Conociendo la base de datos

C. G. Rodríguez, *Estudiante MCIA, UAQ.*
Machine Learning

***Resumen*— En la actualidad, el análisis de datos es un proceso exhaustivo debido a la gran cantidad de datos que se genera día con día. Por lo cual, este proceso consiste en resaltar información útil para la toma de decisiones.**

El presente trabajo se enfoca en realizar un análisis de la base de datos propuesta por medio de la estadística, lo cual resulto en un mejor entendimiento de los datos en cuestión.

***Temas claves*— Estadística, Distribución de datos, Balance de clases.**

I. Introducción

En la actualidad, constantemente se generan grandes volúmenes de datos alrededor del mundo, los cuales son procesados a fin de obtener de ellos información sustancial. Sin embargo, este procesamiento cuenta con ciertos puntos a tener en consideración, como son: el espacio de almacenamiento, la velocidad en el procesamiento, la integración de los datos a una arquitectura estructurada, la veracidad de los datos, etc.

Por tal motivo, la ciencia de datos es un punto esencial para la obtención del máximo aprovechamiento de los datos, ya que por medio de esta disciplina se lleva a cabo el análisis de los datos recolectados con el propósito de extraer información valiosa que optimicen la toma de decisiones y con ello brinde soluciones a diversos problemas. Debido a esto, cada vez más sectores emplean este campo, como es el caso de las ciencias de la salud, los mercados financieros, las redes sociales, entre otros.

En el presente trabajo se hará uso del área de la estadística, en específico las medidas descriptivas, las cuales engloban las medidas de tendencia central, las medidas de posición, las medidas de dispersión; a fin de realizar la identificación de las diferentes características de la base de datos Iris. Así como también, se realizará una comparación de los resultados obtenidos por medio del uso de funciones y librerías pertenecientes al entorno de Python.

II. Objetivo

Introducir al alumno en temas relacionados con el análisis de conjuntos de datos, con el propósito de poder obtener una mejor interpretación de los elementos presentes en la base de datos propuesta.

III. Marco Teórico

A. Ciencia de Datos

La ciencia de datos es un enfoque multidisciplinario debido a que combina diversos campos, como son: las matemáticas, la estadística, la inteligencia artificial y la ingeniería de computación; para el análisis de grandes conjuntos de datos a fin de descubrir patrones entre sus elementos, lo cual proporciona un mejor entendimiento, logrando con ello poder dar respuesta ciertas preguntas: ¿Qué pasó?, ¿Por qué pasó?, ¿Qué pasará?, ¿Qué se puede hacer con los resultados?

Así bien, un componente clave dentro del campo de la ciencia de datos es la minería de datos, la cual es una actividad que se enfoca en tres puntos fundamentales: la extracción de los datos, descubrimiento de patrones y el soporte para el desarrollo de modelos.

Conforme a lo anterior, primeramente, es necesario llevar a cabo la selección de los datos, en donde se realiza la identificación de las posibles fuentes de información de acuerdo con la necesidad a resolver. Una vez identificadas las fuentes en cuestión, se establece el proceso de extracción y transformación de los datos, debido a que es usual contar con la presencia de ciertos inconvenientes que terminan afectando la calidad de los mismos, como son: Valores faltantes, los cuales pueden deberse a errores en su correspondiente integración; Problemas de cardinalidad, en donde se observa un valor inesperado o bien, un valor que no correspondiente con algún atributo en cuestión; y los valores atípicos (outliers), los cuales son valores que se encuentran muy alejados de la tendencia central.

Posteriormente a la detección de alguno de los problemas mencionados en el párrafo anterior es usual la implementación de técnicas de imputación de datos, a fin de poder contar con un conjunto de datos más completo, lo cual daría pie al siguiente punto, el cual se enfoca en la identificación de patrones.

Los patrones son descubiertos por medio del área de la estadística, en donde se suelen utilizar diversas herramientas como es el caso de las medidas descriptivas, las cuales permiten obtener un resumen informativo del contenido de la

C. G. Rodríguez Flores estudia la Maestría en Ciencias en Inteligencia Artificial en la Universidad Autónoma de Querétaro
(e-mail: gaby96.flores@gmail.com)

base de datos en cuestión, entre ellas se encuentran [1]:

- Medidas de Tendencia Central

Las medidas de tendencia central son utilizadas ampliamente debido a que por medio de ellas es posible resumir conjuntos de datos, los cuales posteriormente sometidos en un estudio estadístico. Entre las medidas más empleadas se encuentran:

La media, conocida también como promedio, es la medida más común entre las medidas de tendencia central. Se calcula a través de la sumatoria de un conjunto de datos entre el número total de datos. Sin embargo, el mayor problema de esta medida se debe a su alta sensibilidad al contener valores atípicos (outliers) en la base de datos.

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{1}$$

La mediana se enfoca en representar la posición central de la variable que separa la mitad inferior y la mitad superior del conjunto de datos.

$$\text{median}(x) = \begin{cases} x_{r+1} & \text{Si } m \text{ es impar con } m = 2r+1 \\ \frac{1}{2}(x_r + x_{r+1}) & \text{Si } m \text{ es par con } m = 2r \end{cases} \tag{2}$$

La moda, es la medida que indica el valor que más se suele repetir dentro de una base de datos.

