Técnicas de Imputación y Normalización.

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

***Resumen*— Antes de poder utilizar la información recolectada en modelos de inteligencia artificial es necesario explorar cualquier problema de calidad presente en el conjunto de datos en cuestión, es decir, cualquier dato inusual, como son: problemas de cardinalidad irregular, valores atípicos, valores faltantes, etc.**

La calidad de datos es un proceso de gran importancia que permite garantizar el procesamiento masivo de datos para contribuir a que los resultados obtenidos sean óptimos y contribuyan a la correcta toma de decisiones. Por tal motivo, el presente trabajo se enfoca en realizar un análisis sobre la implementación de las técnicas de imputación y normalización para la base de datos Iris, a fin de comprobar la eficacia de estos métodos.

***Temas claves*— Aprendizaje máquina, calidad de datos, Imputación, Normalización.**

I. Introducción

El aprendizaje máquina ha llamado mucho la atención de los académicos y de las industrias durante las últimas décadas y continúa logrando un rendimiento impresionante a nivel humano en tareas como son: la clasificación de imágenes, el reconocimiento de voz, el procesamiento del lenguaje natural, etc.

Los datos son la materia prima que alimenta los algoritmos de aprendizaje máquina, ya que nos permite tomar decisiones en función del análisis del comportamiento histórico y con ello poder predecir futuros comportamientos. Sin embargo, la gran mayoría de los datos sin procesar se ven afectados por factores negativos como son: ruido, valores perdidos, inconsistencias, tamaño demasiado grande en cualquier dimensión (número de atributos e instancias), etc., lo cual repercute de manera negativa en los resultados que arroje el modelo en cuestión.

Debido a lo anterior, el presente trabajo se enfoca en la calidad de los datos, en específico en la etapa de preprocesamiento, la cual es fundamental en el proceso de descubrimiento de conocimiento; en donde se realiza un análisis comparativo entre el histograma original y los histogramas resultantes al implementar diversas técnicas de imputación, como son: Imputación Vecinos cercanos, Imputación Media e Imputación Hot deck; y normalización, como son: normalización máximo-mínimo, normalización de puntuación z.

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

II. Objetivo

Introducir al lector en el tema de la calidad de los datos, en donde se aborda la etapa de preprocesamiento de los mismos, en específico las diversas técnicas de imputación y normalización, con el fin preservar la forma de la distribución del conjunto de datos en cuestión, y ende evitar introducir sesgos en los resultados.

III. Marco Teórico

La preparación de datos es una etapa fundamental en cualquier proyecto de aprendizaje máquina, debido a que por medio de ella se puede inicializar correctamente los datos para su posterior procesamiento y análisis.

El preprocesamiento de los datos se encuentra conformado principalmente por la limpieza de datos e imputación, escalamiento y valores inconsistentes.

A. Limpieza de datos e imputación

La limpieza de datos cuenta con varios retos a solucionar, entre los cuales se encuentra el tema de los valores faltantes, el cual cuenta con varias maneras de lidiar con él, como son: Dejar intacto el conjunto de datos, debido a que existen modelos que pueden lidiar de manera correcta con este problema en cuestión; Técnicas de Imputación, en donde se estiman los valores faltantes por medio de los demás elementos presentes en el conjunto de datos; Eliminación de la instancia o atributo, esto se presenta si existe un porcentaje igual o mayor al 50 % de los valores faltantes; y readquisición de los datos, esto no es siempre posible ya que las condiciones presentes en el momento de la adquisición de los datos puede no ser las mismas.

Así bien, la limpieza de datos incluye también la búsqueda de instancias duplicadas, búsqueda de altas correlaciones entre atributos, etc., a fin de evitar duplicidades que puedan afectar la distribución de los datos y por ende sesgar la capacidad predictiva del algoritmo en posteriores etapas.

IMPUTACIÓN

Imputar significa sustituir observaciones, ya sea porque se carece de información (missing values) o bien se detecta que algunos de valores recolectados no se corresponden con el comportamiento esperado (outliers). En esta situación, es común que se desee reponer las observaciones y se decida aplicar algún método de sustitución de datos y de imputación.

Los métodos de imputación consisten en estimar los valores ausentes en base a los valores válidos de otras variables y/o casos de la muestra. La estimación se puede hacer a partir de la información del conjunto completo de variables o bien de algunas variables especialmente seleccionadas. Usualmente los métodos de imputación se utilizan con variables métricas (de intervalo o de razón), y deben aplicarse con gran precaución porque pueden introducir relaciones inexistentes en los datos reales.

Idealmente las técnicas de imputación no deberían cambiar la distribución de los datos. Así que si originalmente se tenía una distribución normal (con forma de campana), entonces después de la imputación se debería mantener esta distribución original.

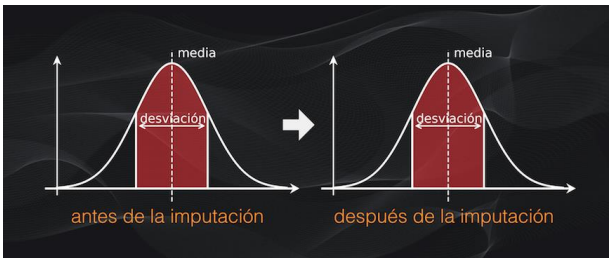


Fig.1. Comportamiento antes y después de la imputación.

En general, las técnicas de imputación se pueden clasificar de la siguiente manera:

- Técnicas por información externa o deductiva, se suele utilizar cuando los datos faltantes se pueden deducir de otras instancias completas.
- Técnicas deterministas, las cuales funcionan cuando, de acuerdo con las mismas condiciones de los datos se producen las mismas respuestas. Entre ellas, se encuentran la Imputación por regresión, Imputación de la media (o moda), Imputación por media de clases, Imputación por vecino más cercano, imputación por algoritmo EM.
- Técnicas estocásticas, son aquellas que cuando, se repite el método bajo las mismas condiciones, producen resultados diferentes. Entre estas técnicas se encuentran: Imputación aleatoria y la Imputación secuencial (hot deck).

Así bien, existen métodos de imputación múltiple, los cuales consisten en realizar varias imputaciones de las observaciones faltantes para luego analizar los conjuntos de datos completados y combinar los resultados obtenidos, con el fin de obtener una estimación final, entre los cuales se encuentran: Imputación múltiple por cadenas de Markov (MCMC) e Imputación múltiple por ecuaciones encadenadas (MICE).

B. Normalización y Escalamiento de los datos

Uno de los requisitos de muchos algoritmos de aprendizaje máquina es que los datos de entrada sean de naturaleza numérica. Sin embargo, en el mundo real, las observaciones no son necesariamente cuantitativos, lo cual resulta ser una restricción para el uso de tales algoritmos. Para representar las características categóricas como números, es necesario realizar una codificación categórica, como es la codificación one-hot, la cual crea una nueva característica que toma un valor booleano para cada etiqueta en una característica categórica. Esta nueva característica indica si esa etiqueta de la categoría está presente para cada observación.

Por otro lado, la escala de características se usa para cambiar los valores de las características y agruparlas dentro de un rango. Para escalar las características, se puede realizar:

La normalización, el cual tiene como objetivo cambiar los valores de las columnas numéricas en el conjunto de datos a una escala común, sin distorsionar las diferencias en los rangos de valores. Así bien, la normalización es útil cuando los datos tienen escalas variables y el algoritmo que está utilizando no hace suposiciones sobre la distribución de sus datos.

Un ejemplo de este tipo de normalización es la escala mínima-máxima, la cual reescala los valores de un rango de característica de 0 a 1, es decir, es la división de la resta de cada observación y el valor mínimo de este atributo entre la resta del valor máximo y mínimo de esa característica. Este método es muy sensible a los valores atípicos, ya que solo considera los valores finales de la característica.

$$x_{norm} = \frac{x - x_{min}}{x_{\max} - x_{norm}} \qquad (1)$$

La estandarización (o normalización de puntuación z) pretende centrar la media en cero y estandarizar la varianza en 1. El procedimiento consiste en restar la media de cada observación y luego dividirla por la desviación estándar. El resultado de la estandarización es conseguir que las características se vuelvan a escalar para que tengan las propiedades de una distribución normal estándar.

$$x' = \frac{x - \bar{x}}{\sigma} \qquad (2)$$

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.

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Virginica
146	6.3	2.5	5.0	1.9	Virginica
147	6.5	3.0	5.2	2.0	Virginica
148	6.2	3.4	5.4	2.3	Virginica
149	5.9	3.0	5.1	1.8	Virginica

Fig.2. Visualización de los atributos e Instancias del conjunto de datos Iris.

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[1], Kaggle[2], como también de Scikit Learn[3].

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.

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 la librería de matplotlib, la cual se utilizó para la visualización de la distribución de los datos antes y después de la implementación de cada una de las técnicas de imputación y normalización seleccionadas.

En la figura 4 se muestra las líneas de código implementadas para la obtención del total de valores comprendidos para cada uno de los porcentajes (10%, 20%, 30% y 50%) solicitados para la realización de esta tarea.

```
[22] Total = len(dataset[Col]) #Total de valores en la columna seleccionada
     P10 = (10*Total)/100 #Cantidad de valores con respecto al porcentaje 10%
     print('Cantidad total de valores comprendidos en el 10%: ' + str(P10))
     P20 = (20*Total)/100 #Cantidad de valores con respecto al porcentaje 20%
     print('Cantidad total de valores comprendidos en el 20%: ' + str(P20))
     P30 = (30*Total)/100 #Cantidad de valores con respecto al porcentaje 30%
     print('Cantidad total de valores comprendidos en el 30%: ' + str(P30))
     P50 = (50*Total)/100 #Cantidad de valores con respecto al porcentaje 50%
     print('Cantidad total de valores comprendidos en el 50%: ' + str(P50))

     Cantidad total de valores comprendidos en el 10%: 15.0
     Cantidad total de valores comprendidos en el 20%: 30.0
     Cantidad total de valores comprendidos en el 30%: 45.0
     Cantidad total de valores comprendidos en el 50%: 75.0
```

Fig.4. Total de valores con respecto al porcentaje en cuestión.

Así bien, una vez que se contó con cada uno de los porcentajes antes mencionados se prosiguió con la generación de valores faltantes, por medio de las siguientes funciones presentadas en la figura 5; debido a que el conjunto de datos seleccionado no contiene valores faltantes.

```
[23] def val_aleatorio(P): # Números aleatorios
       NumP = []
       condicion = 'False'
       while condicion == 'False':
         numero = randint(0, Total)
         if numero not in NumP:
           NumP.append(numero)
           if len(NumP) == P:
             condicion = 'True'
       return NumP
```

```
[24] def NaN1(dataset,Col,P):
       L =[]
       NumP = val_aleatorio(P)
       NumP.sort()
       print('Total de Valores Aleatorios: ',len(NumP), ', los cuales son:', NumP)
       for index, row in dataset.iterrows():
         for i in NumP:
           if index == i:
             row[Col]=np.nan
         L.append(row[Col])
       print('Sustitución de los valores en el dataset: ',L)
       print('')
       dataset[Col] = L
       return dataset
```

Fig.5. Funciones enfocadas a la generación de valores faltantes.

Posteriormente, se desarrollaron las funciones para cada una de las técnicas de imputación y normalización seleccionadas. En la figura 6, se muestra la función encargada de realizar la normalización MinMax, en donde como entrada se requiere el conjunto de datos así como la columna a la que se le desea aplicar la normalización.

```
def Norm(dataset,Col):
  MM = []
  for index, row in dataset.iterrows():
    MinMax =(row[Col]-np.min(dataset[Col]))/(np.max(dataset[Col])-np.min(dataset[Col]))
    MM.append(MinMax)
  dataset[Col] = MM
  return dataset
```

Fig.6. Función Normalización Minmax.

Así mismo, en la figura 7, se muestran las líneas de codigo utilizadas para la creación de la función de normalizacion de puntaje z, en donde de igual manera se requiere como datos de entrada la base de datos y la columna a la que se le desea aplicar la normalización.

```
def Z_score(dataset,Col):
  lista = []
  ZS = []
  for fila in dataset[Col]:
    lista.append(fila)
  sum_n = sum(dataset[Col]) #sumar el conjunto de valores de la columna en cuestión
  len_n = len(dataset[Col].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('Desviación estandar: ',std)

  for index, row in dataset.iterrows():
    D = row[Col] - promedio
    DPct = D / std
    #print(index, row[Col],DPct)
    ZS.append(DPct)
  dataset[Col]=ZS
  return dataset
```

Fig.7. Función Normalización puntaje Z.

Posteriormente, se presentan cada una de las funciones propuestas para el desarrollo de las técnicas de imputación seleccionadas, como es el caso de la figura 8, la cual contiene la función de Imputación Vecinos cercanos, en donde fue necesario de la utilización de dos funciones más para la obtención de la lista de las posiciones de los valores faltantes (figura 9) y el promedio obtenido una vez que se encontraron los valores más cercanos (figura 10).

```
def Imputacion_knn(Copy_knn,K):
  for Columna in Copy_knn:
    df = pd.DataFrame()
    print('Columna: ',Columna)
    V = []
    P = []
    for index, row in Copy_knn.iterrows():
        v.append(row[Columna]) #Obtención de los valores de la columna en cuestión
    print('Lista: ', V)
    P = Posiciones_NaN(V) #Si se encuentra un valor nan agregar su posicion en la lista P
    print('Lista de Posiciones NaN:', P)
    copy_knn = Copy_knn.drop([Columna], axis=1) #Eliminación de la columna en cuestión
    print(copy_knn)
    count=0
    for c in copy_knn:
      count =count+1
    #print(count)
    for n,i in enumerate(P):
      print('')
      print('Posición NaN: ',i)
      for col in copy_knn:
        dist = [] #Lista de distancias obtenidas entre registros
        for index, row in copy_knn.iterrows():
          index1= index + 1
          for index1, row1 in copy_knn.iterrows():
            if index == i:
              if index != index1:
                D = row1[col]-row[col] #Obtención de la distancia con respecto al índex de la columna el valor nan y las demas instancias
                D2 = math.pow(D,2)
                dist.append(D2)
              else: dist.append(000)
                #print('Index1:',index1,'Valor1: ', row1[col], 'Index: ',index, ' Valor: ', row[col], ' D: ', D, ' D2: ', D2)
        #print('')
        df[col] = dist
      print(df)
      print('')
      D_E=[]
      for f in range(df.shape[0]): #Valor de fila
        Valor=[]
        for c in range(df.shape[1]): #Valor de columna
          Valor.append(df.iloc[f, c]) #Agrupacion de las filas iguales de cada columna
        #print(Valor)
        V_nan = 0
        V_sum = 0
        for m, j in enumerate(Valor):
          if math.isnan(j):
            V_nan = V_nan + 1
          else: V_sum = V_sum + j
        #print(Valor, 'Total de valores nan: ', V_nan,'Suma Total: ', V_sum)
        if V_nan !=count:
          d = math.sqrt((len(Valor)/(len(Valor)-V_nan))*V_sum)#Obtención de la distancia euclidiana
          #print('Distancia Euclidiana:',d)
          D_E.append(d)
        else:
          D_E.append(np.nan)
      #print('Tamaño de la lista:',len(D_E),'Distancia Euclidiana: ',D_E)
      I = []
      Min_index =[]
      Min_valores = []
      for m,j in enumerate(dist):
        I.append(m)
      index_value_dist = {I:D_E for (I,D_E) in zip(I,D_E)}
      print('Diccionario:',index_value_dist) #Diccionario de los valores
      for w in sorted(index_value_dist, key = index_value_dist.get): #Ordenar los valores del diccionario de menor a mayor
        #print(w, index_value_dist[w])
        Min_index.append(w)
        Min_valores.append(index_value_dist[w])
      print('Tamaño de la lista:',len(Min_valores), 'Min_valores:',Min_valores)
      print('Tamaño de la lista:',len(Min_index), 'Min_index:',Min_index)
      for m,j in enumerate(Min_index):
        if j == i: #Quitar el índice, el cual se esta iterando
          Min_index.pop(m)
          Min_valores.pop(m)
      for m,j in enumerate(Min_valores):
        if math.isnan(j):
          Min_valores.pop(m)
          Min_index.pop(m)
      print('Tamaño de la lista:',len(Min_valores), 'Min_valores:',Min_valores)
      print('Tamaño de la lista:',len(Min_index), 'Min_index:',Min_index)
      MIN_I=Min_index[0:K]
      for j in range(K):
        Min_index.pop(0)
      print('Despues de quitar valores K:', Min_index)
      condicion = 'False'
      m_v=[]
      min_i = []
      while condicion == 'False': #Comprobación de no contar con valores nan en la lista m_v
        for m,j in enumerate(v):
          for n,o in enumerate(MIN_I):
            if o == m:
              if math.isnan(j):
                MIN_I.pop(n)
                MIN_I.append(Min_index[0])
                Min_index.pop(0)
              else:
                if o not in min_i:
                  min_i.append(o)
                  if len(min_i) == K:
                    condicion = 'True'
      print('Indices sin valores nan:',min_i)
      for m,j in enumerate(v):
          for n,o in enumerate(min_i):
            if o == m:
              m_v.append(j)
      print('Valores de la columna:',m_v)
      promedio = Promedio(m_v,K)
      print('Promedio:',round(promedio,2))
      for m,j in enumerate(v):
        if math.isnan(j):
          if m == i:
            v[m]=round(promedio,2)
      print('Lista modificada:', v)
      print('')
    Copy_knn[Columna] = v
  return Copy_knn
```

Fig.8. Función general enfocadas a la Imputación Vecinos Cercanos.

```
def Posiciones_NaN(V):
  P=[]
  for n, i in enumerate(V):
    if (math.isnan(i)): #si se encuentra un valor nan agregar su posicion en la lista P
      P.append(n)
  return P
```

Fig.9. Función Posiciones NaN.

```
def Promedio(lista,K):
    if len(lista) == 0:
      pass
    if sum(lista) == 0 :
      Promedio = 0
    else:
      Suma = sum(lista) #Suma de los valores comprendidos en la lista
      Promedio = Suma/K #Obtención del promedio de los valores minimos extraido de la lista propuesta
    return Promedio
```

Fig.10. Función Promedio.

