Reporte Kmeans **Lenin Falconí**
Prof. Henry Paz Lab Date: 17/12/18

1. Dataset

Se dispone de un dataset de clientes de un Supermercado. La información disponible en el mismo es: género, edad, ingreso anual, y un score de la tienda. El dataset se puede descargar de:
https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python#Mall_Customers.csv

2. Tramiento del Dataset y programas

El género se sustituye por números: 1-¿male 2 -¿female. El resto de datos se mantiene igual. No fue necesario escalar o normalizar la data debido a que los datos se mentienen con valores homogéneos o similares y no existen datos con valores muy mayores o muy menores respecto a la mayoría.

El código desarrollado para este ejemplo se puede obtener de mi repositorio GIT: https://github.com/LeninGF/KmeansKnn

El script kmeans.py permite obtener la clusterización de la data. Genera las etiquetas, las mismas que se salvan en un archivo de csv denominado *label.csv*. Con estas etiquetas se conformará el dataset de entrenamiento. Este script también genera los datos artificiales para la etapa de prueba.

3. Etiquetado de Datos

Se utilizan como etiquetas los grupos generados por Kmeans utilizando dos centros –que fue lo que se concluyó del análisis de elbow realizado anteriormente–. Esto permite asignar a clase 0 y clase 1 cada ejemplo según Kmeans. El archivo que contiene esta data es el archivo textit-Mall_Customers_Class1.csv que es el archivo de entrenamiento con 200 ejemplos.

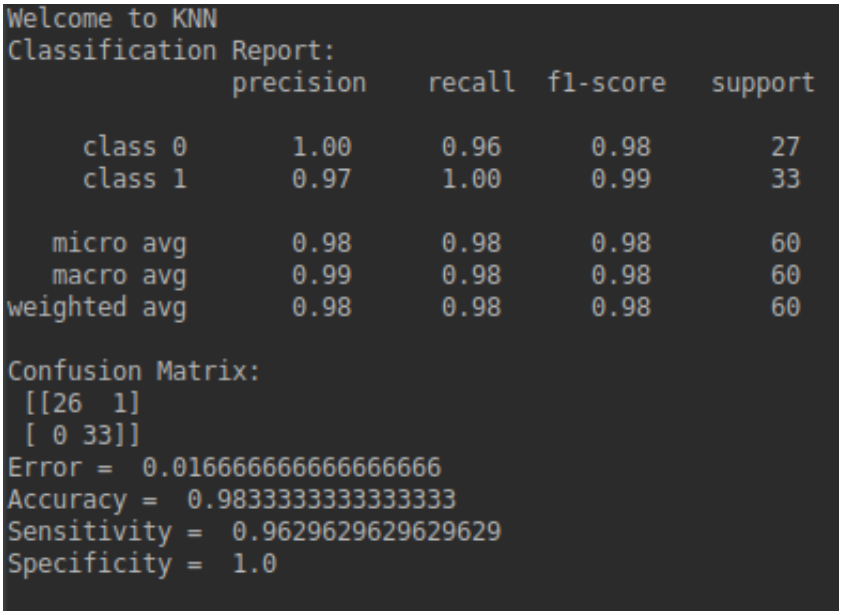

Figura 1: Resultados KNN

4. Entrenamiento KNN

El modelo de KNN se entrena con los 200 ejemplos según la etiqueta antes obtenida de Kmeans.

5. Testeo

Para evaluar el rendimiento de la clasificación de KNN se utilizó como métricas la matriz de confusión y otros valores como la precisión y el recall sobre un dataset de 60 datos generados artificialmente. Las etiquetas se obtuvieron utilizando la función de predicción del kmeans. Estos datos están en los archivos *xtest.csv, ytest.csv*

En la figura 1 se puede observar el resultado de clasificación de KNN en 2 clases '0' y '1'. El modelo obtenido con KNN tiene un accuracy de 98 % con un error del 1.6 %. Por tanto el modelo es capaz de predecir nuevas clases según los datos que ingresen. Esto también se confirma al disponer de una diagonal balanceada en la matriz de confusión.

5.1. Conclusiones

- La clasificación con KNN en 2 grupos confirma que los datos originales clusterizados con Kmeans se pueden separar en 2 grupos diferentes.

- La matriz de confusión indica que existe una clasificación suficientemente distinta entre Verdaderos Positivos y Verdaderos negativos.
- Para el presente ejemplo, KNN clasificó adecuadamente la data según las etiquetas generadas por Kmeans.
- Un bajo rendimiento en la clasificación puede indicar la necesidad de tratamiento de datos o el incremento de variables independientes que describan adecuadamente las observaciones.