En la siguiente figura, se puede observar cada una de las medidas mencionadas anteriormente por medio de la utilización de una campana de Gauss, en donde la acumulación más alta de datos se encuentra en los valores intermedios.

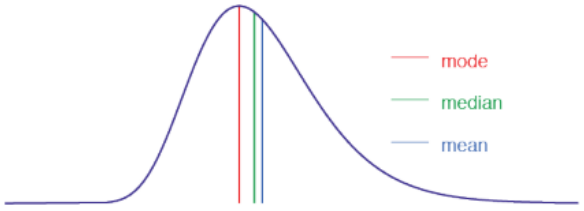


Fig.1. Medidas de tendencia central.

- Medidas de Dispersión

Estas medidas brindan información sobre que tan distintas o similares tienden a ser las observaciones con respecto a un valor en particular, el cual generalmente se refiere a alguna medida de tendencia central.

Entre las medidas de dispersión se encuentran:

- Varianza, en donde se hace referencia a la dispersión de los valores en un conjunto de datos determinado, en otras palabras, es el promedio del cuadrado de las distancias entre cada observación y la media del conjunto de observaciones.

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1} \tag{3}$$

- Desviación estándar indica que tan dispersos están los datos con respecto a la media., se puede determinar por medio de la raíz cuadrada de la varianza. Así bien, esta medida de dispersión es también sensible a los valores atípicos (outliers).

$$s = \sqrt{s^2} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \tag{4}$$

- Covarianza, la cual se enfoca en los cambios que se asocian entre una variable y otra, es decir, la variación lineal entre un par de variables. Sin embargo, se debe de tener cuidado en su implementación si no se ha realizado previamente una inspección de los datos.

$$Cov(X,Y) = \frac{\sum_{1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n} \tag{5}$$

- Medidas de posición

Las medidas de posición se enfocan en dividir el conjunto de datos en partes iguales. Entre ellas se encuentran los cuartiles, los cuales se encargan de dividir un conjunto de datos ordenados en cuatro intervalos.

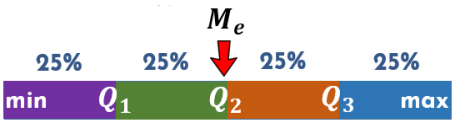


Fig.2. Distribución de Cuartiles.

Por otra parte, una parte esencial para el análisis de la distribución de los conjuntos de datos se lleva a cabo por medio de su representación visual, en donde como su nombre lo indica es posible realizar la implementación de una variedad de gráficos para la identificación de las relaciones entre los elementos, ya que por medio de ellas se tiene un panorama más perceptible.

Entre las diversas representaciones gráficas se encuentran:

- Histograma, el cual permite la visualización de la distribución de los valores de una variable, en donde sus valores se dividen en contenedores

(bins), los cuales dependiendo del número de elementos contenidos en cada contenedor variara la altura de cada barra en cuestión. Así bien, este grafico puede mostrar diferentes distribuciones: Uniforme, Normal (Unimodal), Unimodal sesgado a la izquierda, Unimodal sesgado a la derecha, Multimodal y Exponencial.

- Densidad, otra forma de visualizar como se distribuyen los datos es estimando la densidad. La densidad es una versión más suavizada del histograma.
- Boxplots o diagramas de caja, entrega información sobre la simetría de la distribución de los datos, es decir, si la mediana no se encuentra en el centro del rectángulo, la distribución no es simétrica. Este tipo de representación permite determinar la presencia de valores atípicos (outliers) a partir del rango inter-cuartil (IQR), el cual es la diferencia entre el tercer y el primer cuartil (Q3 − Q1).
- Diagramas de dispersión, usan coordenadas cartesianas para mostrar los valores de dos variables del mismo largo. Los valores de los atributos determinan la posición de los elementos.

B. Base de datos Iris

El conjunto de datos Iris fue introducido por el británico Ronald Fisher en su artículo “El uso de múltiples mediciones en problemas de taxonómicos” a fin de determinar la variación morfológica de la flor Iris. La recolección de los datos fue realizada en la Península de Gaspé, obtenidas en la misma pradera el mismo día a la misma hora.

En la siguiente figura se muestran las tres especies de Iris, mencionadas con anterioridad a fin de que el lector tenga un conocimiento previo sobre la apariencia de las clases a tratar en este trabajo.

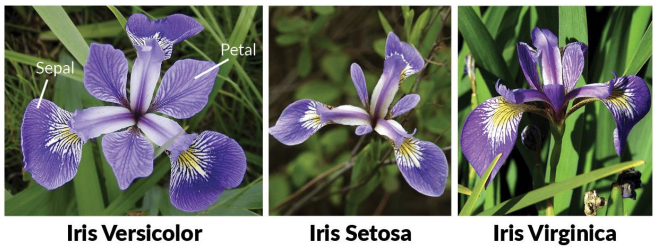


Fig.3. Especies de la flor Iris.

Iris Versicolor

El iris versicolor, conocido de igual manera como arlequín o bandera azul; es una especie de planta perteneciente a la familia Iridaceae. Este tipo de flor es originaria de Norteamérica, así bien, crece en lugares encharcados y pantanosos

Con respecto a su aspecto físico, alcanza los 100 cm de altura con ramas, sus hojas son grandes de aproximadamente 30 cm de longitud con forma de espada recta y estriada. Mientras que los sépalos tienen una forma espatulada o lanceolado. Su fruto corresponde cápsulas con numerosas semillas planas [2].