Para el desarrollo de la función de Imputación por Media, se requirio de la implemetación de una función general, la cual se encarga de identificar cada una de posiciones, en donde se encuentra un valor faltante, a fin de llamar a la función sub_I_M, la cual se enfoca en encontrar los K valores superiores e inferiores a la posición del valor faltante (figura 12).

```
def IM(Copy_Media,Valor):
  for columna in Copy_Media:
    print('')
    print('Columna: ',columna)
    P = [] #posición
    P_adelante = []
    P_atras = []
    V = []
    for index, row in Copy_Media.iterrows():
      V.append(row[columna])

    print('Lista original:', V)
    for n,i in enumerate(V):
      if (math.isnan(i)):
        P.append(n) #Guardar indices con valores nan
    print('Lista de indices con valores nan:', P)

    for n,i in enumerate(V):
      if (math.isnan(i)):
        print('Indice: ', n, 'Valor:', V[n])
        v_medio = sub_I_M(V,P,Valor)
        print('Valor Medio:', v_medio, 'rodear a dos decimales:', round(v_medio,2))
        V[n] = round(v_medio,2)
        print('Lista modificada:', V)
        print('')
        P.pop(0)
    Copy_Media[columna] = V
  return Copy_Media
```

Fig.11. Función general enfocadas a la Imputación Media.

```
def sub_I_M(V,P,Valor):
  P_adelante=[]
  P_atras=[]
  for i in range(len(V)): #Recorrido de los indices de la columna
    if i == P[0]: #Si el indice coincide con el indice con valor nan
      #print(i)
      for k in (V[i+1:]): #Recorrido de los valores posteriores desde el indice con valor nan
        P_adelante.append(k) #Lista de valores secuenciales al valor aleatorio propuesto
      for j in (V[:i]): #Recorrido de los valores anteriores del índice aleatorio con valor nan
        P_atras.append(j) #Lista de valores secuenciales al valor aleatorio propuesto
      print('Lista de valores P_adelante secuenciales al indice con valor nan:', P_adelante)
      new_P_atras = [num for num in reversed(P_atras)] #Invertir la lista
      if len(P_atras)<75:
        for m in range(len(V)-1,P[0],-1): #Recorrido de los valores desde el indice aleatorio
          new_P_atras.append(V[m]) #Lista de valores secuenciales al valor aleatorio propuesto
      print('Lista de valores P_atras secuenciales al indice con valor nan:', new_P_atras)
      P_adelante = [x for x in P_adelante if math.isnan(x) == False] #Eliminar NaN de la lista P_adelante
      new_P_atras = [x for x in new_P_atras if math.isnan(x) == False] #Eliminar NaN de la lista P_atras
      v = []
      v_atras =v_at=0 #Condición inicial de variables
      v_adelante = v_ad= 0 #Condición inicial del contador
      for n,i in enumerate(new_P_atras):
        if v_atras != Valor: #Mientras el v_atras no sea igual a Valor(K)
          v.append(i) #Crear a la lista el valor de la lista
          v_at = v_at + i  #Sumar cada uno de los valores agregado a la lista
          v_atras = v_atras + 1  #Sumar 1 al contador
      for n,i in enumerate(P_adelante):
        if v_adelante != Valor: #Mientras al v_adelante no sea igual al Valor(K)
          v.append(i) #Crear a la lista el valor de la lista
          v_ad = v_ad + i   #Sumar cada uno de los valores agregados a la lista
          v_adelante = v_adelante + 1 #Sumar 1 al contador
      v_medio = ((v_ad + v_at)/len(v))
      return v_medio
```

Fig.12. Función enfocada a la obtención de los valores.

En la figura 13, se puede observar la función Imputación Hot Deck, la cual se encarga de reemplazar cada uno de los valores faltantes por medio de la obtención primeramente de una posición aleatoria, a fin de generar una lista de los valores secuenciales a esta posición.

```python
def Imputacion_HotDeck(columna,Copy_hotdeck,P):
  Total = len(Copy_hotdeck[columna])
  numero = np.random.randint(0, Total)
  V=[] #valores de la columna
  IA = [] #index aleatorio
  I =[]
  for index, row in Copy_hotdeck.iterrows():
    V.append(row[columna])
  Indice= recorrido(Copy_hotdeck, columna,numero)
  IA.append(Indice) #Guarda el indice aleatorio
  print('Lista de valores de la columna:')
  print(V)
  print('')
  for i in range(len(V)): #Recorrido de los indices de la columna
    if i == IA[0]: #Si el indice coincide con el indice aleatorio
      #print(V[i]) #comprobar que el valor coincida con el valor del registro
      for i in (V[i:]): #Recorrido de los valores desde el indice aleatorio
        I.append(i) #Lista de valores secuenciales al valor aleatorio propuesto
  while len(I)<(int(P)*2): #Si la cantidad de valores no es mayor al doble de P50
    for i in (V[:Total]): #Recorrido de los valores desde el indice aleatorio
        I.append(i) #Lista de valores secuenciales al valor aleatorio propuesto
  print('Lista de valores secuenciales al valor aleatorio propuesto:')
  print(I)
  I.pop(0) #Eliminación del primer valor de la lista
  print('Lista de valores secuenciales al valor aleatorio propuesto:')
  print(I)
  print('')
  newlist = [x for x in I if math.isnan(x) == False] #Lista de valores secuenciales al valor aleatorio propuesto sin valores faltantes
  print('Lista de valores secuenciales al valor aleatorio propuesto sin valores faltantes:')
  print(len(newlist), newlist)
  print('')
  for n, i in enumerate(V):
    if (math.isnan(i)): #Encontrar la posición que contenga valor nan
      V[n] = newlist[0] #Reemplazar el valor nan por el primer valor de lista newlist
      newlist.pop(0) #Eliminar el primer valor de la lista de valores secuenciales al valor aleatorio propuesto sin valores faltantes
  print('Sustitución de los valores en la lista original')
  print(V)
  print('')
  Copy_hotdeck[columna] = V
  return Copy_hotdeck
```

Fig.13. Función general enfocadas a la Imputación Hot Deck.

Diagrama de Flujo

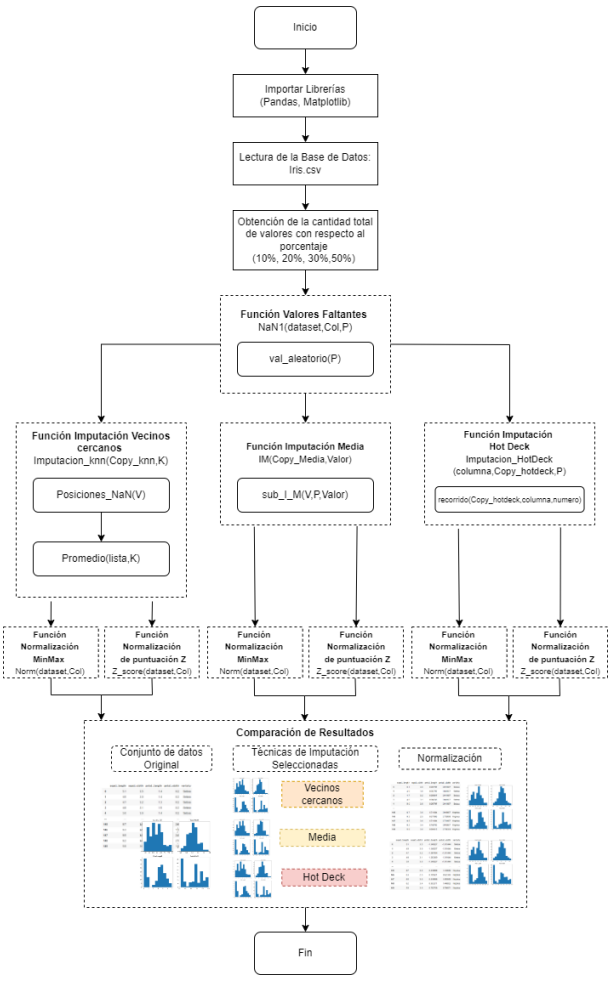


Fig.14.Diagrama de Flujo.

V. Resultados

A continuación, se presentan los histogramas obtenidos al realizar cada una de las técnicas de imputación y normalización en cuestión para los diferentes porcentajes de datos faltantes.

A. Porcentaje de datos faltantes 10%

Imputación Vecinos Cercanos

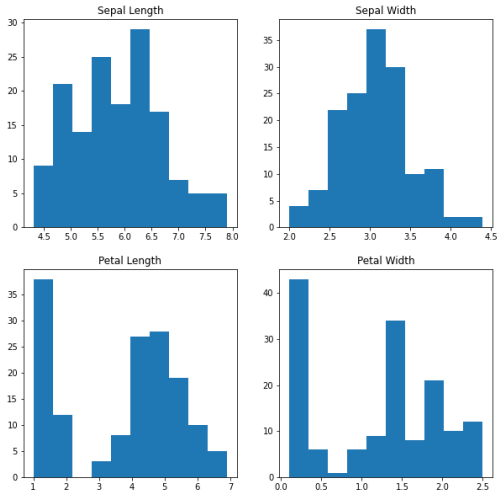


Fig.15. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 3.

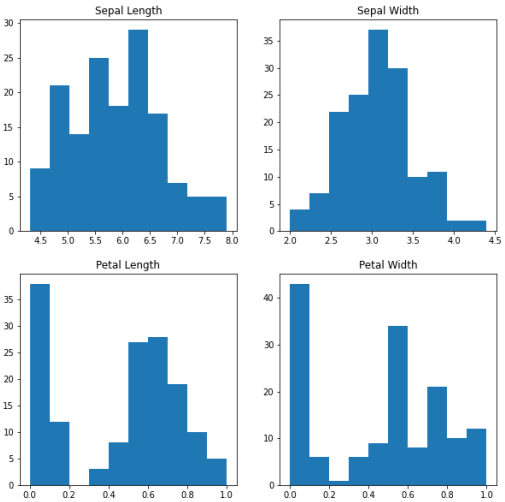


Fig.16. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 3 y la normalización MinMax.

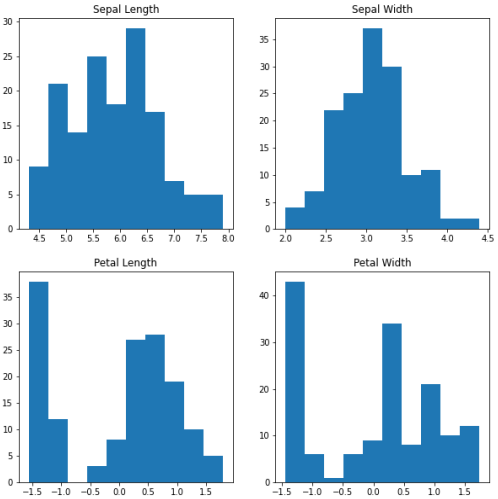


Fig.17. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 3 y la normalización Z score.

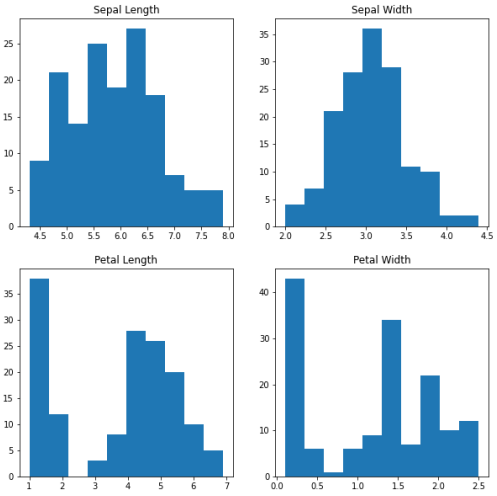


Fig.18. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 5.

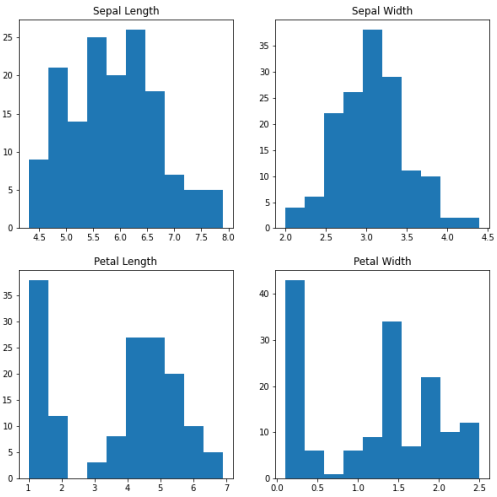


Fig.21. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 10.

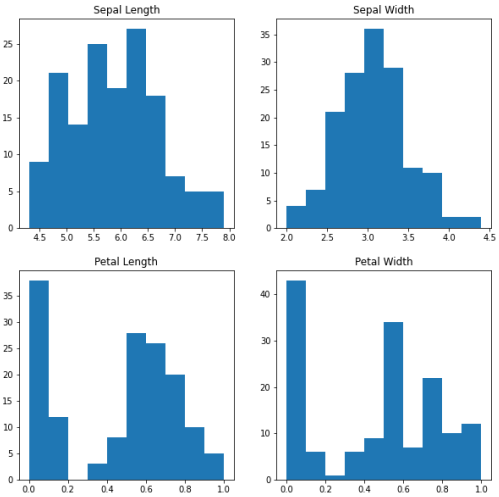


Fig.19. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 5 y la normalización MinMax.

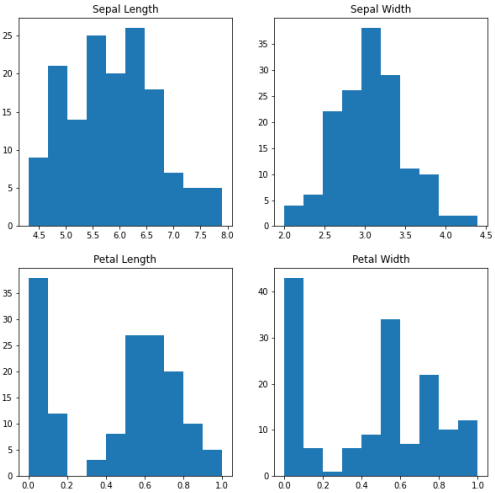


Fig.22. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 10 y la normalización MinMax.

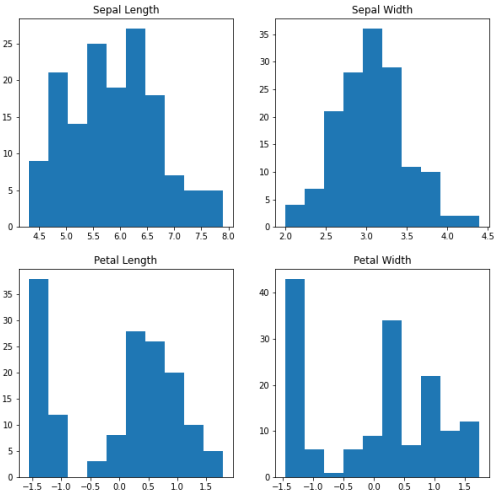


Fig.20. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 5 y la normalización Z score.

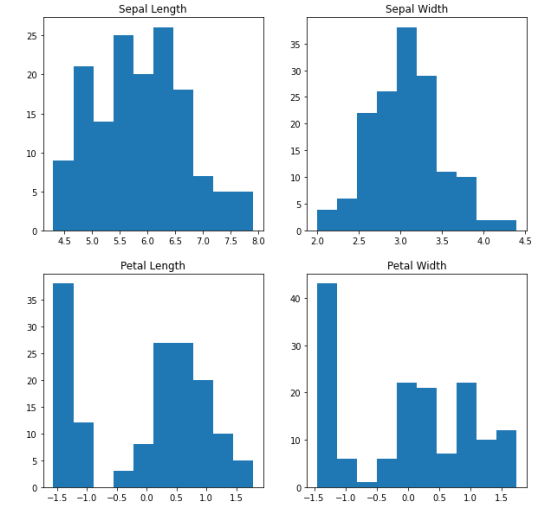


Fig.23. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 10 y la normalización Z score.

Imputación Media

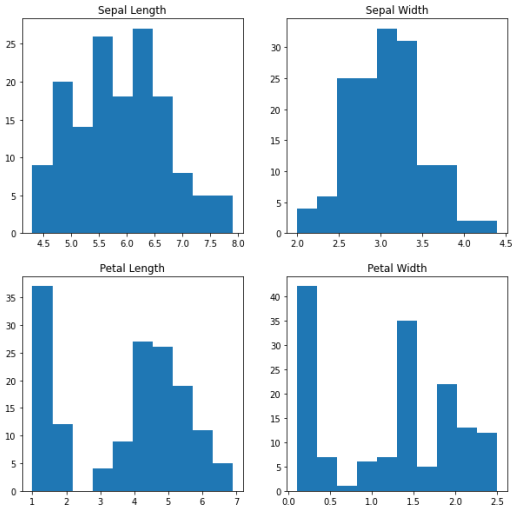


Fig.24. Histogramas de la base de datos Iris después de la Imputación Media con K = 3.

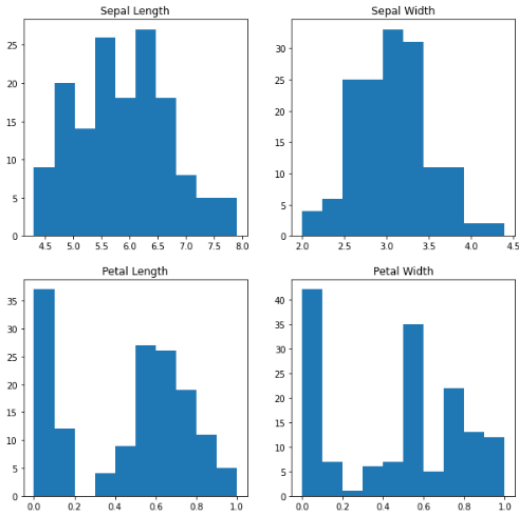


Fig.25. Histogramas de la base de datos Iris después de la Imputación Media con K = 3 y la normalización MinMax.

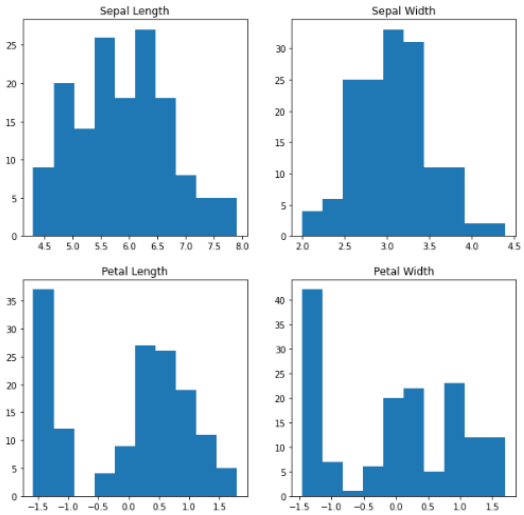


Fig.26. Histogramas de la base de datos Iris después de la Imputación Media con K = 3 y la normalización Z score.

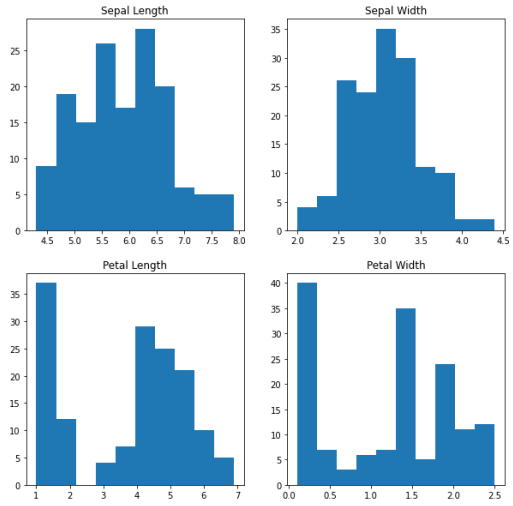


Fig.27. Histogramas de la base de datos Iris después de la Imputación Media con K = 5.

