dataset utilizado en esta práctica se obtiene a partir de las siguientes características:

**Característica 1**

Se obtienen la razón de filas y columnas de toda la imagen:

Razón total = filas/columnas



FILAS

COLUMNAS

**Imagen 4.-** Imagen binaria que indica filas y columnas de la imagen.

**Característica 2**

Calcular el número de unos que existen en toda la imagen y este resultado se divide sobre el tamaño total de la imagen, es decir: #1’s / (#Columnas \* #filas)



**1**

**Imagen 5.-** Imagen binaria en donde los unos corresponden a l área que ocupa la imagen.

**Característica 3, 4 y 5**

Calcula cuantos unos existen en la fila que se encuentra a la mitad, a un cuarto y a tres cuartos de la imagen, y este resultado se divide sobre el número total de columnas.

**Característica 6, 7,8**

Calcula cuantos unos existen en la columna que se encuentra a la mitad, a un cuarto y a tres cuartos de la imagen, este resultado se divide sobre el número total de filas.

Filas:

Mitad

Un cuarto

Tres cuartos

Columnas:

Mitad

Un cuarto

Tres cuartos



1

**Imagen 6.-** Imagen binaria que indica las posiciones en las que se calculan el número de unos existentes en la imagen, características 3 a 8.

**Característica 9, 10,11**

Calcula el número de cortes, es decir: el número de cambios de cero a uno y uno a cero que existen en la imagen, en la mitad, un cuarto y tres cuartos de las filas de la misma.

**Característica 12, 13,14**

Calcula el número de cortes, es decir: el número de cambios de cero a uno y uno a cero que existen en la imagen, en la mitad, un cuarto y tres cuartos de las columnas de la misma.



0

1

Cortes

**Imagen 7.-** Imagen binaria que indica los cortes en las posiciones de la imagen 6.

\*Nota: La nueva instancia es una imagen segmentada y binarizada.

**Clasificación**

Se utilizó el método KNN para clasificar, que consiste en introducir el número de vecinos que se encuentren más cercanos a la nueva instancia, a partir de ellos se realiza una votación para saber que clase se repite más entre estos. La clase con mayor repetición será a la que pertenece la nueva instancia.

**Procedimiento:**

**Generación del dataset**:

Fase 1: Leer las carpetas

Fase 2: Por carpeta leer imágenes

Fase 3: Extraer las 14 características de cada una de las imágenes y se guardan en un archivo .csv con el nombre de: dataset.csv

**Clasificación:**

Fase 4: Pedir al usuario que seleccione una imagen que será la nueva instancia.