Además, este tipo de planta cuenta con ciertos beneficios para el ser humano en el ámbito de la salud, como son [3]:

- El aumento en la producción de orina y bilis
- Actuar como un laxante suave
- El rizoma fresco puede destilarse con agua y usarse en pequeñas dosis para tratar diversas afecciones hepáticas como la hepatitis.
- Las hojas secas o la raíz pueden ser utilizadas en té, el cual ha sido utilizado como tratamiento para dolencias digestivas y afecciones relacionadas con el abuso de drogas y alcohol, o exposición excesiva a productos químicos y contaminantes industriales.
- Es un buen estimulante para el sistema circulatorio y linfático.
- Tratamiento ideal para una variedad de enfermedades de la piel, ya que puede ser usado para aliviar el dolor y la inflamación asociados con diversas llagas, quemaduras y hematomas en la piel.
- Tratamiento de dolores de cabeza o migrañas, sífilis y afecciones reumáticas.

Iris Setosa

El Iris Setosa cuenta con varios nombres comunes: Wild flag iris, Alaska iris, Arctic Iris incluido 'Beachhead Iris' debido a que tolera el aire salado o las condiciones marítimas. Por lo cual, este tipo de especie tolera muchos tipos de hábitats, ya que puede encontrarse en ciénagas, prados, junto a ríos, a orillas de lagos, playas, dunas, promontorios y bosques claros. Debido a la variedad de hábitats, este tipo de especie es considerada una de las más resistentes.

Con respecto, al ámbito de la salud cada una de las partes que compone al Iris setosa son consideradas venenosas. Un ejemplo de ello es el rizoma, el cual contiene iridina, sustancia puede afectar el hígado y los órganos digestivos. Así bien, esta flor puede causar reacciones alérgicas como erupciones cutáneas graves, como de igual manera, causar vómitos o diarrea.

Aunque la planta es venenosa, sus raíces ricas en almidón pueden ser utilizadas para el consumo humano mediante la cocción. Los aleutianos hacían una bebida de las raíces, la cual servía como laxante. También puede ser utilizada para hacer una tintura, la cual en pequeñas cantidades puede ayudar a calmar la inflamación linfática. Además, puede combinarse con árnica para aliviar los moretones [4].

Iris Virginica

Iris virginica es una especie perenne de planta con flores, es común encontrarla a lo largo de la llanura costera desde Florida hasta Georgia en el sureste de los Estados Unidos.

Los aspectos físicos que caracterizan a esta especie es que cuenta con 2 a 4 hojas erectas o arqueadas, de color verde brillante, las cuales tienen forma de lanza aplanada. Estas hojas miden alrededor de 1 a 3 cm de ancho y en ocasiones tiene un tamaño más alargado que el tallo de la flor.

En el ámbito médico, la raíz de esta flor ha sido utilizada para diferentes propósitos, como son: ungüento para la piel, la infusión hecha de la raíz puede servir para tratar dolencias en el hígado, una decocción de la raíz se usa para tratar la orina de color amarillento, etc. [5]

IV. Materiales y Métodos

MATERIALES

Equipo:	Herramienta:
Laptop Dell g15 Procesador: Intel® Core™ i7-10870H de 10.ª generación	Plataforma de Google Colab

Base de datos:

Esta base de datos contiene tres especies de Iris (Virginica, Versicolor y Setosa), las cuales contienen 50 muestras de cada una. Cada una de las muestras fue medida de acuerdo con cuatro características: ancho y largo de los pétalos y sépalos.

Así bien, este repositorio suele utilizarse ampliamente en ejemplos de minería de datos, clasificación y agrupamiento, por lo cual, es posible encontrarlo en el repositorio de aprendizaje automático UCI Machine Learning[6], Kaggle[7], como también de Scikit Learn[8].

MÉTODOS

Para la realización de esta tarea, fue necesario primeramente importar la librería de Pandas, así como el montaje de Google Drive en el entorno de ejecución a fin de poder hacer uso de la base de datos propuesta.

De igual manera, se requirió importar las librerías de matplotlib y seaborn, las cuales se utilizaron para realizar la parte de la visualización de los datos.

Cada uno de los puntos solicitados para esta actividad fueron abordados por medio del desarrollo de diferentes funciones, como es el caso de la figura 4, en donde se puede visualizar cada una de las líneas que componen la función propuesta. Esta función se encarga de brindar ciertos datos generales como son: número total de atributos, nombre de cada atributo junto con su correspondiente contenido, las observaciones de cada uno de atributos, el número total de instancias por cada atributo, el tipo de atributo, y el porcentaje de valores faltantes de cada columna.

```
[ ]  def analisis_general(dataset):
       print('Número de Atributos:' + str(dataset.shape[1]))
       print('')
       for column in dataset:
         print('Nombre de la columna: ', column)
         print('Contenido de la columna: ', dataset[column].values)
         print('')
         #Observaciones de cada Atributo
         Lists = [] #Creación de una lista
         Tipo = []
         for List in dataset[column].values: #Recorrer los valores de la columna en cuestión
           if List not in Lists: #Si el valor no se encuentra en la lista
             Lists.append(List) #Agregar a la lista ese valor
         print('Observaciones de los Atributos:',Lists)
         print('')
       #Número de Instancias
       print('Número de Instancias por Atributo:' + str(dataset.shape[0]))
       print('')
       print('Tipo de Atributo')
       print(dataset.dtypes)
       #Valores Faltantes (Porcentaje)
       print(' ----------------------------------------------------------------------')
       print('')
       Porcentaje_NaN = ((dataset.isnull().sum() / len(dataset))*100).sort_values(ascending = False)
       print('Porcentaje de Valores Faltantes')
       print(Porcentaje_NaN)
       print('')
```

Fig.4. Función Análisis General.