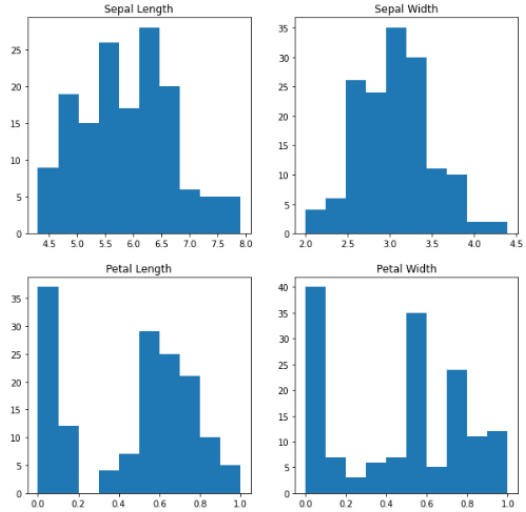


Fig.28. Histogramas de la base de datos Iris después de la Imputación Media con K = 5 y la normalización MinMax.

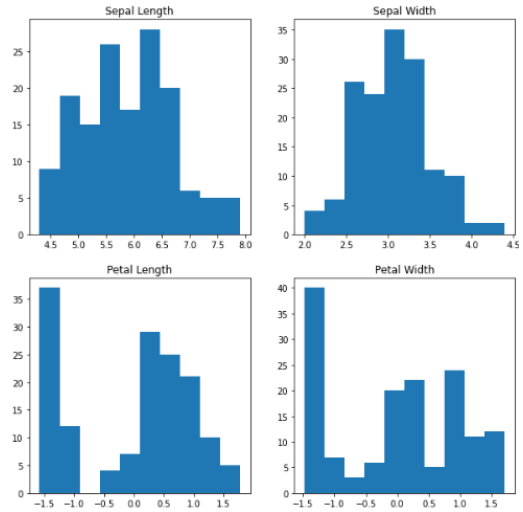


Fig.29. Histogramas de la base de datos Iris después de la Imputación Media con K = 5 y la normalización Z score.

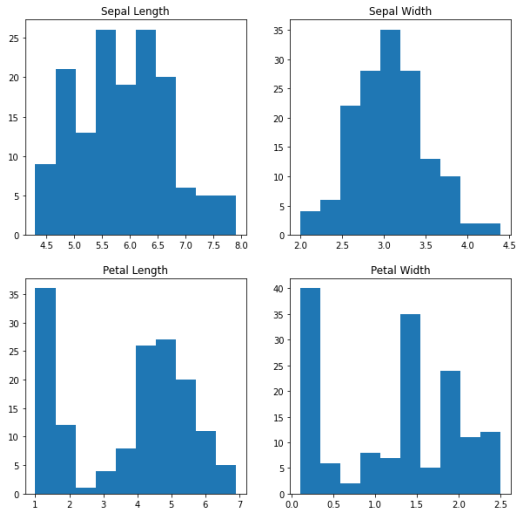


Fig.30. Histogramas de la base de datos Iris después de la Imputación Media con K = 10.

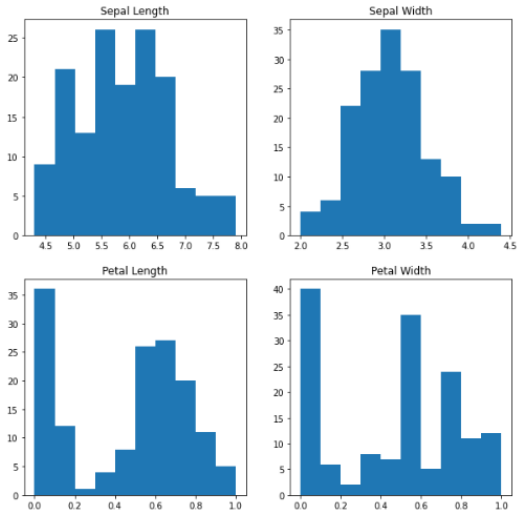


Fig.31. Histogramas de la base de datos Iris después de la Imputación Media con K = 10 y la normalización MinMax.

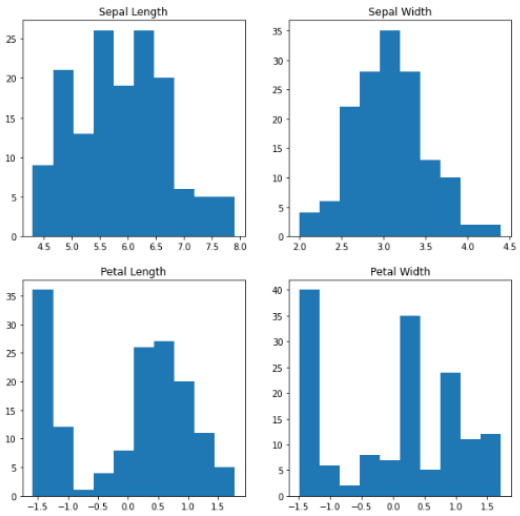


Fig.32. Histogramas de la base de datos Iris después de la Imputación Media con K = 10 y la normalización Z score.

Imputación Hot Deck

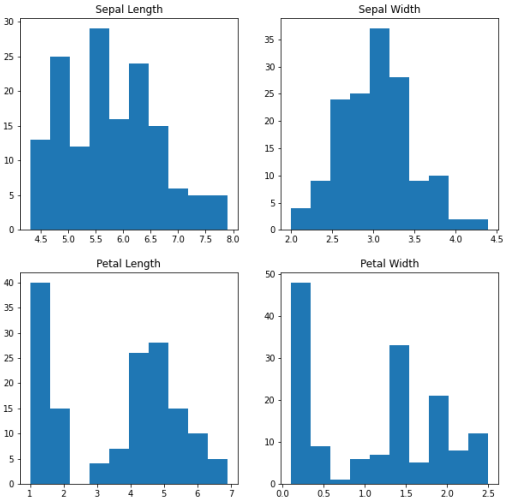


Fig.33. Histogramas de la base de datos Iris después de la Hot Deck

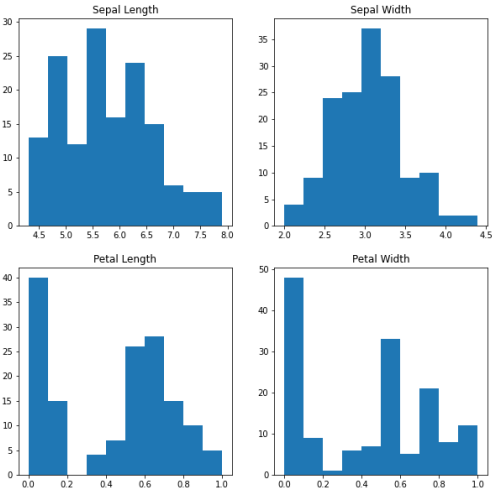


Fig.34. Histogramas de la base de datos Iris después de la Imputación Hot Deck y la normalización MinMax.

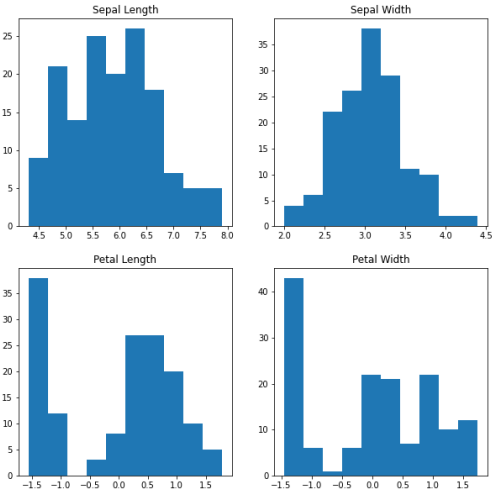


Fig.35. Histogramas de la base de datos Iris después de la Imputación Hot Deck y la normalización Z score.

B. Porcentaje de datos faltantes 20%

Imputación Vecinos Cercanos

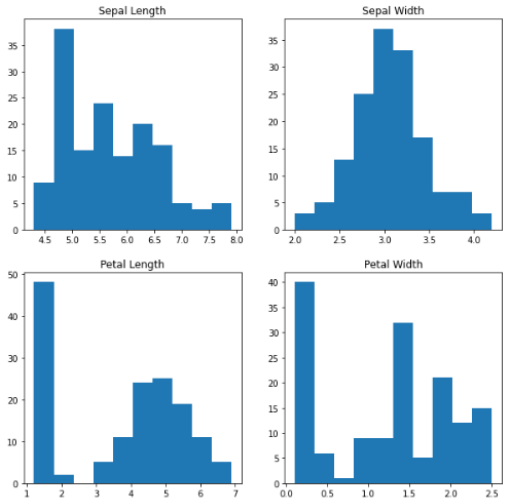


Fig.36. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 3.

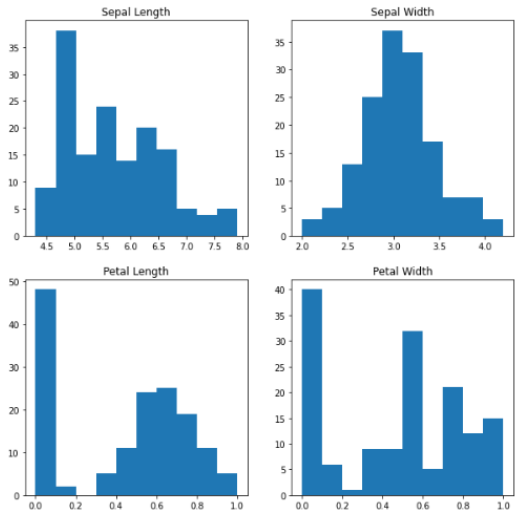


Fig.37. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 3 y la normalización MinMax.

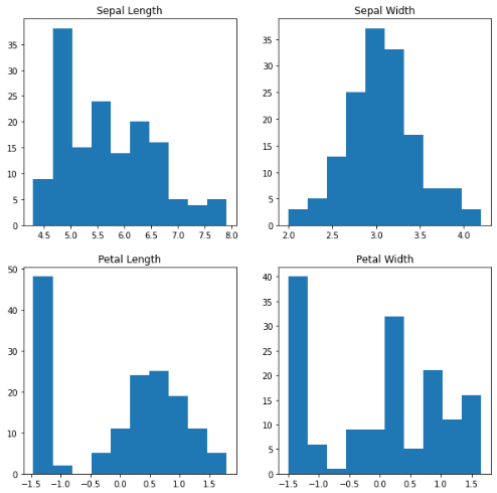


Fig.38. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 3 y la normalización Z score.

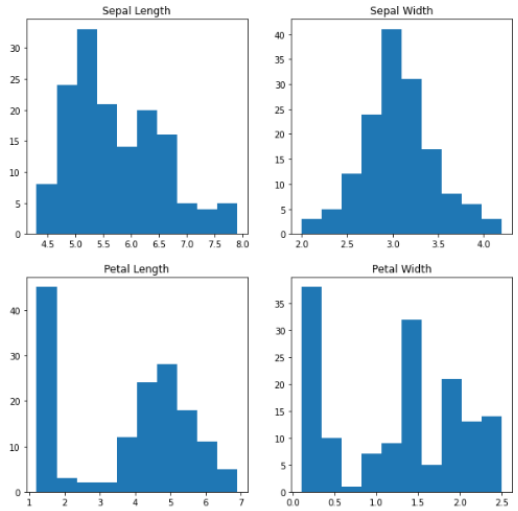


Fig.39. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 5.

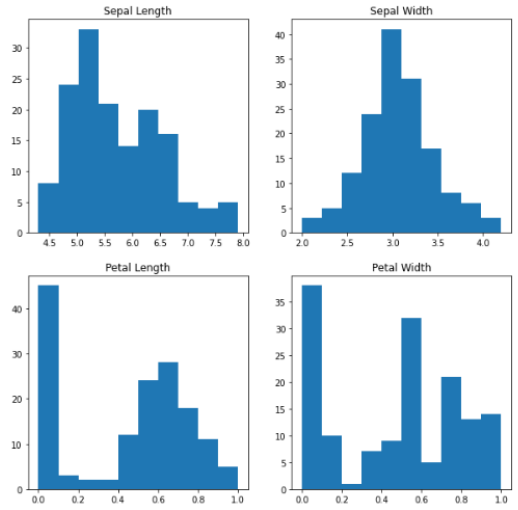


Fig.40. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 5 y la normalización MinMax.

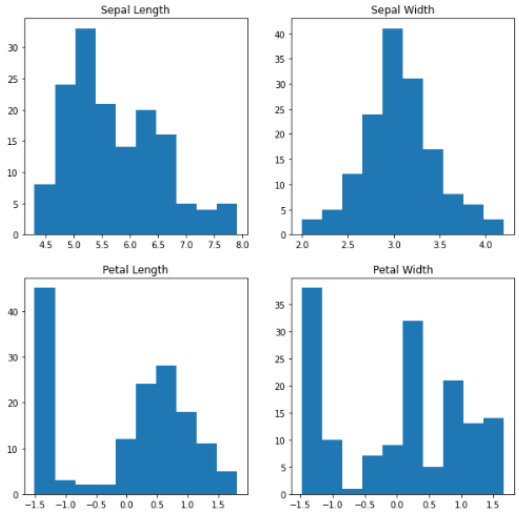


Fig.41. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 5 y la normalización z score.

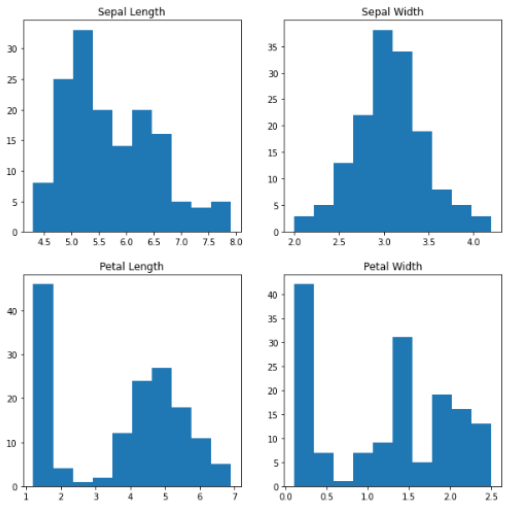


Fig.42. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 10.

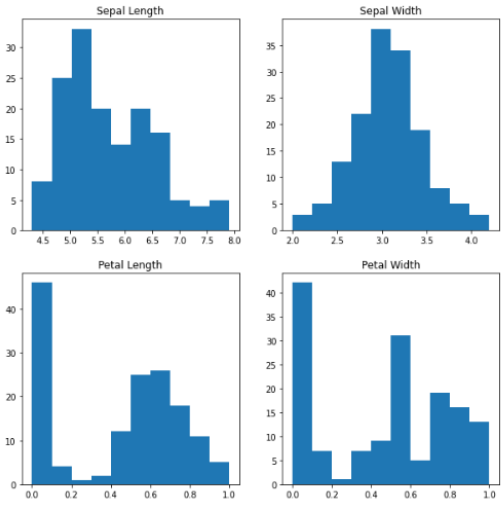


Fig.43. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 10 y la normalización MinMax.

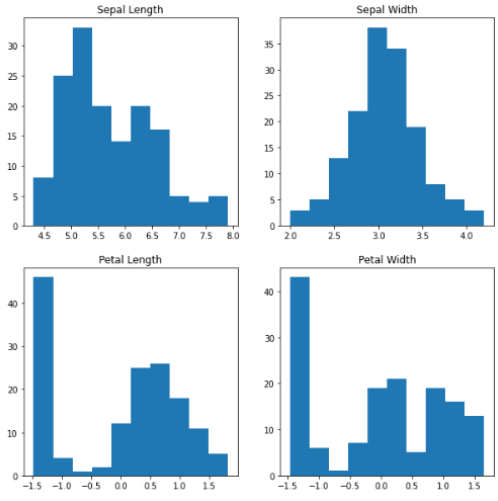


Fig.44. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 10 y la normalización z score.

Imputación Media

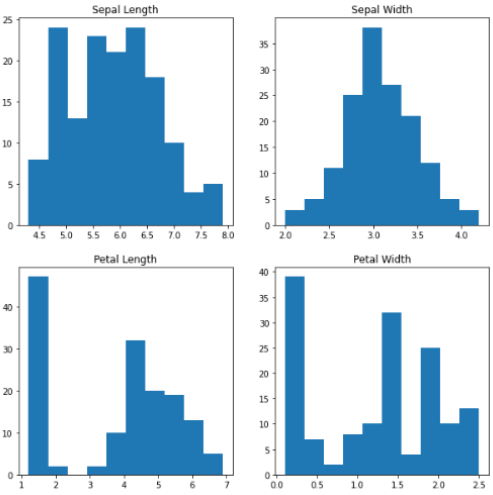


Fig.45. Histogramas de la base de datos Iris después de la Imputación Media con K = 3.

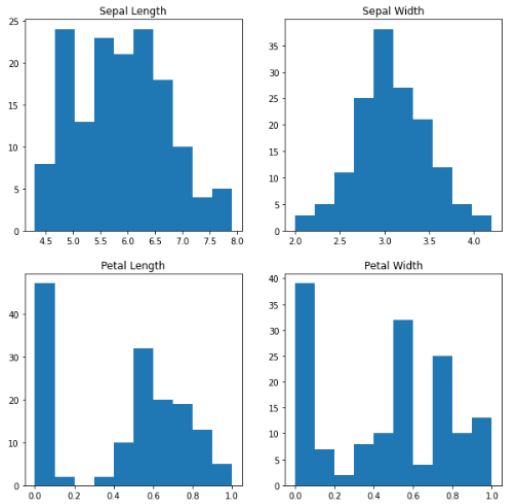


Fig.46. Histogramas de la base de datos Iris después de la Imputación Media con K = 3 y la normalización MinMax.

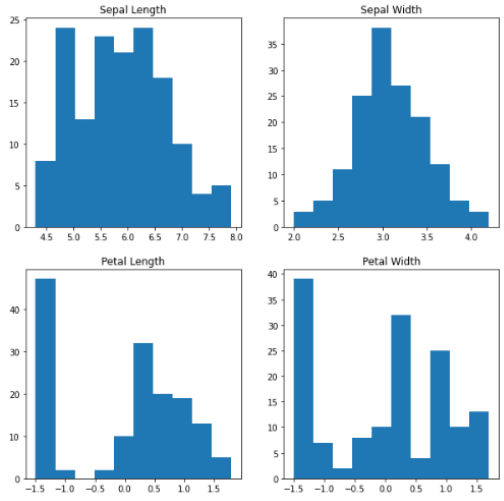


Fig.47. Histogramas de la base de datos Iris después de la Imputación Media con K = 3 y la normalización z score.

