**Pasos para implementar un modelo**

**Primero:** Importar las librerías necesarias

* import **pandas** as pd
* from **sklearn** import metrics
* import **matplotlib.pyplot** as plt
* from **sklearn.model\_selection** import **train\_test\_split** #Para separar nuestro dataset
* from **sklearn.linear\_model** import **LogisticRegression**
* import **seaborn** as sns

**Segundo:** Cargamos el dataset con el que vamos a trabajar.

dataset = **pd.read.csv**(‘diabetes.csv’)

**Segundo.5 :** Preparar los datos.

feature\_cols = [ 'Pregnancies', 'Insulin', 'BMI', 'Glucose', 'BloodPressure','DiabetesPedigreeFunction' ]

**Notita:** “.iloc” es para localizar cuáles son los datos que debemos utilizar para entrenar

**X** = **diabetes**[ feature\_cols ] #Con values estamos accediendo a los valores

**Y** = **diabetes.**Outcome

**Tercero:** Pasamos a separar los datos para entrenamiento y verificación

**X\_train**, X\_test, **Y\_train**, Y\_test = **train\_test\_split(X , Y , test\_size =0,2, random\_state =0)**

**Cuarto:** Pasamos a preparar nuestro modelo logístico.

logreg = **LogisticRegression( )**

**Quinto:** Pasamos a entrenar nuestro modelocon **“.fit”**

logreg**.fit**(**X\_train, Y\_train**)

**Sexto:** Procedemos a mostrar el resultado del entrenamiento

y\_pred = logreg.predict(X\_test)

**Matriz de confusión, pero en un arreglo simple**

cnf\_matrix = metrics.confusion\_matrix(Y\_test, y\_pred)

**Ahora pasamos a graficar los resultados en un heatmap con seaborn**

class\_name = [0,1]

fig, ax = plt.subplots()

tick\_marks = np.arange(len(class\_name))

plt.xticks(tick\_marks, class\_name)

plt.yticks(tick\_marks, class\_name)

sns.heatmap(pd.DataFrame(cnf\_matrix), annot=True, cmap = 'Blues\_r', fmt = 'g' )

ax.xaxis.set\_label\_position('top')

plt.title('Matriz de Confusiónn', y=1.1)

plt.xlabel('Valor real')

plt.ylabel('Valor estimado')

**random.randint(Start, Stop):** escoge aleatoriamente un núemro entre Start y Stop

**np.random.randn(10) :** Crea una lista de 10 números aleatorios entre [0, 1]

**Dato:** Si queremos agregar una fila adicional a nuestro data o archivo .csv

datos = pd.read\_csv(‘energy.csv’ , header = ‘0’)

Una opción es