Así bien, en la figura 5 se muestra la función enfocada a la obtención de los valores minimos y maximos de cada columna, en donde fue necesario implementar dos ciclos for anidados a fin de poder recorrer cada una de las instancias de cada columna, con lo cual los valores perteneciente a cada instancia fue fueron añadiendo en una lista con el proposito de ir comparando cada uno de los valores de la lista y con ello poder determinar cuales son los valores maximos y minimos de cada columna.

```
[ ]  def sta_atributos(dataset):
       print(' ----------------------------------------------------------------------')
       print('')
       columnas= []
       #Estadisticas de sus atributos
       for column in dataset:
         if dataset[column].dtypes == 'float64'or dataset[column].dtypes == 'int64':#Considerar unicamente las columnas sin valores de tipo string
           lista = []
           for fila in dataset[column]:
             lista.append(fila)
           max = lista[0] #Asignarle el primer valor de la lista a la variable max
           min = lista[0] #Asignarle el primer valor de la lista a la variable min
           for i in lista: #Recorrer la lista
             if i > max: #Si el valor de i es mayor que el valor que se encuentra en la variable max
               max= i #Asignarle el nuevo valor a la variable max
             if i < min:#Si el valor de i es menor que el valor que se encuentra en la variable min
               min = i #Asignarle el nuevo valor a la variable min
           columnas.append(column) #Guardar en la lista de columnas el nombre de la columna
           print('Columna ',column)
           print('Valor máximo: ',max)
           print('Valor mínimo: ',min)

           sum_n = sum(dataset[column]) #sumar el conjunto de valores de la columna en cuestión
           len_n = len(dataset[column].values) #Obtener el numero total de valores de la columna en cuestión
           promedio = (sum_n/len_n) #Obtener el promedio de la columna en cuestión
           print('Valor promedio: ', promedio)
           # Varianza
           x = 0
           for i in lista:
             x = x + (pow((i - promedio),2)) #Suma acumulativa del cuadrado de la diferencia entre el valor de la posicion de la lista y el promedio de la columna
           varianza = x / (len_n-1) #Formula de la Varianza
           std = pow(varianza,0.5) #Formula para la obtención de la Desviación Estandar
           #print('Varianza: ', varianza)
           print('Desviación estandar: ',std)
           print('')
       print(' ----------------------------------------------------------------------')
       #print('')
       cov(dataset,columnas)
       print(' ----------------------------------------------------------------------')
       return columnas
```

Fig.5. Primera parte de la función de Estadística de Atributos.

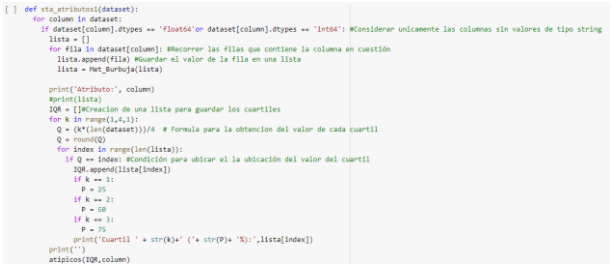
Asimismo, tambien esta función se encarga de obtener los valores promedios y la desviación estandar de cada columna. Ademas, en las ultimas lineas de la función se encuentra la llamada a otro función, la cual se encarga de realizar el calculo de la covarianza. En esta función, se le realiza una copia a la base de datos propuesta ya que por medio del condicional if se localizan las columnas con valores categorios, con el proposito de eliminarlas.

```
def cov(dataset,columnas):
  y=z=0
  copy = dataset.copy()
  for column in copy: #Recorrer la columnas de la copia del dataset
    for col in columnas: #Recorrer las valores de la lista de columnas
        if column != col: #Si los valores de las columnas entre la copia del dataset y la lista de colu
          copy.drop(col, axis=1) #Eliminar de la copia del dataset la columna en cuestión

  for index in range(copy.shape[1]): #Recorrer las posiciones de las columnas de la copia del dataset
    if copy.iloc[:,index].dtypes == 'float64'or copy.iloc[:,index].dtypes == 'int64' :#Considerar unica
        index1= index + 1
        y = (copy.iloc[:,index] - ((sum(copy.iloc[:,index]))/len(copy)))#Diferencia entre el valor de z
        for index1 in range(copy.shape[1]): #Recorrer las posiciones de las columnas de la copia del da
          if copy.iloc[:,index1].dtypes == 'float64'or copy.iloc[:,index].dtypes == 'int64': #Considera
            z = (copy.iloc[:,index1]- ((sum(copy.iloc[:,index1]))/len(copy))) #Diferencia entre el valo
            #print(copy.iloc[:, index1].values)
            #print(index, index1)
            c = sum(y*z)/len(copy) # Formula para la covarianza
            print('Covarianza de la columna',columnas[index], 'y la columna', columnas[index1],':', c)
  print('')
```

Fig.6. Función Covarianza.