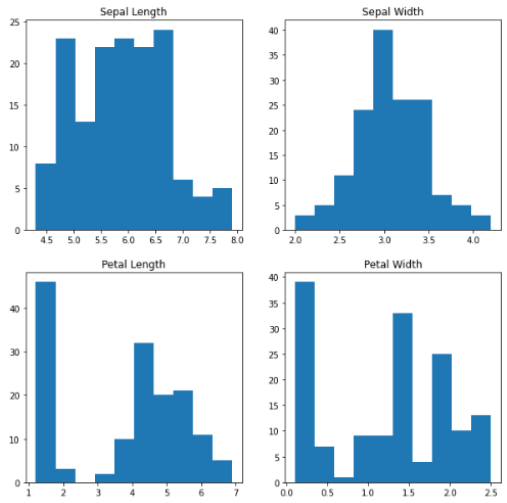


Fig.48. Histogramas de la base de datos Iris después de la Imputación Media con K = 5.

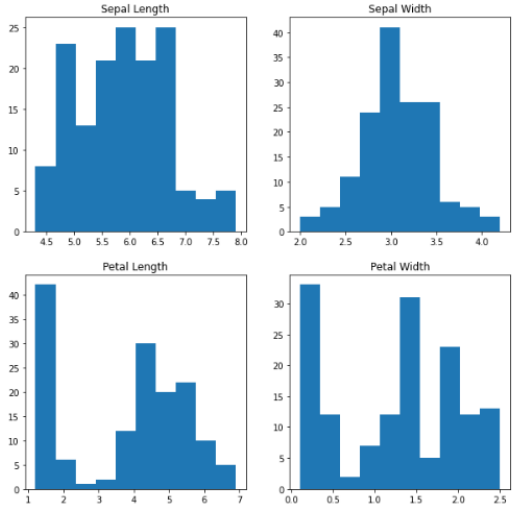


Fig.51. Histogramas de la base de datos Iris después de la Imputación Media con K = 10.

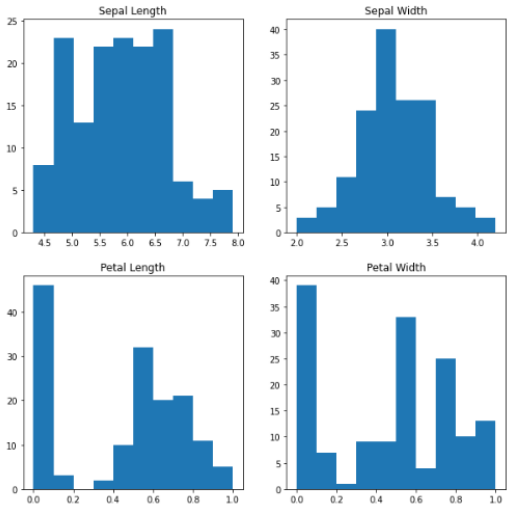


Fig.49. Histogramas de la base de datos Iris después de la Imputación Media con K = 5 y la normalización MinMax.

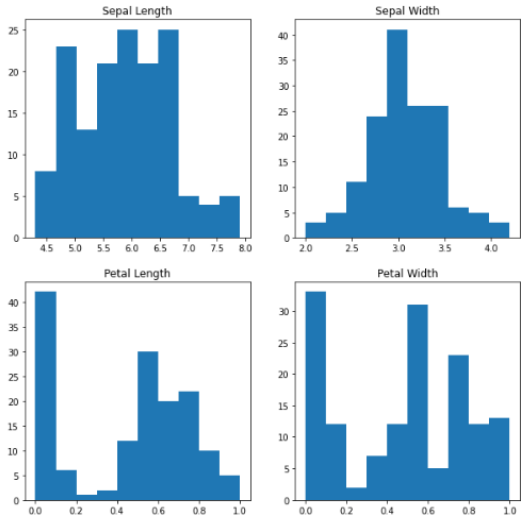


Fig.52. Histogramas de la base de datos Iris después de la Imputación Media con K = 10 y la normalización MinMax.

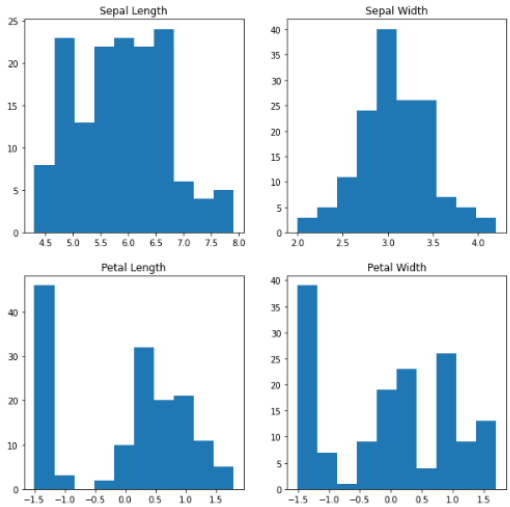


Fig.50. Histogramas de la base de datos Iris después de la Imputación Media con K = 5 y la normalización z score.

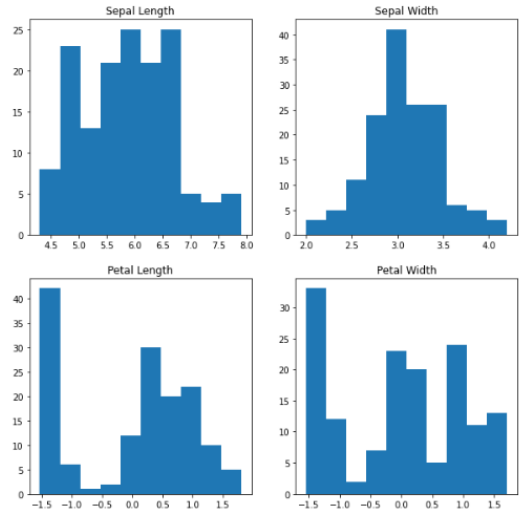


Fig.53. Histogramas de la base de datos Iris después de la Imputación Media con K = 10 y la normalización z score.

Imputación Hot Deck

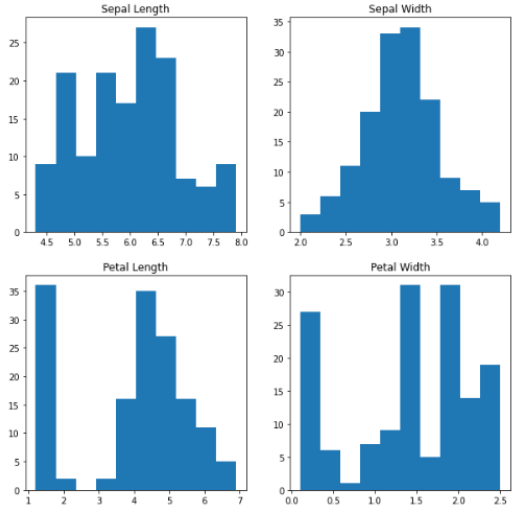


Fig.54. Histogramas de la base de datos Iris después de la Hot Deck

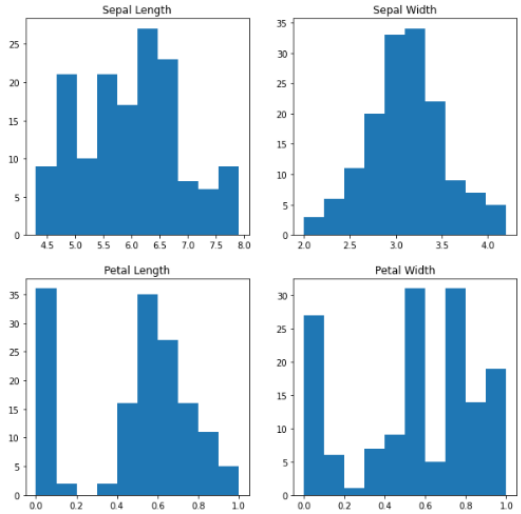


Fig.55. Histogramas de la base de datos Iris después de la Imputación Hot Deck y la normalización MinMax.

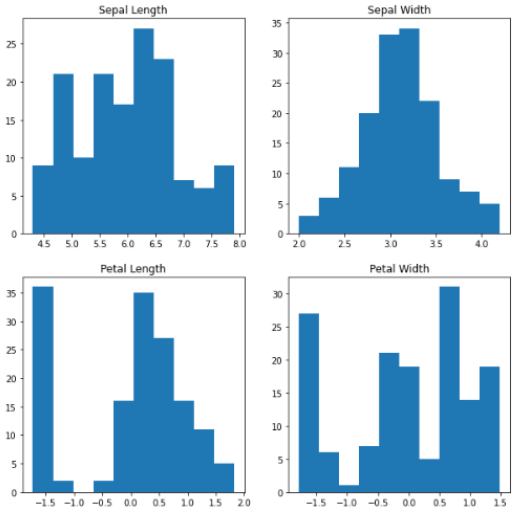


Fig.56. Histogramas de la base de datos Iris después de la Imputación Hot Deck y la normalización z score.

C. Porcentaje de datos faltantes 30%

Imputación Vecinos Cercanos

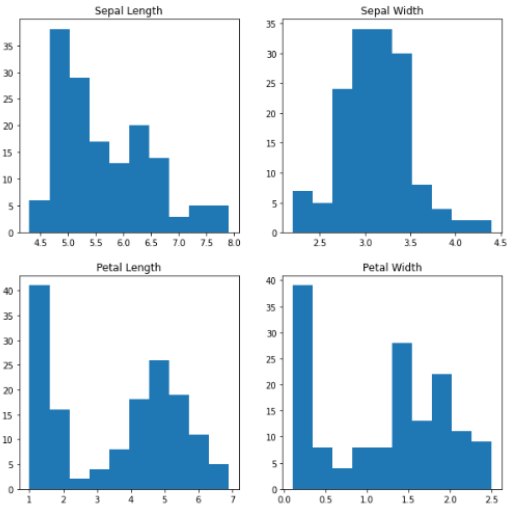


Fig.57. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 3.

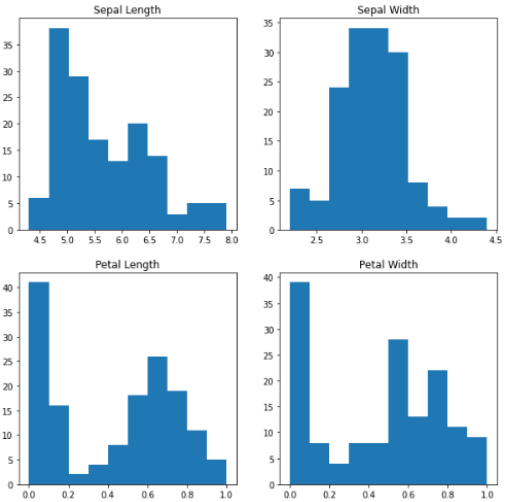


Fig.58. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 3 y la normalización MinMax.

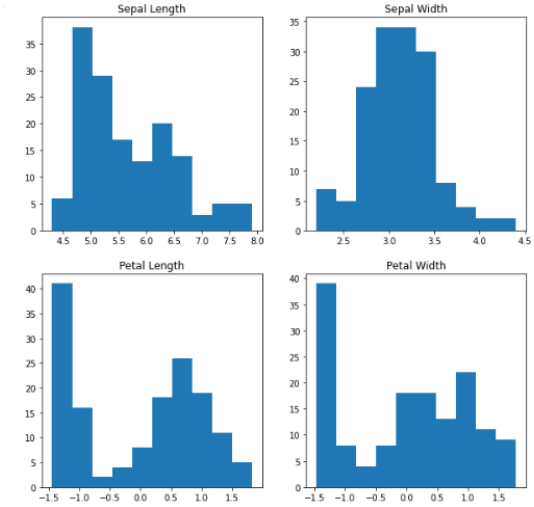


Fig.59. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 3 y la normalización MinMax.

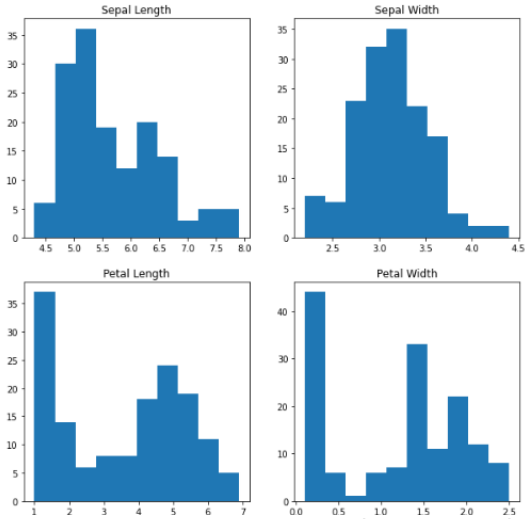


Fig.60. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 5.

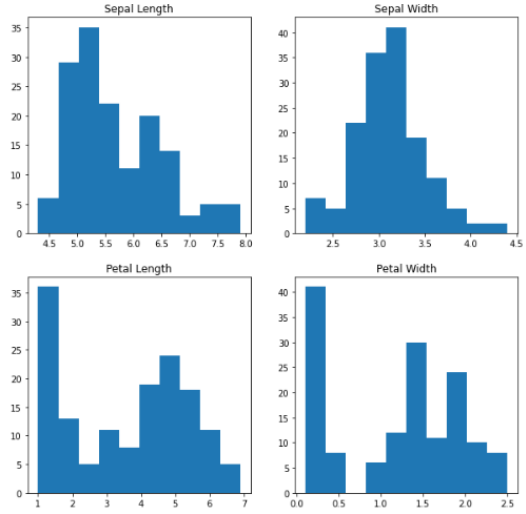


Fig.63. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 10.

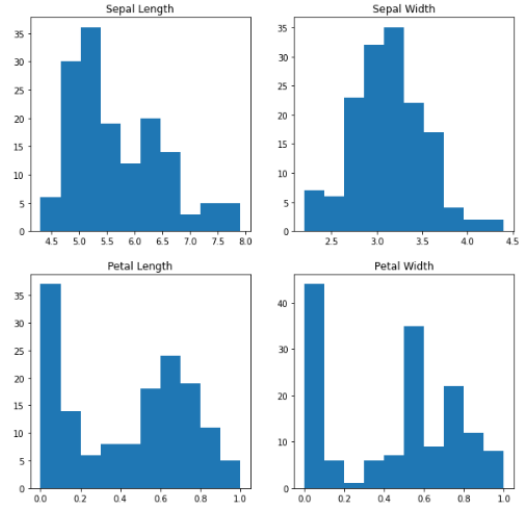


Fig.61. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 5 y la normalización MinMax.

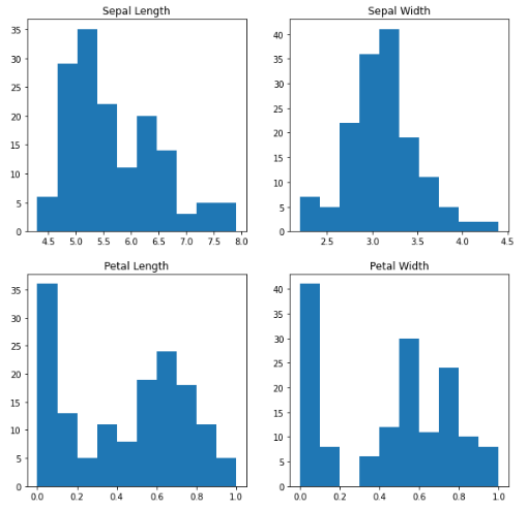


Fig.64. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 10 y la normalización MinMax.

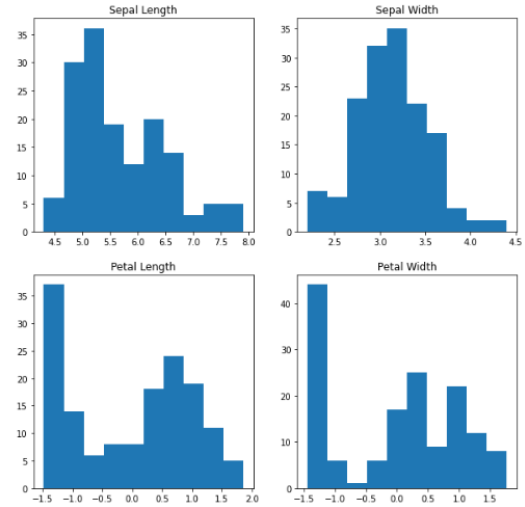


Fig.62. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 5 y la normalización z score.

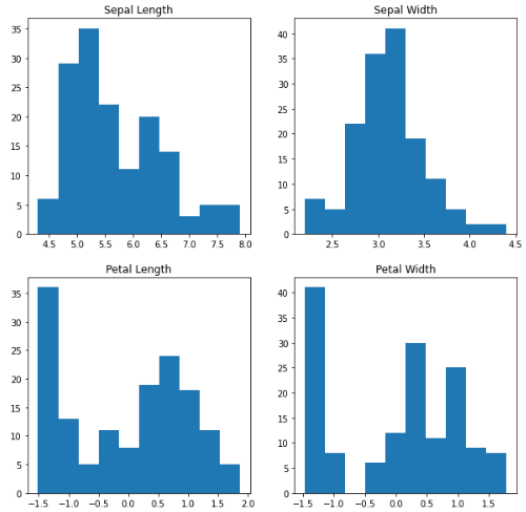


Fig.65. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 10 y la normalización z score

Imputación Media

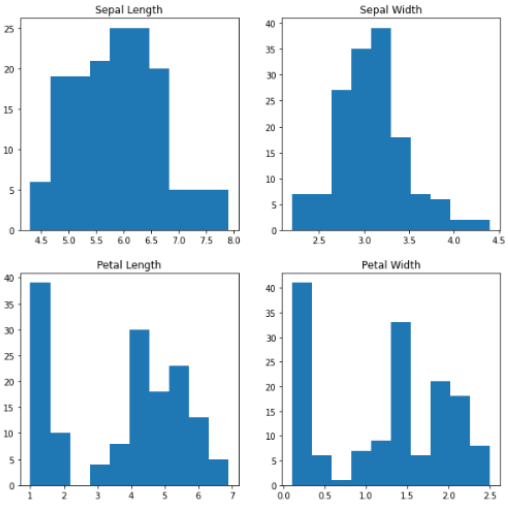


Fig.66. Histogramas de la base de datos Iris después de la Imputación Media con K = 3.

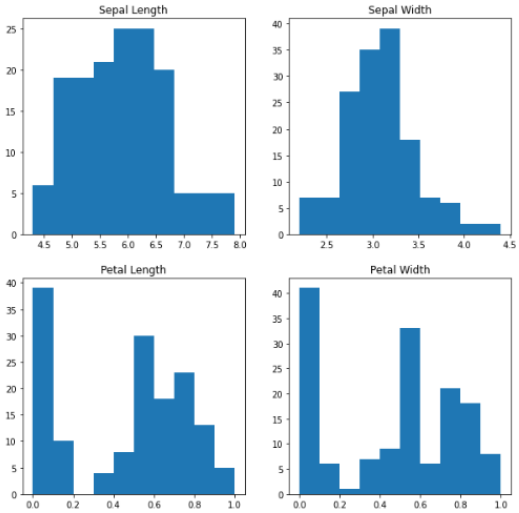


Fig.67. Histogramas de la base de datos Iris después de la Imputación Media con K = 3 y la normalización MinMax.