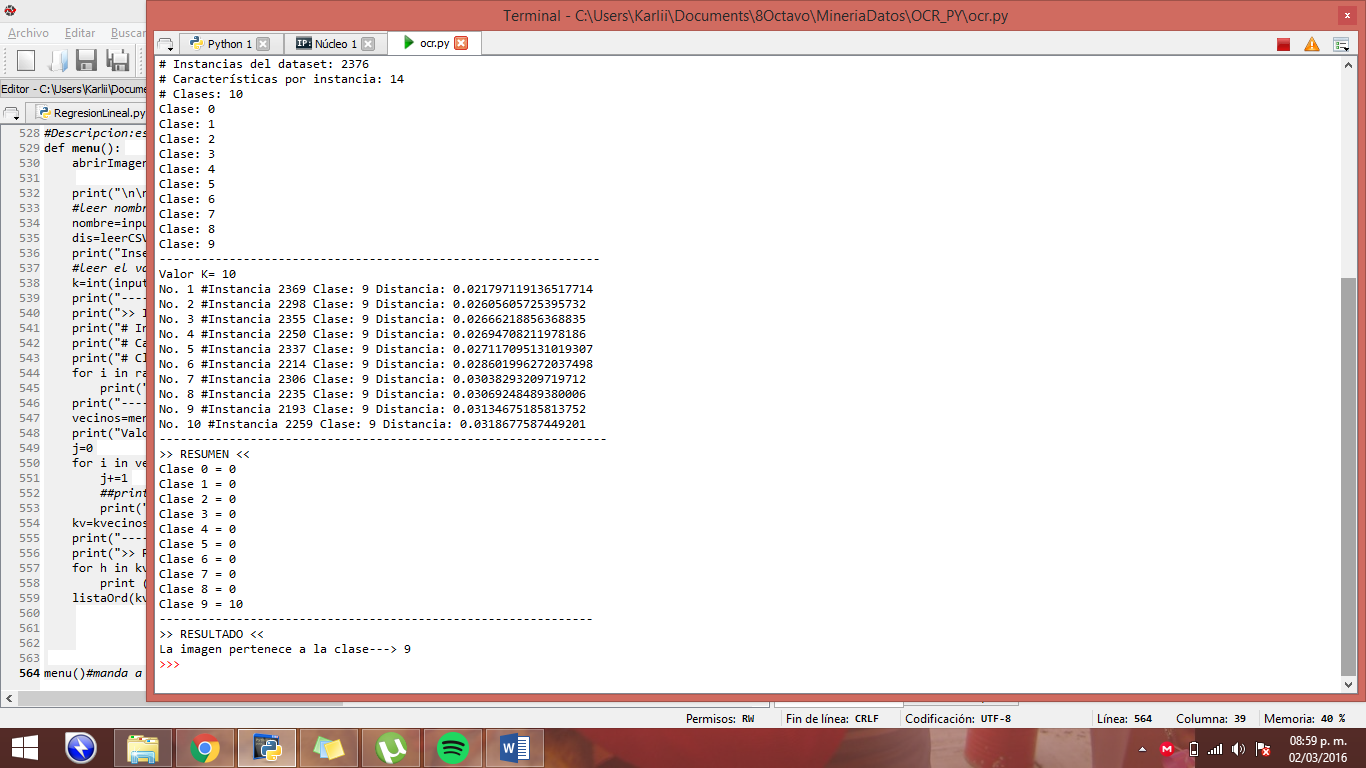
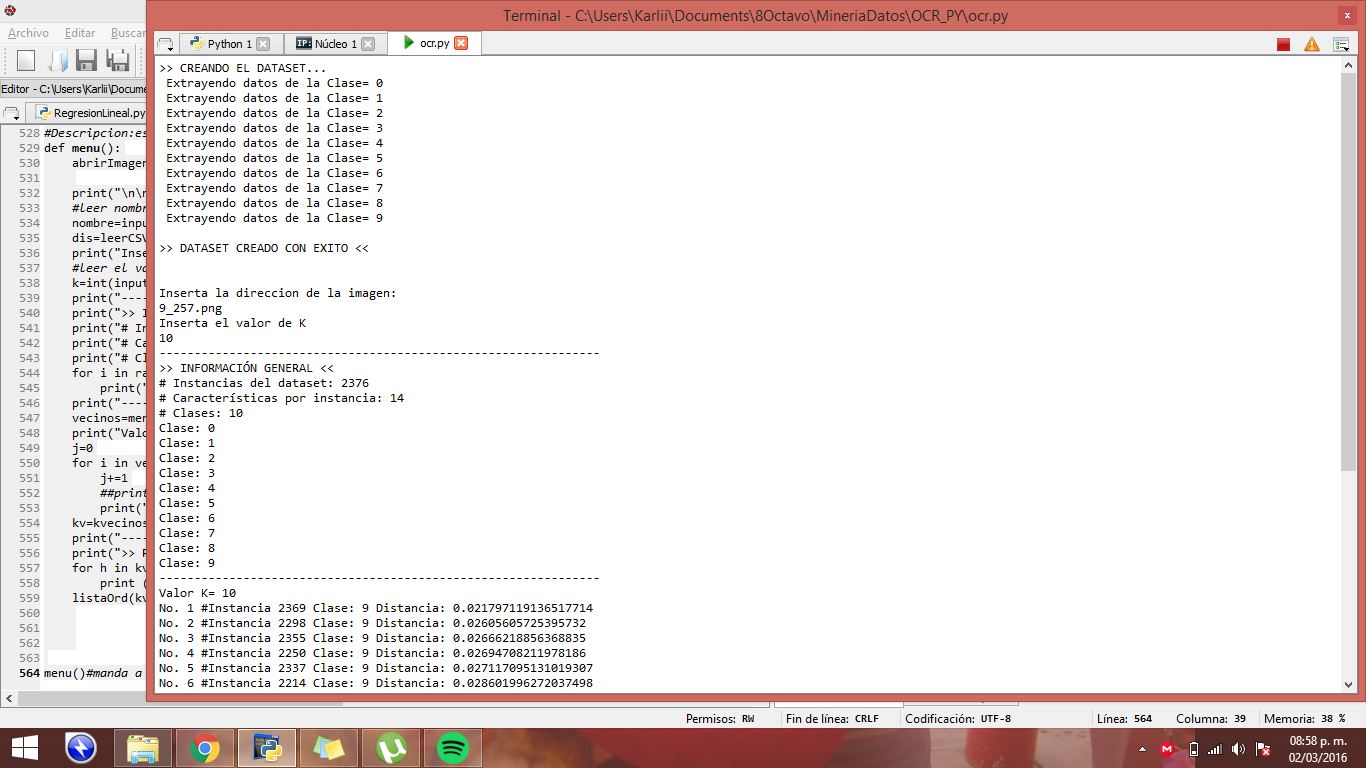
Fase 5: Se obtienen las 14 características de la nueva instancia.

Fase 6: Se pide al usuario el valor de K.

Fase 7: Se aplica KNN entre el dataset y la nueva instancia.

Fase 8: Se muestra en pantalla a que clase pertenece la nueva instancia.

**Resultados:**

Salida del programa

**Código disponible en Github**

**https://gifhub.com/KarlaHH/OCR\_PY.git**