En la figura 7, se muestra la segunda función encargada de obtener los cuartiles de cada columna, en donde se realiza el mismo procedimiento para considerar unicamente las columnas con valores numericos. De igual manera, se añaden en una lista los valores de los cuartiles a fin de poder utilizar dicha lista en la función ‘atipicos’, la cual se encuentra dentro de la función ‘sta_atrubutos1’.



```
[ ] def sta_atributos1(dataset):
      for column in dataset:
        if dataset[column].dtypes == 'float64'or dataset[column].dtypes == 'int64': #Considerar unicamente las columnas sin valores de tipo string
          lista = []
          for fila in dataset[column]: #Recorrer las filas que contiene la columna en cuestión
            lista.append(fila) #Guardar el valor de la fila en una lista
            lista = Met_Burbuja(lista)

          print('Atributo:', column)
          #print(lista)
          IQR = []#Creacion de una lista para guardar los cuartiles
          for k in range(1,4,1):
            Q = (k*(len(dataset)))/4  # Formula para la obtencion del valor de cada cuartil
            Q = round(Q)
            for index in range(len(lista)):
              if Q == index: #Condición para ubicar el la ubicación del valor del cuartil
                IQR.append(lista[index])
                if k == 1:
                  P = 25
                if k == 2:
                  P = 50
                if k == 3:
                  P = 75
                print('Cuartil ' + str(k)+' ('+ str(P)+ '%):',lista[index])
          print('')
          atipicos(IQR,column)
```

Fig.7. Función Estadística de los Atributos.

La función ‘atipicos’ se encarga de obtener el rango intercuartil de cada columna por medio de la diferencia obtenida entre el cuartil 3 con respecto al cuartil 1. Esto con el fin de encontrar los limites superiores e inferiores y poder determinar si la columna en cuestión cuenta con valores atipicos.

```
[ ] def atipicos(IQR,column):
        iqr = IQR[-1] - IQR[0] #Diferencia Q3 - Q1
        print('Rango Intercuartil(IQR):',iqr)
        lim_superior = IQR[-1] + (1.5*iqr)
        lim_inferior = IQR[0] - (1.5*iqr)
        print('Limite Superior:',lim_superior, 'y Limite Inferior:', lim_inferior)
        Atipico=[]
        for fila in dataset[column]:
          if fila>lim_superior or fila<lim_inferior:
            Atipico.append(fila)
        #print('Datos Atipicos en la columna: ' +column)
        if len(Atipico) == 0:
          print('No hay Datos Atipicos')
          print('')
        else:
          print('Datos Atipicos:',Atipico)
          print('')
```

Fig.8. Función Valores Atípicos.

En la figura 9, se muestra la función ‘clasif’, la cual se enfoca en identificar cada una de las clases pertenecientes de la columna seleccionada con el fin de proporcionar el número total de muestras de cada clase y por ende saber si el conjunto de datos se encuentra balanceado.

```
[ ] def clasif(Col_clasif):
      print(' ----------------------------------------------------------------------')
      Clasif = []
      for F in dataset[Col_clasif]:
        if F not in Clasif: #Si el valor no se encuentra en la lista
          Clasif.append(F) #Agregar a la lista ese valor
      print('Clases:', Clasif)
      print('')
      for C in range(len(Clasif)):
        print('Clase', Clasif[C],':')
        count = []
        for F in dataset[Col_clasif]:
          if Clasif[C] == F :
            count.append(F)
        print(len(count))
```

Fig.9. Función de Clasificación del Balance de Clases.

Cada una de las funciones mencionadas anteriormente será llamada por medio de la función principal, ya que lo que se pretende es poder visualizar los resultados de una manera ordenada. Para esta función unicamente se necesito tener como entrada tanto la base de datos propuesta, asi como la columna con valores categoricos seleccionada por el usuario. Asi bien, esta función regresa las columnas con valores numericos, ya que para la parte de la comprobación se requiere de ella.

```
[ ] def FG(dataset,Col_clasif):
      analisis_general(dataset)
      columnas = sta_atributos(dataset)
      sta_atributos1(dataset)
      clasif(Col_clasif)
      return columnas
```

Fig.10. Función General.