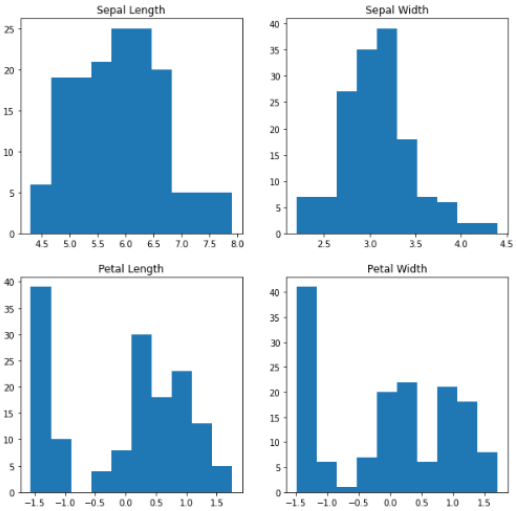


Fig.68. Histogramas de la base de datos Iris después de la Imputación Media con K = 3 y la normalización z score.

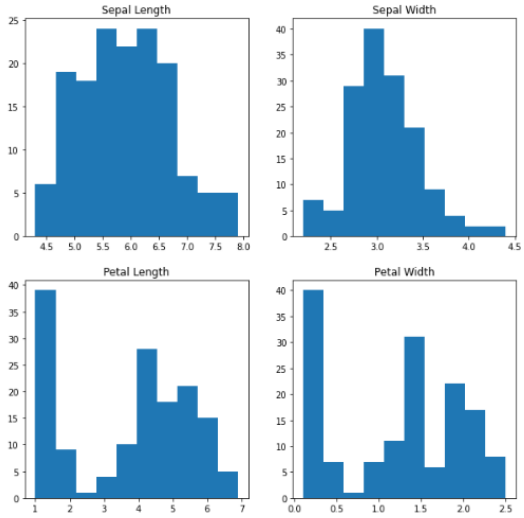


Fig.69. Histogramas de la base de datos Iris después de la Imputación Media con K = 5.

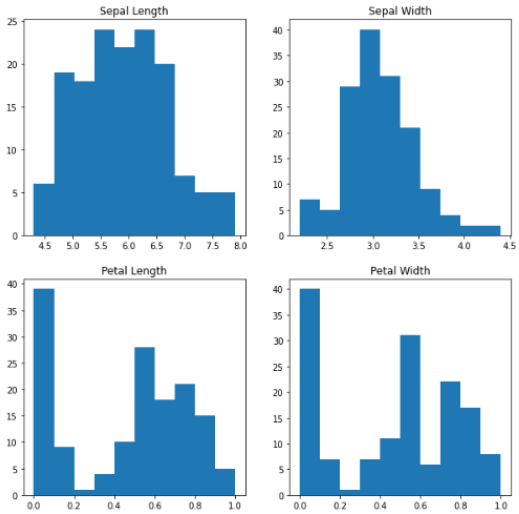


Fig.70. Histogramas de la base de datos Iris después de la Imputación Media con K = 5 y la normalización MinMax.

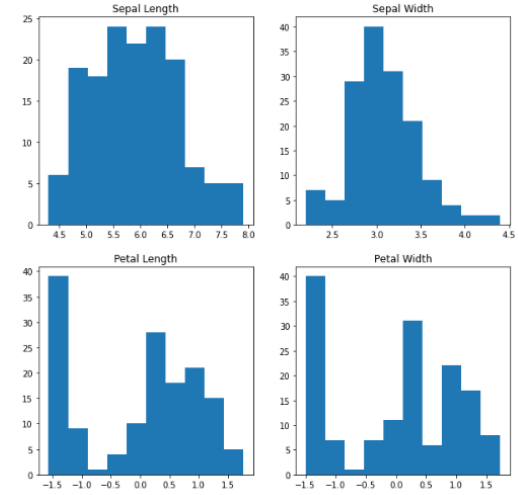


Fig.71. Histogramas de la base de datos Iris después de la Imputación Media con K = 5 y la normalización z score.

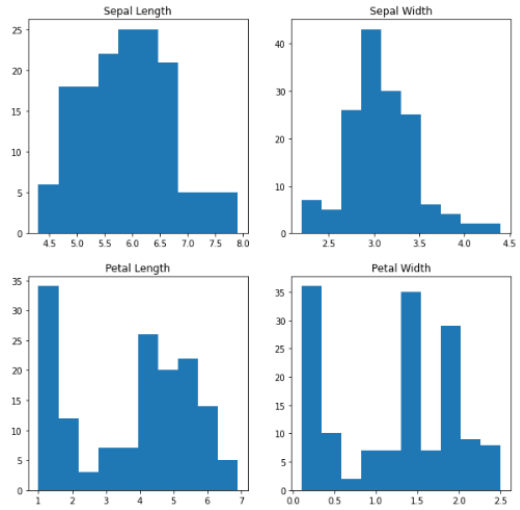


Fig.72. Histogramas de la base de datos Iris después de la Imputación Media con K = 10.

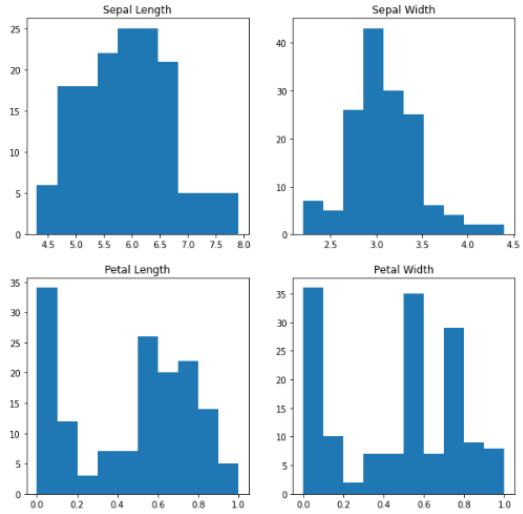


Fig.73. Histogramas de la base de datos Iris después de la Imputación Media con K = 10 y la normalización MinMax.

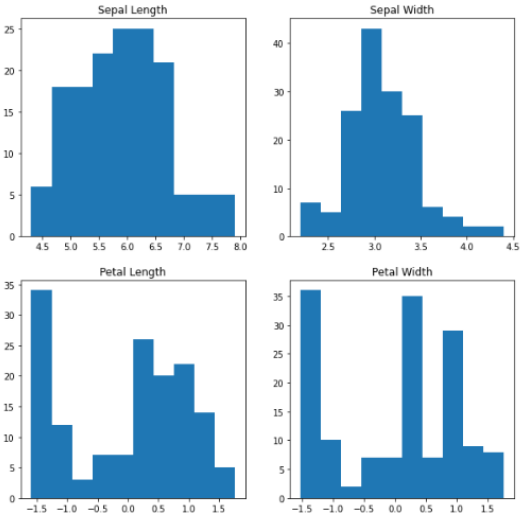


Fig.74. Histogramas de la base de datos Iris después de la Imputación Media con K = 10 y la normalización z score.

Imputación Hot Deck

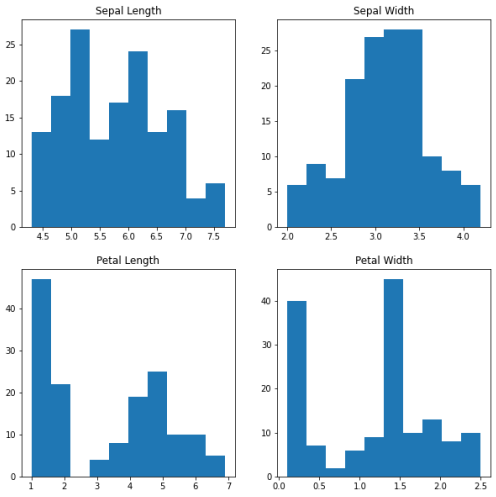


Fig.75. Histogramas de la base de datos Iris después de la Hot Deck

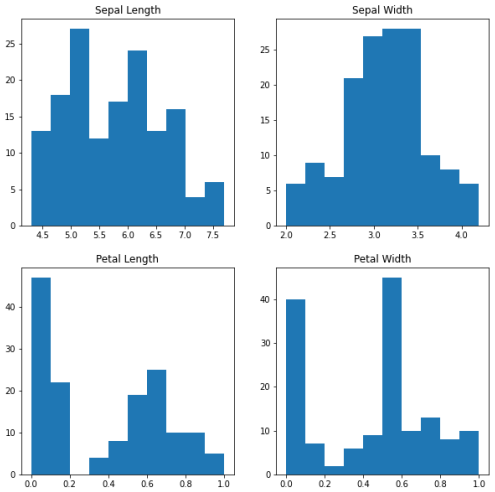


Fig.76. Histogramas de la base de datos Iris después de la Imputación Hot Deck y la normalización MinMax.

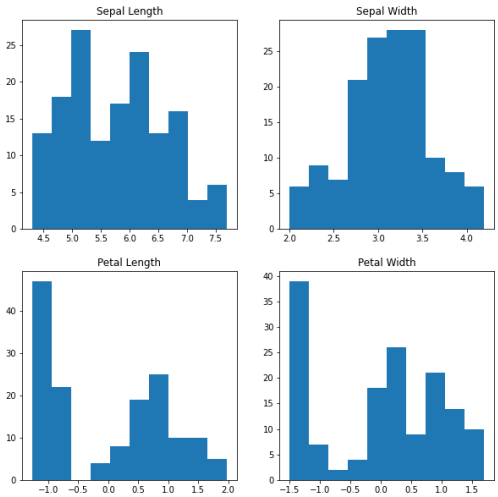


Fig.77. Histogramas de la base de datos Iris después de la Imputación Hot Deck y la normalización z score.

D. Porcentaje de datos faltantes 50%

Imputación Vecinos Cercanos

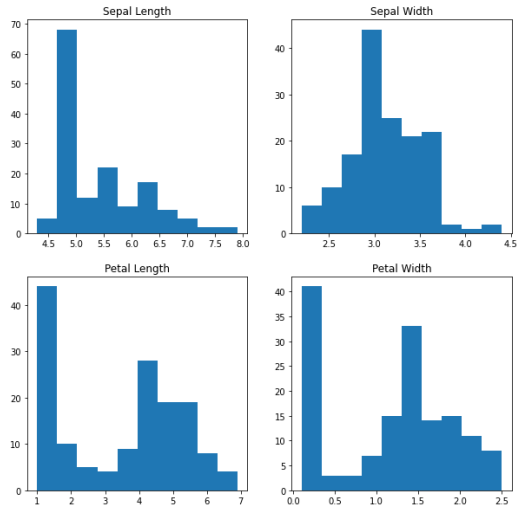


Fig.78. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 3.

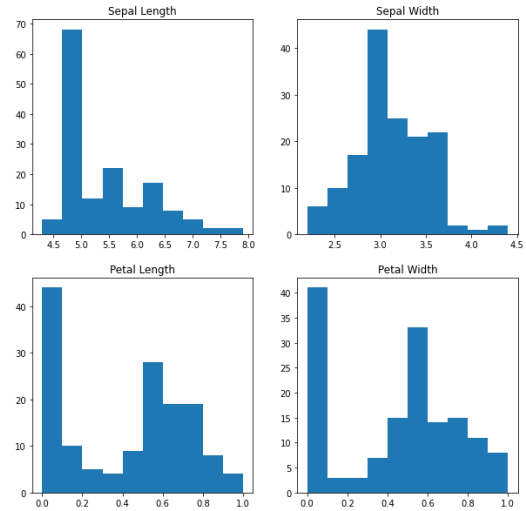


Fig.79. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 3 y la normalización MinMax.

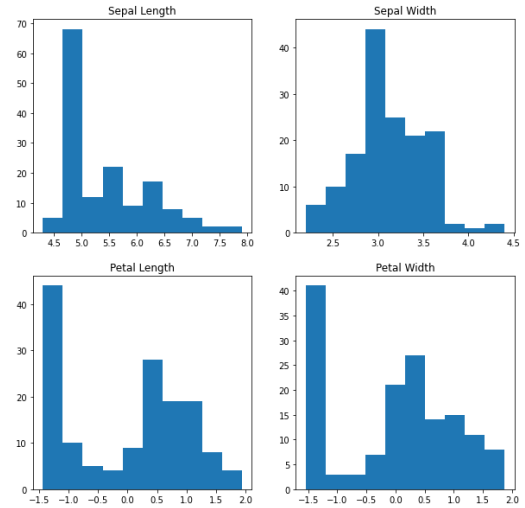


Fig.80. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 3 y la normalización z score.

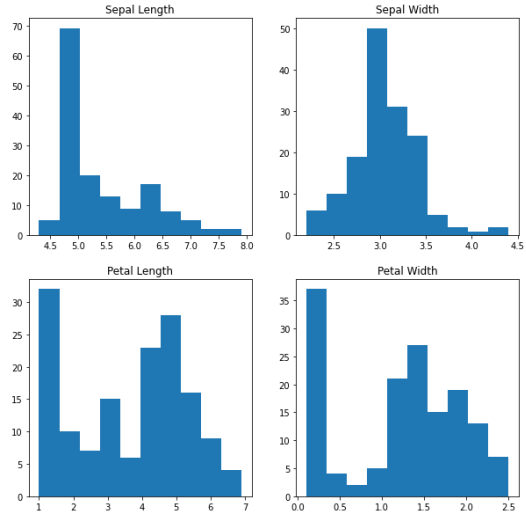


Fig.81. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 5.

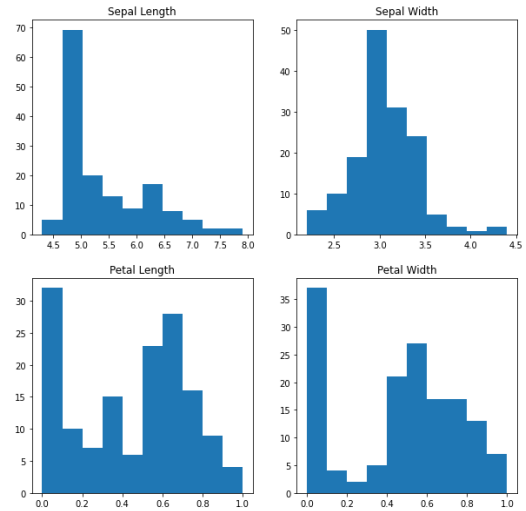


Fig.82. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 5 y la normalización MinMax.

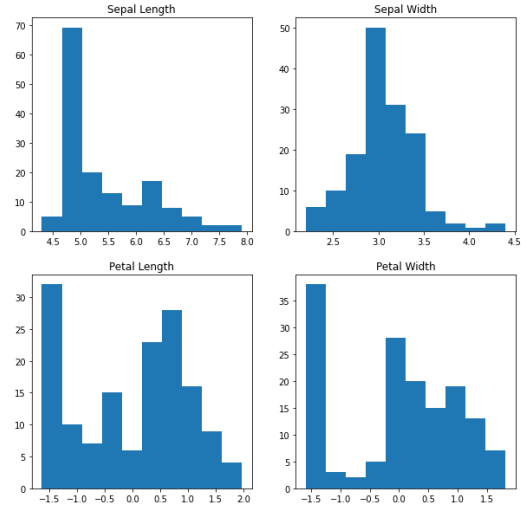


Fig.83. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 5 y la normalización z score.

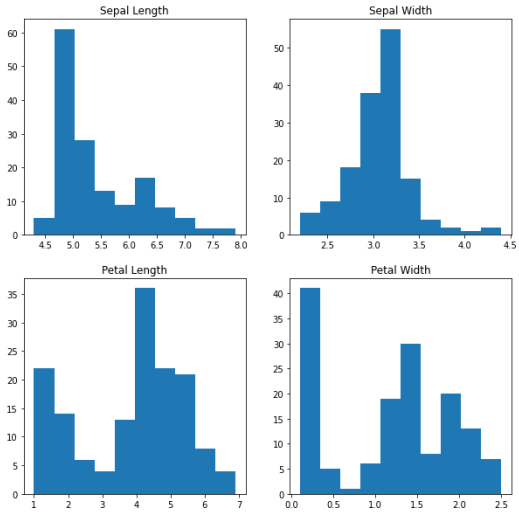


Fig.84. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 10.

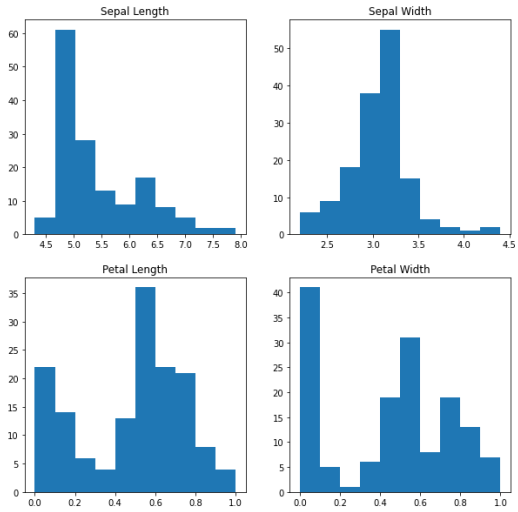


Fig.85. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 10 y la normalización MinMax.

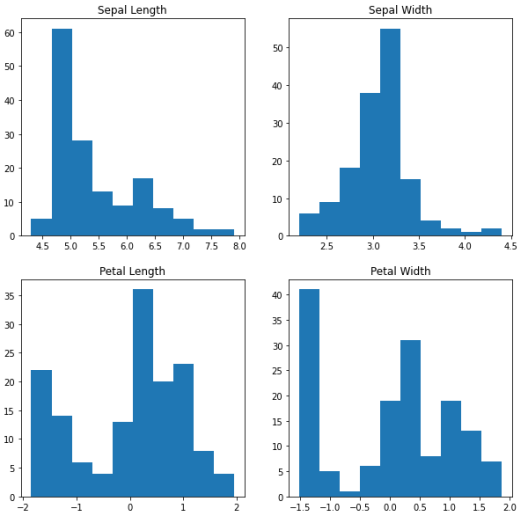


Fig.86. Histogramas de la base de datos Iris después de la Imputación Vecinos cercanos con K = 10 y la normalización z score.

Imputación Media

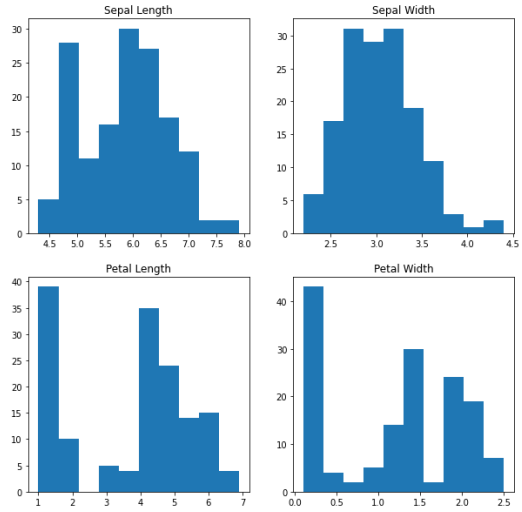


Fig.87. Histogramas de la base de datos Iris después de la Imputación Media con K = 3.