Diagrama de Flujo

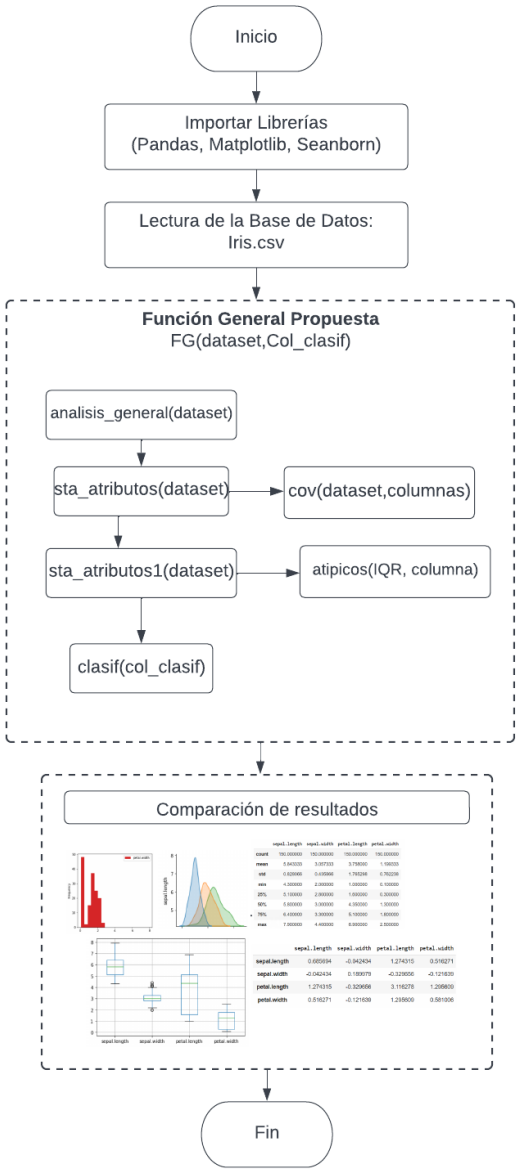


Fig.11.Diagrama de Flujo.

V. Resultados

En las siguientes tablas se puede visualizar los resultados obtenidos al realizar el correspondiente análisis de la base de datos propuesta, en donde es posible observar en la Tabla 1 la comprobación de la información referente a la cantidad de atributos e instancias del repositorio.

TABLA 1
NÚMERO DE ATRIBUTOS E INSTANCIAS DE LA BASE DE DATOS.

Base de datos	Núm. Atributos	Núm. Instancias por Atributo
Iris	5	150

Por medio de la Tabla 2, es posible visualizar que cuatro de las cinco columnas corresponden a un tipo de atributo numérico de tipo flotante, mientras que únicamente la última columna corresponde a un tipo de atributo categórico. De igual manera, es posible visualizar las diferentes observaciones de cada uno de los atributos.

TABLA 2
CARACTERÍSTICAS DE LOS ATRIBUTOS

Atributo	Tipo de Atributo	Observaciones de Atributos
sepal.length	float64	[5.1, 4.9, 4.7, 4.6, 5.0, 5.4, 4.4, 4.8, 4.3, 5.8, 5.7, 5.2, 5.5, 4.5, 5.3, 7.0, 6.4, 6.9, 6.5, 6.3, 6.6, 5.9, 6.0, 6.1, 5.6, 6.7, 6.2, 6.8, 7.1, 7.6, 7.3, 7.2, 7.7, 7.4, 7.9]
sepal.width	float64	[3.5, 3.0, 3.2, 3.1, 3.6, 3.9, 3.4, 2.9, 3.7, 4.0, 4.4, 3.8, 3.3, 4.1, 4.2, 2.3, 2.8, 2.4, 2.7, 2.0, 2.2, 2.5, 2.6]
petal.length	float64	[1.4, 1.3, 1.5, 1.7, 1.6, 1.1, 1.2, 1.0, 1.9, 4.7, 4.5, 4.9, 4.0, 4.6, 3.3, 3.9, 3.5, 4.2, 3.6, 4.4, 4.1, 4.8, 4.3, 5.0, 3.8, 3.7, 5.1, 3.0, 6.0, 5.9, 5.6, 5.8, 6.6, 6.3, 6.1, 5.3, 5.5, 6.7, 6.9, 5.7, 6.4, 5.4, 5.2]
petal.width	float64	[0.2, 0.4, 0.3, 0.1, 0.5, 0.6, 1.4, 1.5, 1.3, 1.6, 1.0, 1.1, 1.8, 1.2, 1.7, 2.5, 1.9, 2.1, 2.2, 2.0, 2.4, 2.3]
variety	object	['Setosa', 'Versicolor', 'Virginica']

Debido a que la base de datos propuesta es usualmente utilizada para un primer acercamiento de temas relacionados con el área de aprendizaje máquina no cuenta con valores faltantes, por lo cual en la siguiente tabla se puede apreciar que el porcentaje de valores faltantes de cada una de las columnas es cero.

TABLA 3
PORCENTAJE DE VALORES FALTANTES EN LA BASE DE DATOS.

Atributo	Porcentaje (%)
sepal.length	0.0
sepal.width	0.0
petal.length	0.0
petal.width	0.0
variety	0.0

En la Tabla 4, se muestran los valores máximos, valores mínimos, el promedio y la desviación estándar de cada una de las cuatro columnas numéricas correspondientes al ancho y largo del pétalo, así como el ancho y largo del sépalo.

TABLA 4
MEDIDAS DESCRIPTIVAS DE LA BASE DE DATOS.

Atributo	Valor Máximo	Valor Mínimo	Promedio	Desv. Estándar
sepal.length	7.9	4.3	5.84333	0.82806
sepal.width	4.4	2.0	3.05733	0.43586
petal.length	6.9	1.0	3.75800	1.76529
petal.width	2.5	0.1	1.19933	0.76223

La Tabla 5 presenta los resultados obtenidos al implementar la fórmula de la covarianza entre los cuatro atributos, en donde es posible observar que columnas cuenta con una mayor relación entre sí, ya que su covarianza será diferente de cero.

TABLA 5
COVARIANZA DE LOS ATRIBUTOS

Atributo	sepal.length	sepal.width	petal.length	petal.width
sepal.length	0.68112	-0.04215	1.26581	0.51282
sepal.width	-0.04215	0.18871	-0.32745	-0.12082
petal.length	1.26581	-0.32745	3.09550	1.28697
petal.width	0.51282	-0.12082	1.28697	0.57713

Mientras que la Tabla 6 permite visualizar cada uno de los cuartiles obtenidos al implementar la función en la función general para la obtención de los valores de los cuartiles.

TABLA 6
CUARTILES DE LOS ATRIBUTOS

Q	sepal.length	sepal.width	petal.length	petal.width
Q1	5.1	2.8	1.6	0.3
Q2	5.8	3.0	4.4	1.3
Q3	6.4	3.3	5.1	1.8

En la Tabla 7, muestran los valores atípicos de cada una de las cuatro columnas, en donde únicamente la columna correspondiente al largo del pétalo contiene cuatro valores atípicos.

TABLA 7
DATOS ATÍPICOS DE LOS ATRIBUTOS

Atributo	Datos Atípicos
sepal.length	-
sepal.width	-
petal.length	[4.4, 4.1, 4.2, 2.0]
petal.width	-

Por último, la Tabla 8 contiene la cantidad de muestras correspondiente a cada una de las clases del conjunto de datos Iris, lo cual corresponde con lo mencionado anteriormente en la breve descripción de la base de datos.

TABLA 8
BALANCE DE CLASES DE LA COLUMNA 'VARIETY'

Clases	Balance de Clases
Setosa	50
Versicolor	50
Virginica	50

Análisis de resultados

Primeramente, para la realización de la comprobación de los datos obtenidos en la sección de resultados se requirió de la utilización de funciones propias del entorno de Python, como es el caso de describe(), el cual proporciona una tabla con los valores máximos y mínimos, el valor promedio, el número de instancia de cada atributo, los cuartiles y la desviación estándar; y cov(), en donde brinda los valores correspondientes de cada una de las covarianzas.

	sepal.length	sepal.width	petal.length	petal.width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Fig.12. Visualización de los valores obtenidos por medio de la función describe.

	sepal.length	sepal.width	petal.length	petal.width
sepal.length	0.685694	-0.042434	1.274315	0.516271
sepal.width	-0.042434	0.189979	-0.329656	-0.121639
petal.length	1.274315	-0.329656	3.116278	1.295609
petal.width	0.516271	-0.121639	1.295609	0.581006

Fig.13. Visualización de los valores obtenidos por medio de la función cov.

La tabla 9, muestra la comparación de los resultados obtenidos conforme a la función propuesta y la función de Python, en donde se observar que existe una pequeña variación de los valores obtenidos.

TABLA 9
COMPARATIVA DE LA COVARIANZA OBTENIDA POR MEDIO DE LA FUNCIÓN PROPIA Y LA FUNCIÓN DE PYTHON.

Covarianza	Función propia	Función Python
sepal.length – sepal.length	0.68112	0.685694
sepal.length – sepal.width	-0.04215	-0.042434
sepal.length – petal. length	1.26581	1.274315
sepal.length – petal. width	0.51282	0.516271
sepal.width – sepal. width	0.18871	0.189979
sepal.width – petal. length	-0.32745	-0.329656
sepal.width – petal. width	-0.12082	-0.121639
petal. length – petal. length	3.09550	3.116278
petal. length – petal. width	1.28697	1.295609
petal. width – petal. width	0.57713	0.581006

Por medio de utilización de la librería de matplotlib se llevó a cabo la visualización de cada uno de los histogramas correspondientes a las columnas del largo y ancho de los pétalos, así como del largo y ancho de los sépalos, presentes en la figura 14.

Los histogramas correspondientes a las columnas de petal.length y petal.width muestran una distribución multimodal, debido a que se pueden ver dos o más valores de tendencia en la misma distribución. Sin embargo, se observa que el histograma de petal.width corresponde de igual manera a un histograma unimodal sesgado hacia la izquierda.

Así bien, con respecto a las columnas de sepal.length y sepal.width se puede observar que se trata de un histograma normal, ya que cuenta con una cierta simetría.

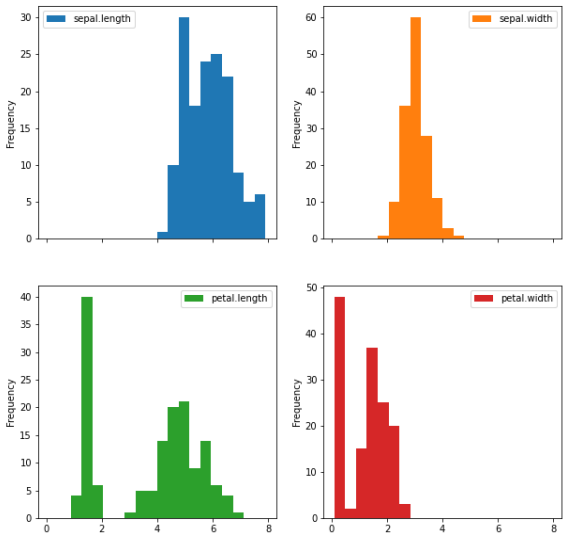


Fig.14. Visualización de la distribución de los histogramas.

Por otra parte, en la figura 15 se observan las gráficas de cajas de las cuatro columnas mencionadas anteriormente por medio de la utilización de la función boxplots.

Se puede observar que el atributo de sepal.width cuenta con ciertos valores atípicos, los cuales concuerda con el resultado obtenido en la función general propia.