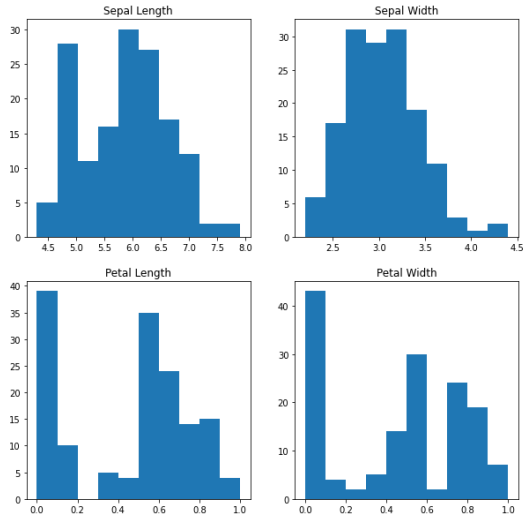


Fig.88. Histogramas de la base de datos Iris después de la Imputación Media con K = 3 y la normalización MinMax.

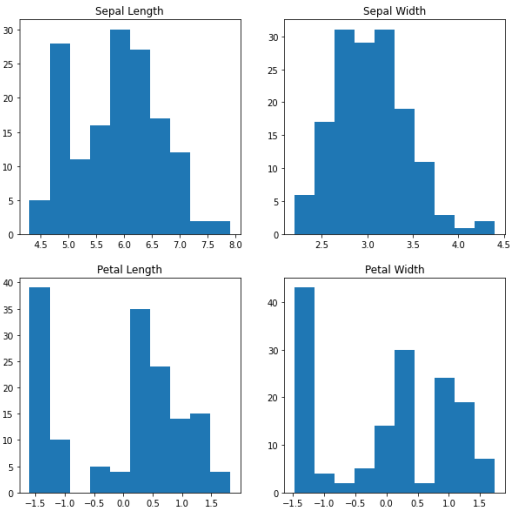


Fig.89. Histogramas de la base de datos Iris después de la Imputación Media con K = 3 y la normalización z score.

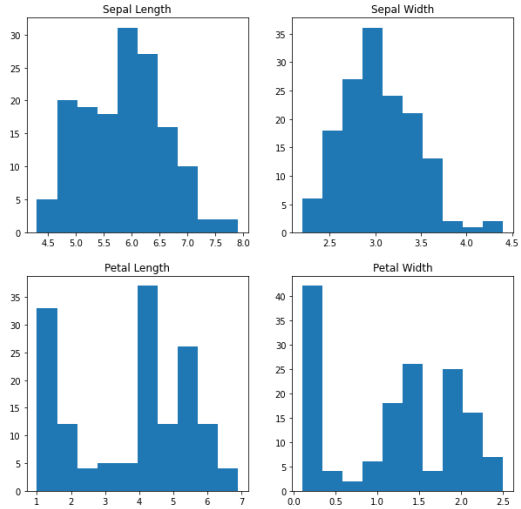


Fig.90. Histogramas de la base de datos Iris después de la Imputación Media con K = 5.

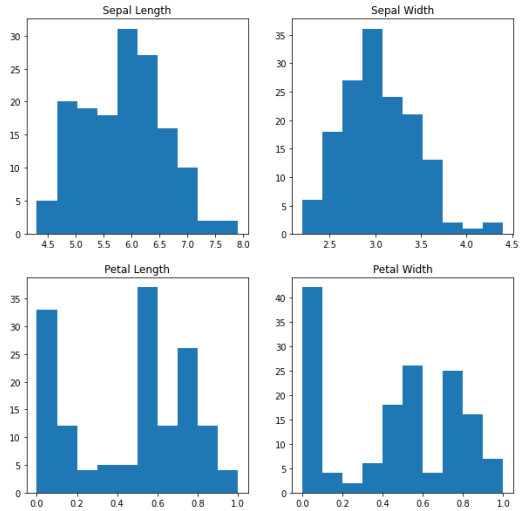


Fig.91. Histogramas de la base de datos Iris después de la Imputación Media con K = 5 y la normalización MinMax.

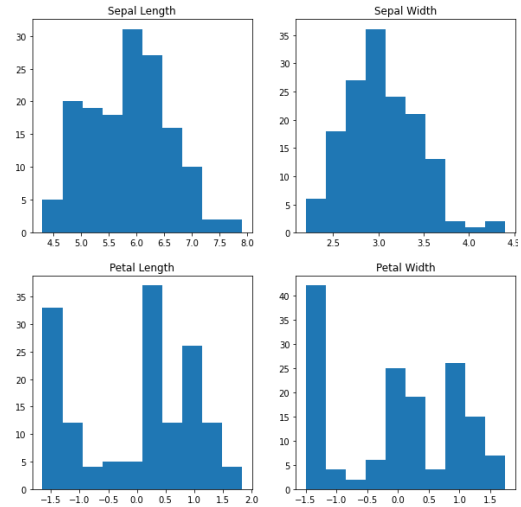


Fig.92. Histogramas de la base de datos Iris después de la Imputación Media con K = 5 y la normalización z score.

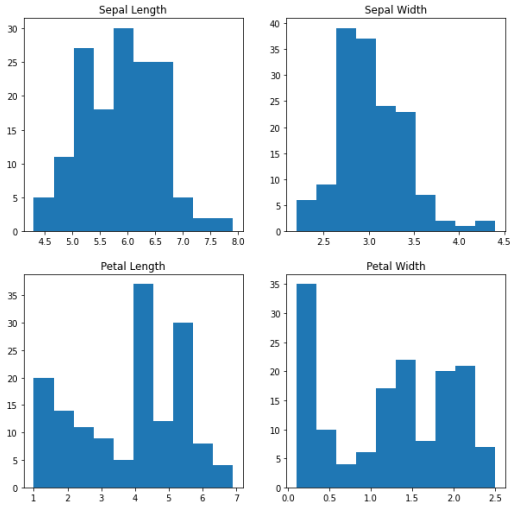


Fig.93. Histogramas de la base de datos Iris después de la Imputación Media con K = 10.

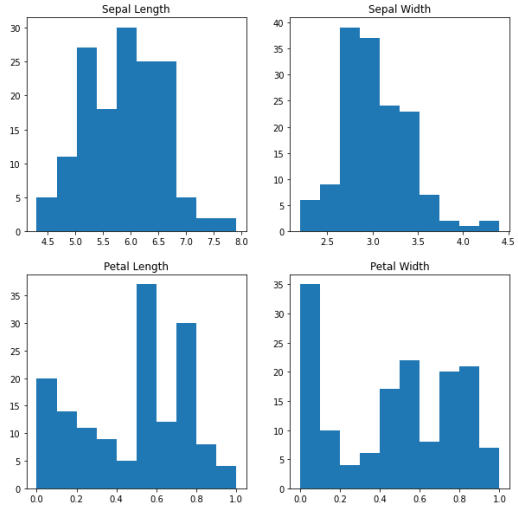


Fig.94. Histogramas de la base de datos Iris después de la Imputación Media con K = 10 y la normalización MinMax.

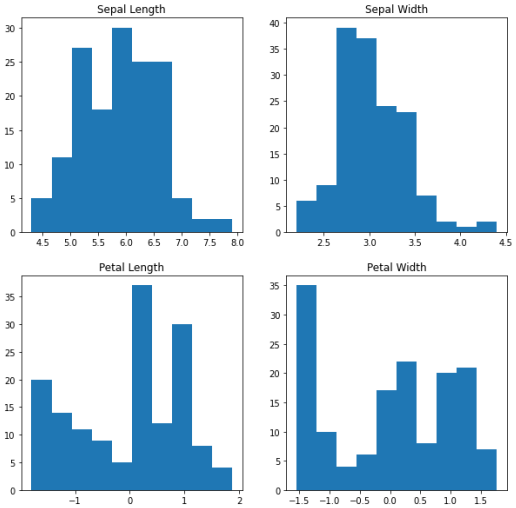


Fig.95. Histogramas de la base de datos Iris después de la Imputación Media con K = 10 y la normalización z score.

Imputación Hot Deck

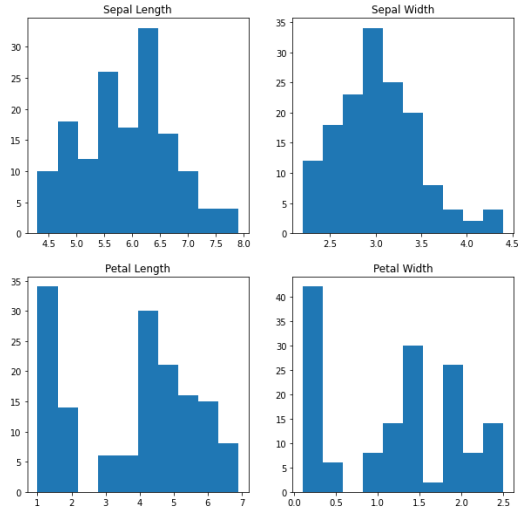


Fig.96. Histogramas de la base de datos Iris después de la Imputación Hot Deck

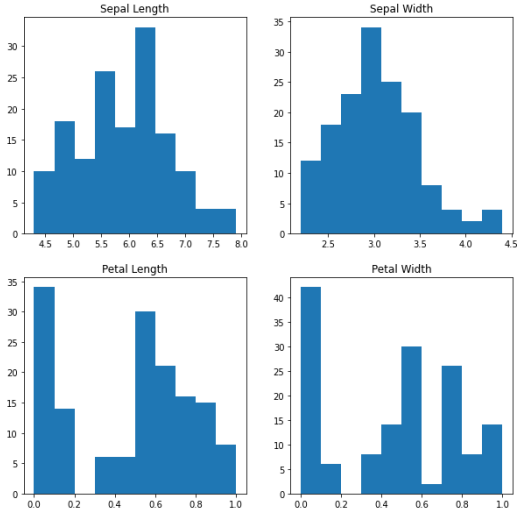


Fig.97. Histogramas de la base de datos Iris después de la Imputación Hot Deck y la normalización MinMax.

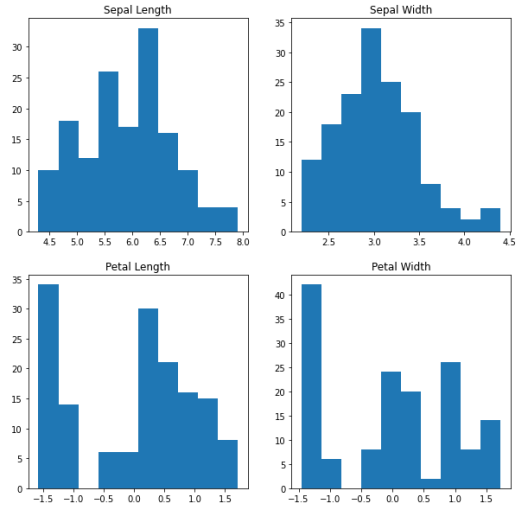


Fig.98. Histogramas de la base de datos Iris después de la Imputación Hot Deck y la normalización Z score.

Análisis de los resultados

Aunque existen distintos procedimientos para sustituir la falta de observaciones, es recomendable primeramente analizar las distribuciones resultantes al realizar cada una de las técnicas en cuestión antes de elegir alguna de ellas como definitiva. Por tal motivo, en esta sección se realizará una comparación de la distribución de los datos originales y los resultados obtenidos previamente, a fin de discernir cual de las técnicas de imputación y normalización son las más convenientes para el conjunto de datos en cuestión.

Técnicas de Imputación

A continuación, se muestra la comparación de las distribuciones obtenidas con respecto a la distribución original, en donde se puede observar primeramente los cambios sutiles que se producen en cada una de las distribuciones de los atributos al modificar el valor de K y el porcentaje de valores faltantes.

Imputación Vecinos Cercanos

En el caso de la Imputación de Vecinos Cercanos se puede observar en la figura 99 como se va modificando la cresta en el atributo de 'Petal Length' conforme va aumentando el valor de K, es decir, en la distribución multimodal de este atributo se puede visualizar el valor de tendencia que se encuentra del lado derecho y conforme se incrementa el valor de K, se observa como se terminan igualando dos columnas al mismo nivel.

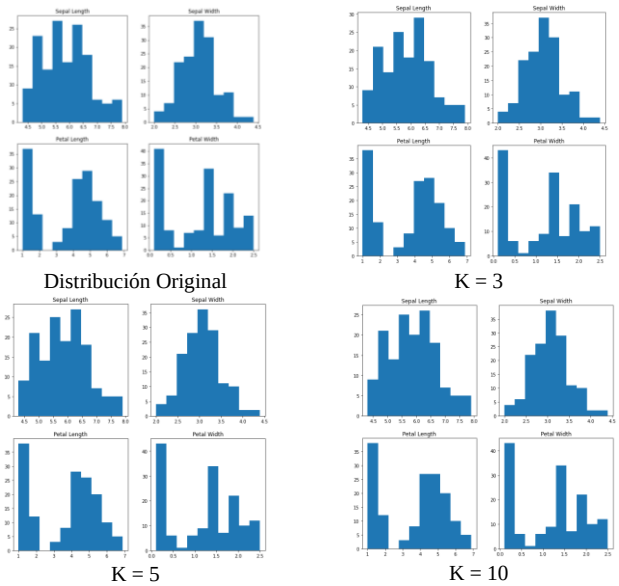


Fig.99. Comparación de la distribución original y las distribuciones con la implementación de la Imputación Vecinos cercanos para K = 3,5,10 cuando se cuenta con el 10% de los valores faltantes.

Así mismo, tomando de nueva cuenta ese mismo atributo, se puede observar en la figura 100 que al incrementar el porcentaje de los valores faltantes en el conjunto de datos en cuestión, la distribución de este atributo se compacto.

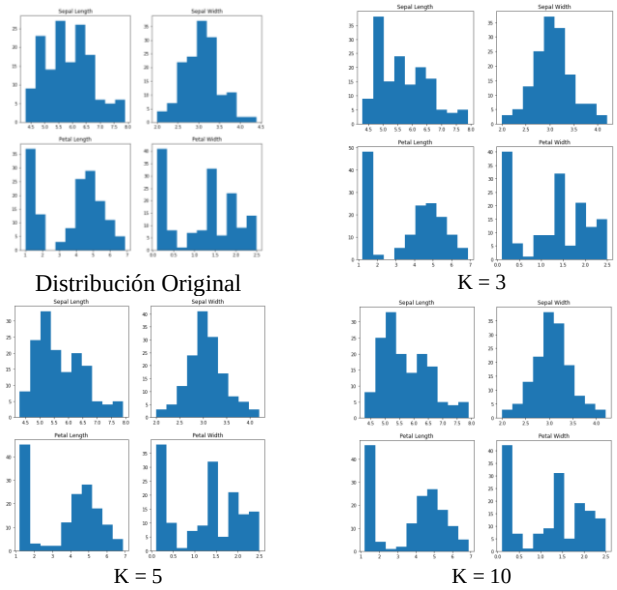


Fig.100. Comparación de la distribución original y las distribuciones con la implementación de la Imputación Vecinos cercanos para K = 3,5,10 cuando se cuenta con el 20% de los valores faltantes.

Sin embargo, el comportamiento mencionado anteriormente cambia al incrementar de nueva cuenta el porcentaje de valores faltantes. En esta ocasión, conforme va aumentando el valor de K, la distribución de este atributo se incrementa de tal forma que se termina obteniendo el mismo valor de tendencia de la distribución de lado izquierdo.

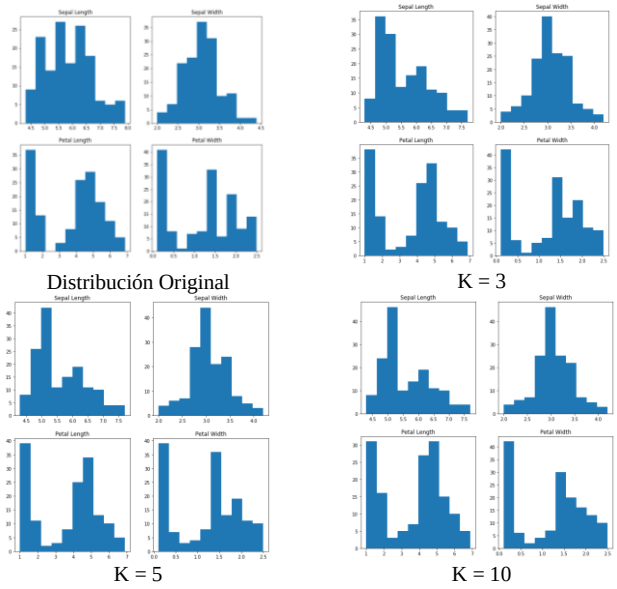


Fig.101. Comparación de la distribución original y las distribuciones con la implementación de la Imputación Vecinos cercanos para K = 3,5,10 cuando se cuenta con el 30% de los valores faltantes.

Así bien, en la figura 102, al incrementar de nueva cuenta el porcentaje de valores faltantes se puede observar como la distribución de la columna ‘Petal length’ ha cambiado en el caso K = 10, ya que únicamente se visualiza un solo valor de tendencia en la distribución.

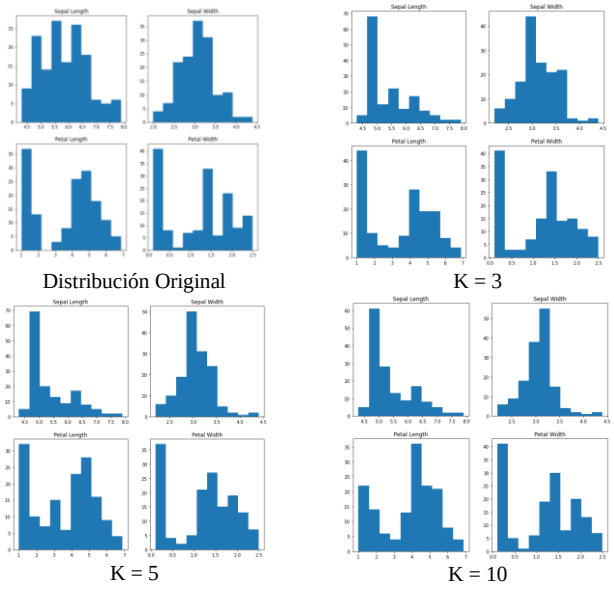


Fig.102. Comparación de la distribución original y las distribuciones con la implementación de la Imputación Vecinos cercanos para K = 3,5,10 cuando se cuenta con el 50% de los valores faltantes.

Imputación Media

En el caso de la Imputación de Media, se puede observae de nueva cuenta como va cambiando la cresta en la distribución del atributo ‘Petal Length’ conforme aumenta el valor de K. Así mismo, en la distribución de la columna ‘Petal Width’, también se puede observar un pequeño cambio en la parte final de la distribución del lado derecho.