Mientras que los atributos de sepal.length y sepal.width tienen una mediana casi centrada lo que puede llevar a suponer que estos dos atributos tienen una distribución normal de los datos. Así bien, en el caso de petal.width, su mediana se encuentra tendiendo hacia la derecha, mientras que el atributo de petal.width cuenta con una tendencia hacia la izquierda.

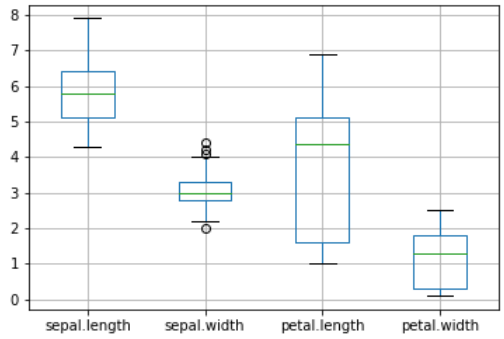


Fig.15. Visualización de las diversas gráficas de caja del conjunto de datos.

En la figura 16, se pueden observar las combinaciones de los atributos dados en forma de dispersión a fin de conocer las relaciones entre ellas, y por ende se contará con un mejor análisis de las clases en cuestión. Así bien, las gráficas de densidad permiten conocer de manera visual la superposición de las clases, como se observa los atributos ‘sepal.length’ y ’sepal.width’.

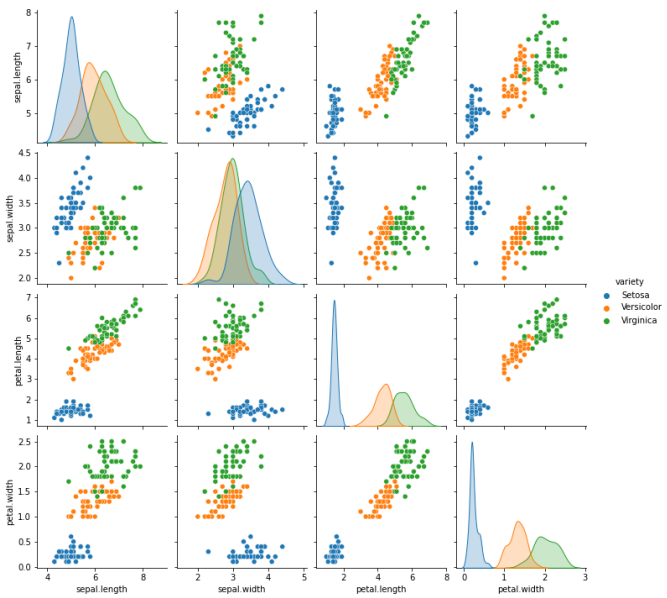


Fig.16. Gráficas de dispersión y gráficas de densidades de las clases: Setosa, Versicolor y Virginica.

VI. Conclusiones

La realización de esta tarea se enfocó en el análisis de los atributos comprendidos en el conjunto de datos propuesto, en donde fue necesario de la utilización del área de la estadística, en específico de las medidas descriptivas y de la representación de los datos por medio de gráficas; y del área de la programación para el desarrollo de una función general enfocada en brindar información relevante sobre la base datos en cuestión a fin de poder obtener una mejor compresión de sus elementos.

La función general se conformó por funciones propias a fin de realizar una comparación de los resultados obtenidos con respecto a los resultados brindados por las funciones de Python, en donde se observaron resultados similares entre sí.

Sin embargo, la realización de un código general para el análisis de datos trae consigo ciertos retos, ya que se deberán contemplar diferentes escenarios debido a que cada base de datos es diferente entre sí, es decir, puede contener elementos que terminen afectando su calidad, como son los valores faltantes, valores atípicos, un desbalance de clases, etc.

VII. Referencias

[1] Bravo Márquez, F. (2013) Análisis Exploratorio de Datos en R. Recuperado el día 23 de Agosto de 2022 de https://www.cs.waikato.ac.nz/~fbravoma/teaching/explora.pdf

[2] Urbipedia(s.f) Iris versicolor. Recuperado el día 23 de Agosto de 2022 de https://www.urbipedia.org/hoja/Iris_versicolor

[3] Netinbag(s.f)¿Qué es Iris Versicolor? Recuperado el día 23 de Agosto de 2022 de https://www.netinbag.com/es/medicine/what-is-iris-versicolor.html

[4] Naturalista(s.f.) Iris setosa. Recuperado el día 23 de Agosto de 2022 de https://www.naturalista.mx/taxa/164134-Iris-setosa

[5] Naturalista. (s.f) Iris Virginica. Recuperado el día 23 de Agosto de 2022 de https://www.naturalista.mx/taxa/117444-Iris-virginica

[6] UCI Machine Learning (s.f) Iris Data Set. Recuperado el día 19 de Agosto de 2022 de http://archive.ics.uci.edu/ml/datasets/Iris

[7] UCI Machine Learning (2016) Iris Species. Recuperado el día 19 de Agosto de 2022 de https://www.kaggle.com/datasets/uciml/iris

[8] Scikit-learn(s.f) The Iris Dataset. Recuperado el día 19 de Agosto de 2022 de https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html

[9] GeeksforGeeks(2019) Exploración de diagramas de caja e histogramas en datos de Iris. Recuperado el día 20 de Agosto de 2022 de https://www.geeksforgeeks.org/box-plot-and-histogram-exploration-on-iris-data/