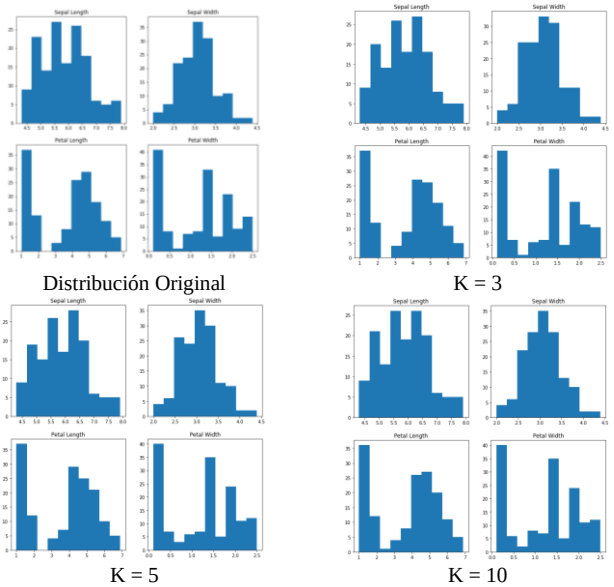


Fig.103. Comparación de la distribución original y las distribuciones con la implementación de la Imputación Media cercanos para K = 3,5,10 cuando se cuenta con el 10% de los valores faltantes.

De igual forma, al incrementar el porcentaje de valores faltantes del conjunto de datos en cuestión, se puede observar en la figura 104 como va cambiando la distribución de los atributos de ‘Sepal Length’, ‘Sepal Width’ y ‘Petal Length’ al implementar la imputación de Media.

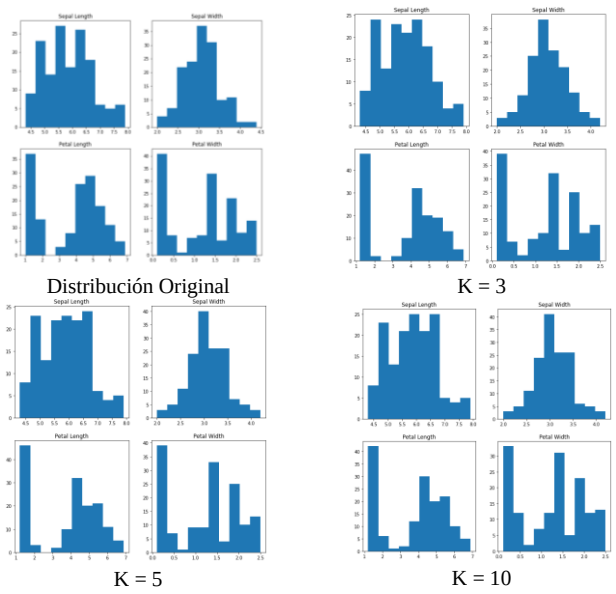


Fig.104. Comparación de la distribución original y las distribuciones con la implementación de la Imputación Media cercanos para K = 3,5,10 cuando se cuenta con el 20% de los valores faltantes.

A diferencia de la Imputación de Vecinos Cercanos, la Imputación de Media si muestra un mayor cambio en la distribución de sus atributos cuando se incrementa el porcentaje de valores faltantes, como se puede observar en las figuras 105 y 106.

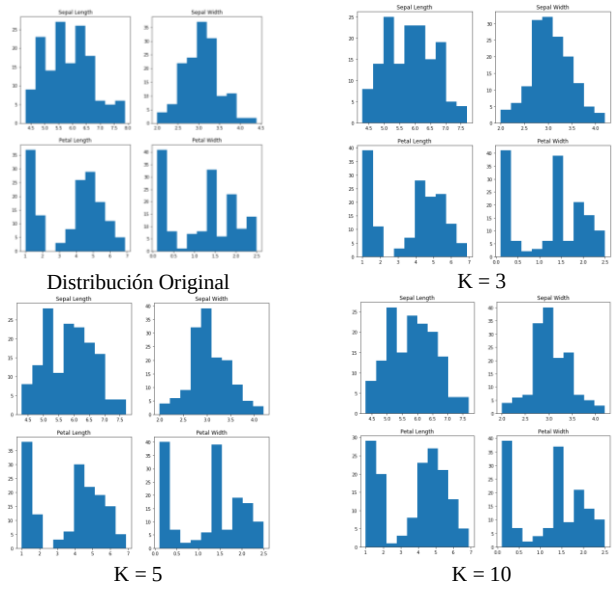


Fig.105. Comparación de la distribución original y las distribuciones con la implementación de la Imputación Media cercanos para K = 3,5,10 cuando se cuenta con el 30% de los valores faltantes.

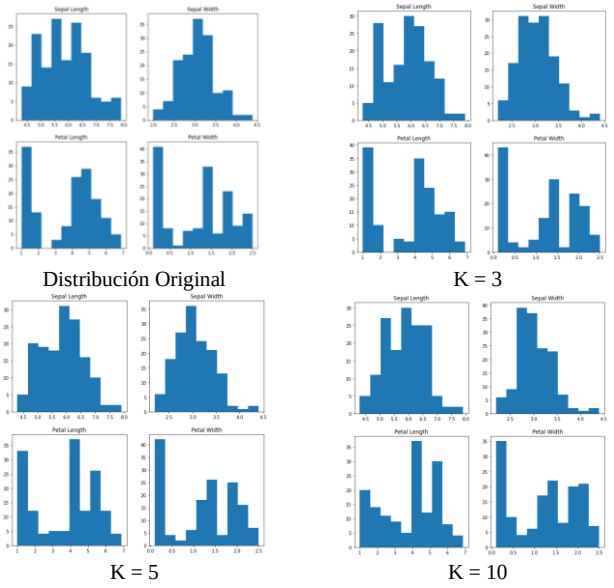


Fig.106. Comparación de la distribución original y las distribuciones con la implementación de la Imputación Media cercanos para K = 3,5,10 cuando se cuenta con el 50% de los valores faltantes.

Imputación Hot Deck

En el caso de la Imputación de Hot Deck, es posible observar en a figura 107 y 108 que al tener el 10% y 20 % de valores faltantes en el conjunto de datos en cuestión la distribución de cada uno de los atributos no varia de forma drástica.

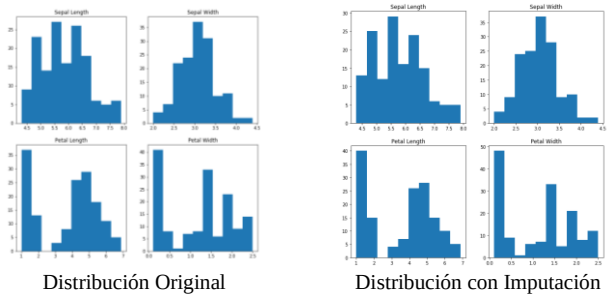


Fig.107. Comparación de la distribución original y las distribuciones con la implementación de la Imputación Hot Deck, cuando se cuenta con el 10% de los valores faltantes.

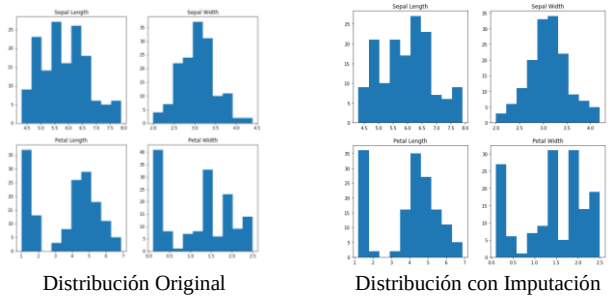


Fig.108. Comparación de la distribución original y las distribuciones con la implementación de la Imputación Hot Deck, cuando se cuenta con el 20% de los valores faltantes.

No obstante, cuando el porcentaje de valores aumenta como es el caso de la figura 109 y 110, se puede visualizar un cambio más notario en la distribución de los datos.

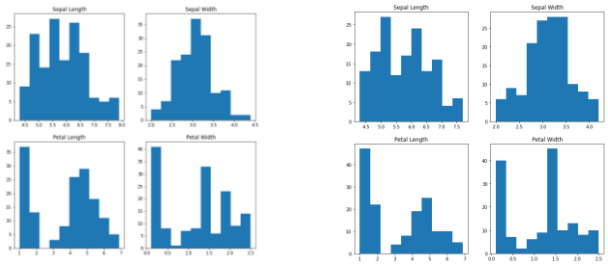


Fig.109. Comparación de la distribución original y las distribuciones con la implementación de la Imputación Hot Deck, cuando se cuenta con el 30% de los valores faltantes.

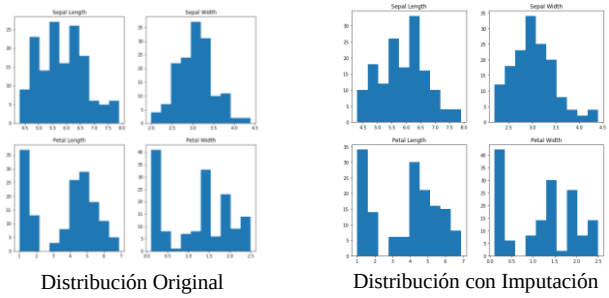


Fig.110. Comparación de la distribución original y las distribuciones con la implementación de la Imputación Hot Deck, cuando se cuenta con el 50% de los valores faltantes.

Técnicas de Normalización

Por otra parte, se decido realizar una comparación de los resultados obtenidos al implementar cada una de las normalizaciones seleccionadas con funciones propias y funciones de la librería de Sklearn, a fin de verificar si se realizó una correcta normalización.

Así bien, únicamente se realizó la normalización para los atributos de 'Petal Length' y 'Petal Width' del conjunto de datos en cuestión, debido a que las distribuciones de estas dos columnas son diferentes a una distribución gaussiana.

- Normalización MinMax

Imputación Vecinos Cercanos

En la figura 111, es posible observar que la distribución obtenida es muy similar tanto para la función propia, como para la librería de Sklearn cuando K tiene valores de 3 y 5. Sin embargo en el caso K = 10, se puede visualizar una diferencia en el atributo 'Petal Length', en específico en la cresta de la figura que se encuentra del lado derecho.

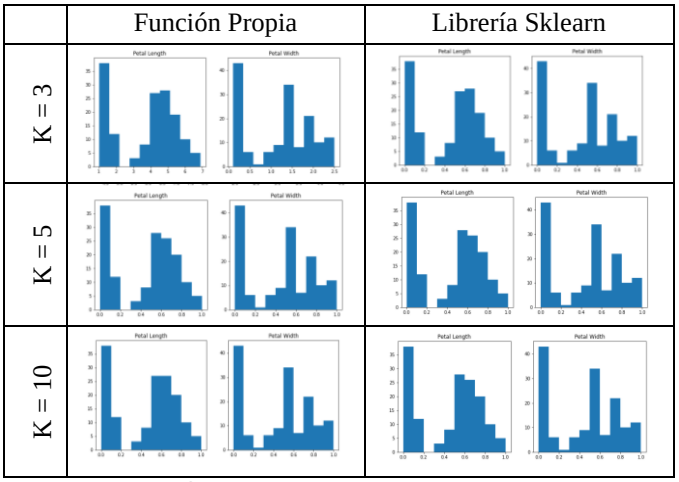


Fig.111.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización MinMax posterior a la Imputación de Vecinos cercanos con el 10% de los valores faltantes.

Por otra parte, al realizar el aumento en el porcentaje de los valores faltantes, se puede observar en la figura 112 y en la figura 113 que a simple vista no se cuenta con alguna diferencia entre las distribuciones presentadas por la función propia y la librería de Sklearn.

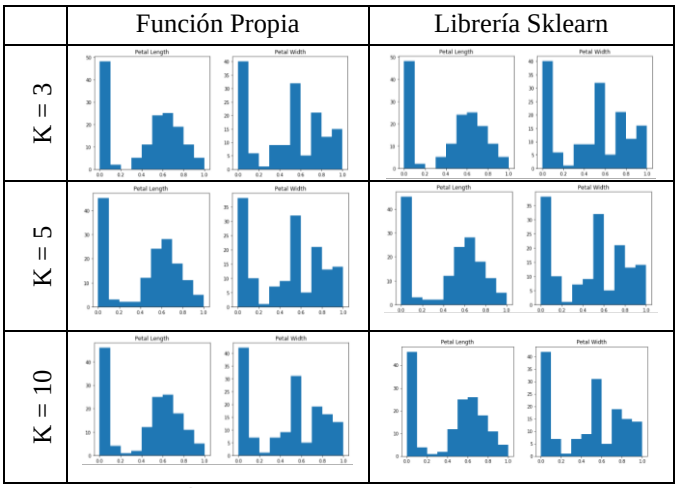
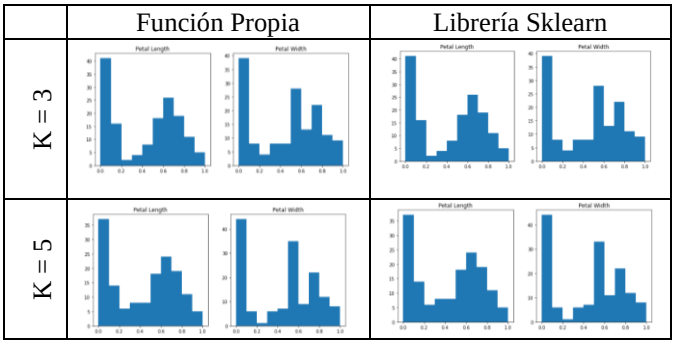


Fig.112.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización MinMax posterior a la Imputación de Vecinos cercanos con el 20% de los valores faltantes



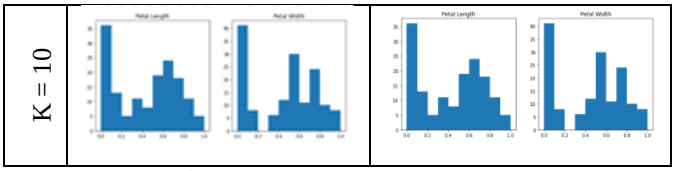


Fig.113.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización MinMax posterior a la Imputación de Vecinos cercanos con el 30% de los valores faltantes

Sin embargo, en la figura 114 es posible notar una diferencia en la distribución de la columna 'Petal Width' cuando K es igual a 5, lo cual no sucede en los otros casos cuando el valor de K es 3 y 10.

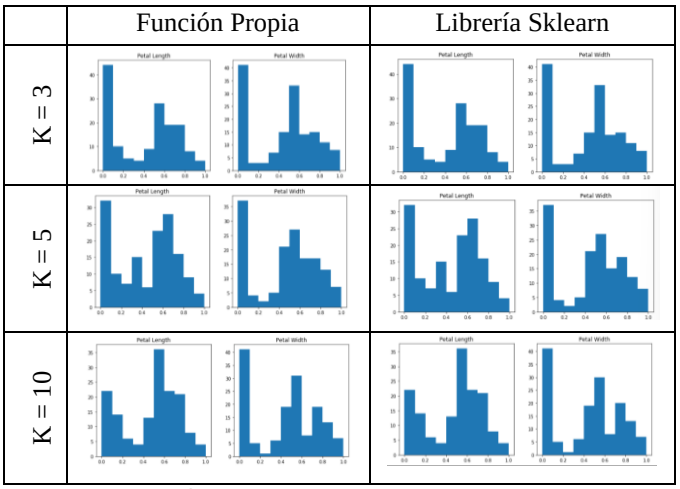


Fig.114.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización MinMax posterior a la Imputación de Vecinos cercanos con el 50% de los valores faltantes

Imputación Media

En la figura 115, es posible observar ciertas diferencias en las distribuciones de los atributos de 'Petal Length' y ' Petal Width' cuando K tiene un valor de 3.

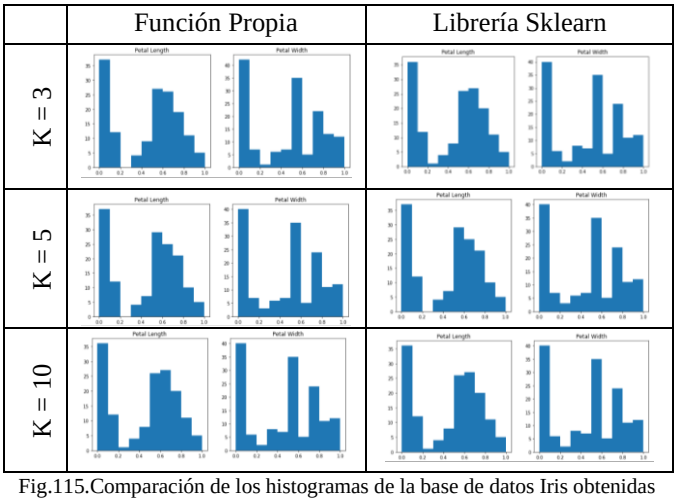


Fig.115.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización MinMax posterior a la Imputación de Media con el 10% de los valores faltantes

Así bien, se puede observar en la figura 116, 117 y 118 que a simple vista que existe una similitud entre las distribuciones obtenidos por la función propuesta, así como con la librería de Sklearn.

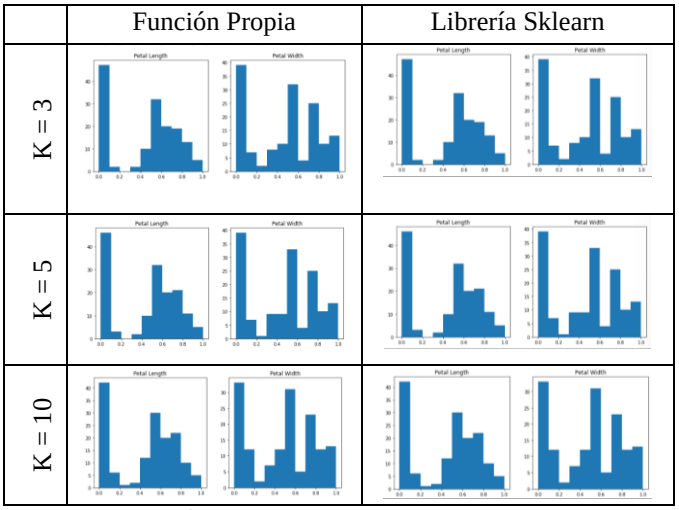


Fig.116.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización MinMax posterior a la Imputación de Media con el 20% de los valores faltantes

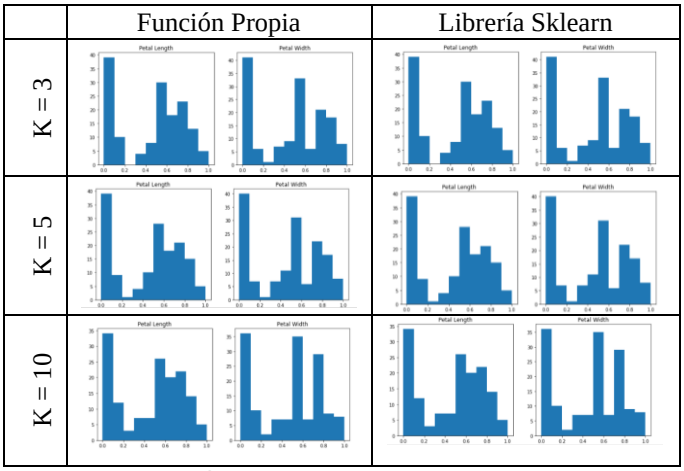
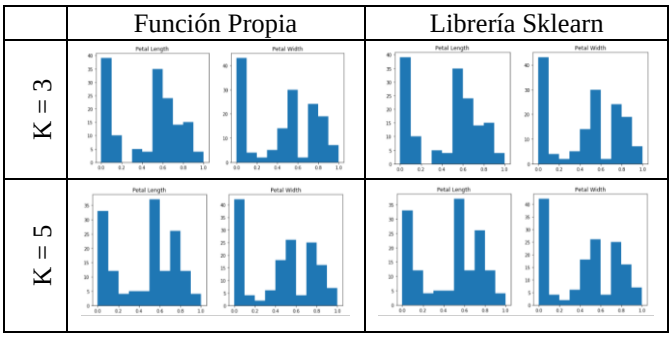


Fig.117.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización MinMax posterior a la Imputación de Media con el 30% de los valores faltantes



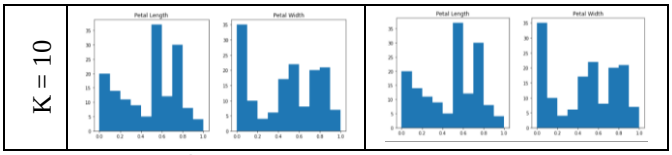


Fig.118.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización MinMax posterior a la Imputación de Media con el 50% de los valores faltantes

Imputación Hot Deck

En la figura 119, no se observa ningún cambio significativo en la distribución por parte de la función propia y la librería de Sklean.

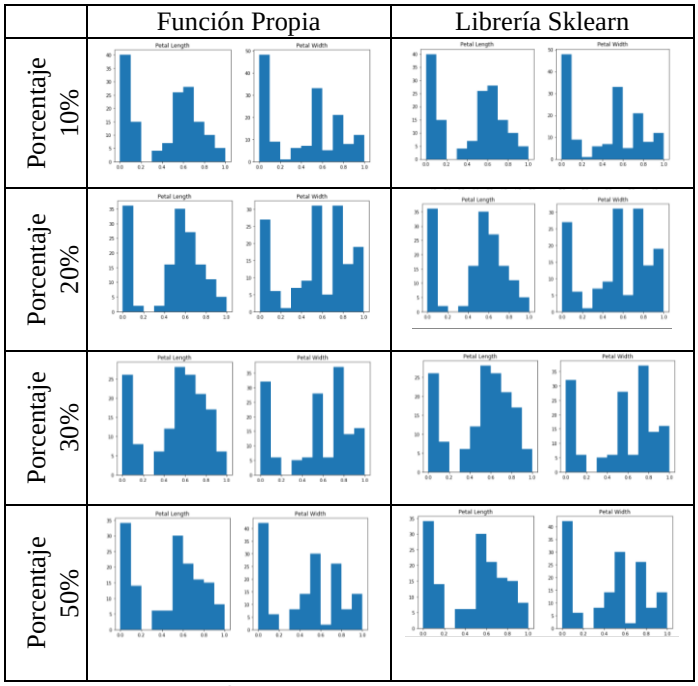
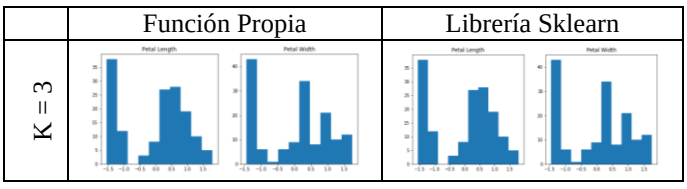


Fig.119.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización MinMax posterior a la Imputación de Hot Deck.

- Normalización puntaje Z

Imputación Vecinos Cercanos

En la figura 120, se puede observar que en el caso en el que K tiene valores de 5 y 10, la distribución obtenida en el atributo 'Petal Width' es diferente entre la función propia y la librería de Sklearn.



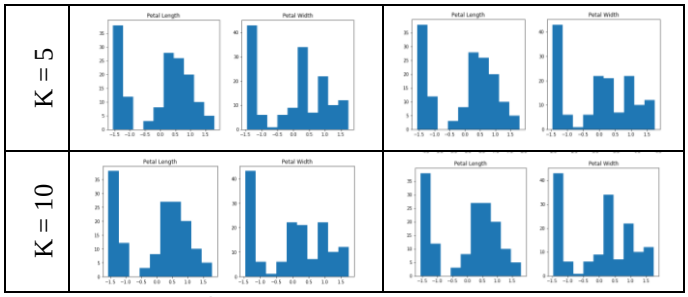


Fig.120.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización z score posterior a la Imputación de Vecinos cercanos con el 10% de los valores faltantes

Así mismo, en la figura 121, 122 y 123 se pueden observar en la columna menciona anteriormente diferencias en las distribuciones obtenidas cuando K tiene valores de 3 y 10.

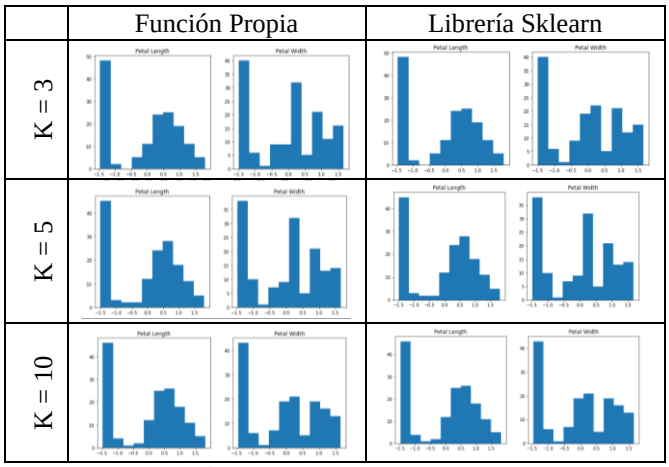


Fig.121.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización z score posterior a la Imputación de Vecinos cercanos con el 20% de los valores faltantes

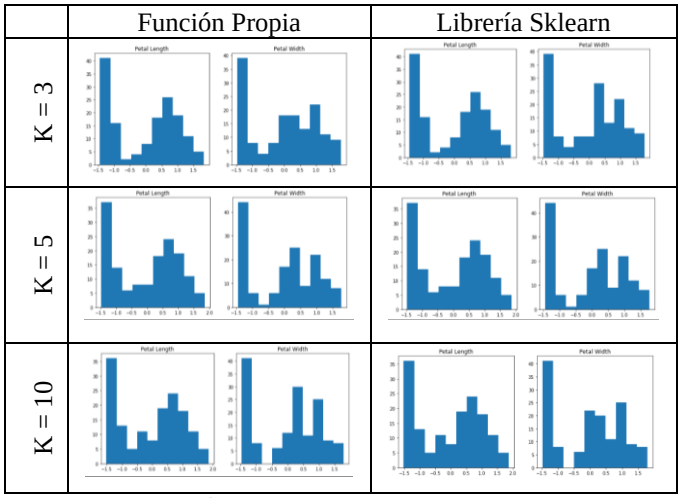


Fig.122.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización z score posterior a la Imputación de Vecinos cercanos con el 30% de los valores faltantes

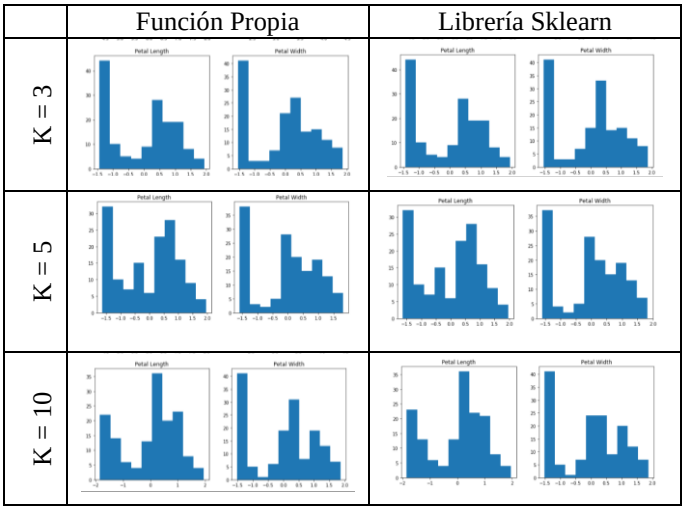


Fig.123.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización z score posterior a la Imputación de Vecinos cercanos con el 50% de los valores faltantes.

Imputación Media

En la figura 124, se puede visualizar diferencias en la distribución del atributo mencionado anteriormente tanto en la función propia, como en la librería de Sklearn cuando K es igual a 3.

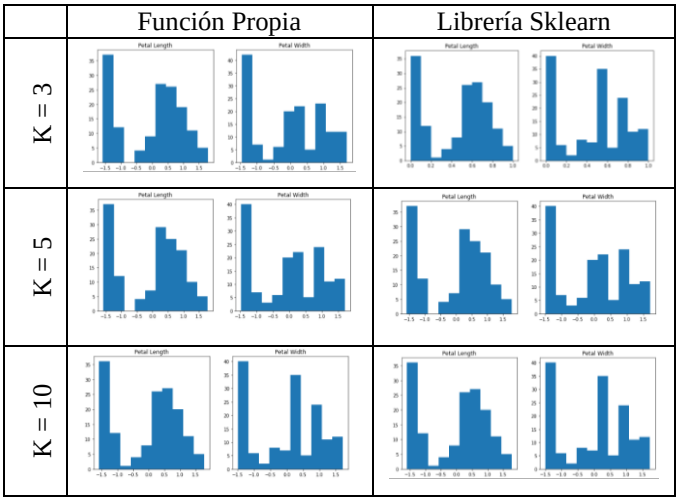


Fig.124.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización z score posterior a la Imputación de Media con 10% de los valores faltantes.

En la figura 125, se puede visualizar diferencias en la distribución del atributo ‘Petal Width’ tanto en la función propia, como en la librería de Sklearn.

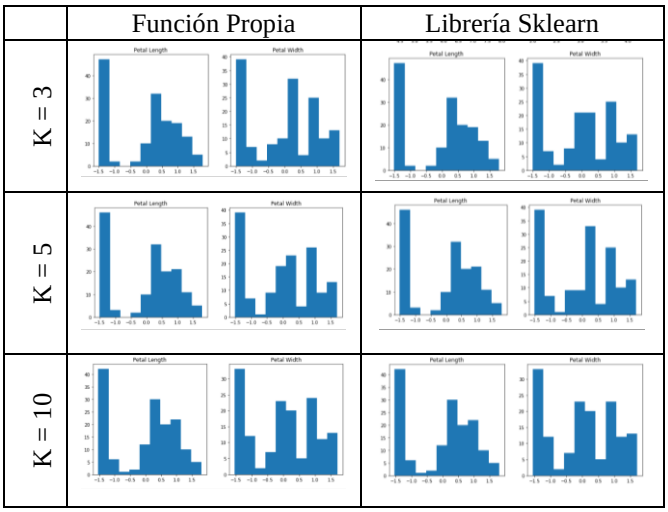


Fig.125.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización z score posterior a la Imputación de Media con 20% de los valores faltantes.

En la figura 126, se puede observar diferencia en la distribución del atributo ‘Petal Width’ en la función propia, como en la librería de Sklearn cuando K es igual a 3 y 5.

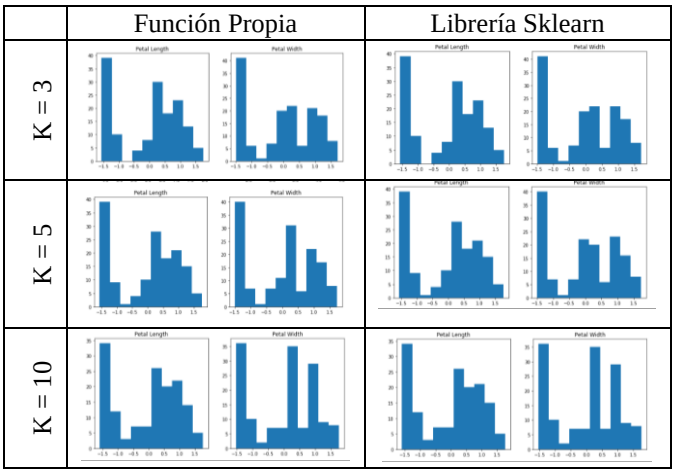


Fig.126.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización z score posterior a la Imputación de Media con 30% de los valores faltantes

Mientras, que en la figura 127, no se observa ninguna diferencia en las distribuciones obtenidas.

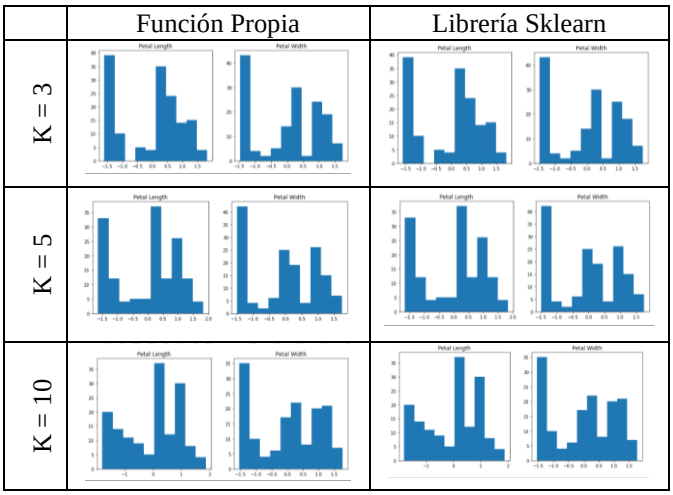


Fig.127.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización z score posterior a la Imputación de Media con 50% de los valores faltantes

Imputación Hot Deck

En la figura 128, se observa una diferencia en la columna 'Petal Width' en los porcentajes de 10%, 20 % y 30%

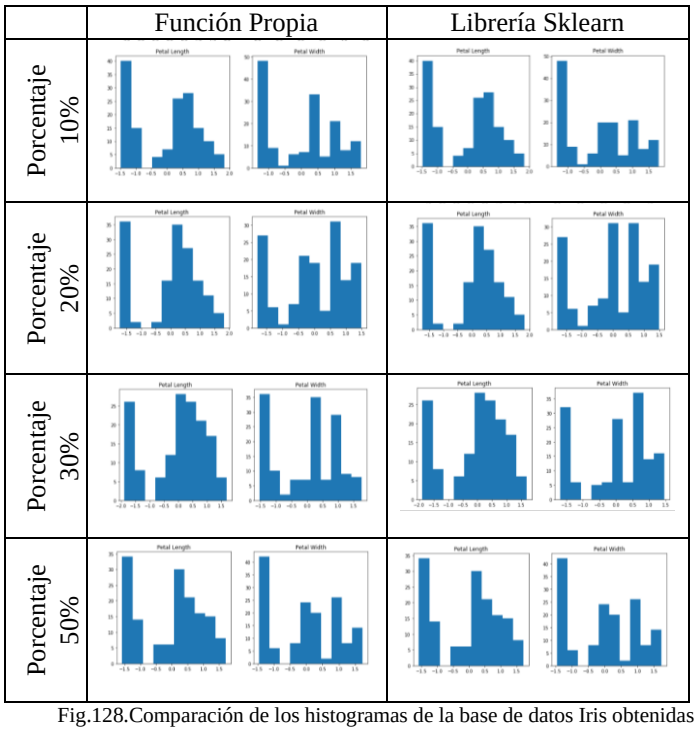


Fig.128.Comparación de los histogramas de la base de datos Iris obtenidas al utilizar la función propia y la librería Sklearn con respecto a la técnica de normalización z score posterior a la Imputación de Hot Deck.

VI. Conclusiones

El desarrollo de esta tarea se enfocó en el análisis de las diversas técnicas de imputación y normalización implementadas en el conjunto de datos propuesto, en donde fue necesario de herramientas de visualización, como son los histogramas a fin de poder obtener una mejor compresión de los resultados obtenidos.

Sin embargo, la realización de un código general para realizar alguna de estas técnicas trae consigo ciertos retos, ya que se deben contemplar diversos escenarios debido a que cada base de datos es diferente entre sí, es decir, puede contener elementos que terminen afectando su correspondiente calidad, como es el caso de los valores faltantes.

Por lo cual, es necesario como primer paso el realizar un análisis de los valores faltantes antes de aplicar cualquier tipo de procedimiento sobre los datos para etapas posteriores, en donde se puede observar el comportamiento que presentan las distribuciones antes y después de implementar algunas de las técnicas de imputación, con el fin de observar cual técnica mantiene en cierto sentido la forma de la distribución original del conjunto de datos en cuestión.

En el caso de la presente base de datos se pudo observar que se tuvo un mejor comportamiento al utilizar la imputación vecinos cercanos, en comparación de las otras dos imputaciones, debido a que hubo una menor variación en las distribuciones obtenidas.

Así bien, otro punto importante a considerar es el tema de la normalización, ya que no todos los valores de tipo cuantitativo requieren de un proceso de normalización, debido a que es necesario considerar desde el rango en cual se encuentran los valores hasta el tipo de algoritmo a implementar.

VII. Referencias

[1] 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

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

[3] Scikit-learn(s.f) The Iris Dataset. Recuperado el día 19 de Agosto de 2022

[4] Aceves-Fernández, M.A. (2021). Inteligencia Artificial para programadores con prisa. Kindle.

[5] F. Medina y M. Galván(2007) Estudios estadísticos y prospectivos: Imputación de datos: teoría y práctica.

[6] Valores faltantes (s.f) Recuperado el día 16 de Septiembre de 2022 de https://www.uv.es/webgid/Descriptiva/23_valores_faltantes.html#:~:text=Los%20m%C3%A9todos%20de%20imputaci%C3%B3n%20consisten,de%20algunas%20variables%20especialmente%20seleccionadas.

[7] Brita. Inteligencia Artificial (2021) Las seis técnicas principales utilizadas en la Ingeniería de Machine Learning. Recuperado el día 17 de Septiembre de 2022 de https://brita.mx/las-seis-tecnicas-principales-utilizadas-en-la-ingenieria-de-machine-learning

[8] Universo Machine Learning (2017) ¿Qué hacer cuando tenemos valores faltantes? Recuperado el día 17 de Septiembre de 2022 de https://conocemachinelearning.wordpress.com/tag/imputacion/

[9] Equipo editorial (2020) ¿Cómo, cuándo y por qué debería normalizar/estandarizar/reescalar sus datos? Recuperado el día 27 de Septiembre de 2022 de https://towardsai.net/p/data-science/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff