reporte-1

September 12, 2023

1 1. Dataset

Características por las que se eligió el set de datos:

1. **Variedad de características**: El set de datos cuenta con 9 columnas que representan una característica diferente, lo que le permite al árbol de decisión poder elegir entre más opciones para separar los datos, brindando un mejor ajuste y desempeño general.
2. **Tamaño**: Eliminando los valores faltantes, el set de datos cuenta con una longitud de 2011 registros, lo cual es suficiente para que el árbol pueda generalizar de forma correcta y obtener resultados decentes.
3. **Predicción de clases**: Los árboles de decisión son muy buenos para clasificar y la columna de salida del set de datos es una clase binaria (potable, no potable).
4. **Fuente confiable**: El set de datos fue obtenido de Kaggle, la cual es una plataforma conocida que cuenta con una gran variedad de datasets de calidad para diferentes aplicaciones.

[64]:
```
from typing_extensions import dataclass_transform
import pandas as pd

# Cargar datos
data = pd.read_csv("diabetes.csv")
# Eliminar duplicados y valores faltantes
data = data.drop_duplicates()
data = data.dropna()

#One-hot encode the "gender" and "smoking_history" columns, creating dummy␣
 ↪variables
dummies = pd.get_dummies(data[["gender", "smoking_history"]])
data = data.drop(["gender", "smoking_history"], axis=1)

# Observations and target data
data = pd.concat([data, dummies], axis=1)

print(data.head())
print("Filas: ", data.shape[0])
print("Columnas: ", data.shape[1])
```

```
    age  hypertension  heart_disease    bmi  HbA1c_level  blood_glucose_level  \
0  80.0             0              1  25.19          6.6                  140
1  54.0             0              0  27.32          6.6                   80
```

```
2  28.0             0              0  27.32          5.7                  158
3  36.0             0              0  23.45          5.0                  155
4  76.0             1              1  20.14          4.8                  155

   diabetes  gender_Female  gender_Male  gender_Other  \
0         0              1            0             0
1         0              1            0             0
2         0              0            1             0
3         0              1            0             0
4         0              0            1             0

   smoking_history_No Info  smoking_history_current  smoking_history_ever  \
0                        0                        0                     0
1                        1                        0                     0
2                        0                        0                     0
3                        0                        1                     0
4                        0                        1                     0

   smoking_history_former  smoking_history_never  smoking_history_not current
0                       0                      1                            0
1                       0                      0                            0
2                       0                      1                            0
3                       0                      0                            0
4                       0                      0                            0
Filas:  96146
Columnas:  16
```

2 2. Separación y evaluación del modelo.

Para separar los datos en entrenamiento y validación se utilizó la proporción de 80% entrenamiento y 20% pruebas. Esta proporción es una de las más utilizadas, ya que está comprobado que para sets de datos medianos, el modelo generaliza de mejor manera y se garantiza un equilibrio adecuado entre la evaluación y el entrenamiento del modelo.

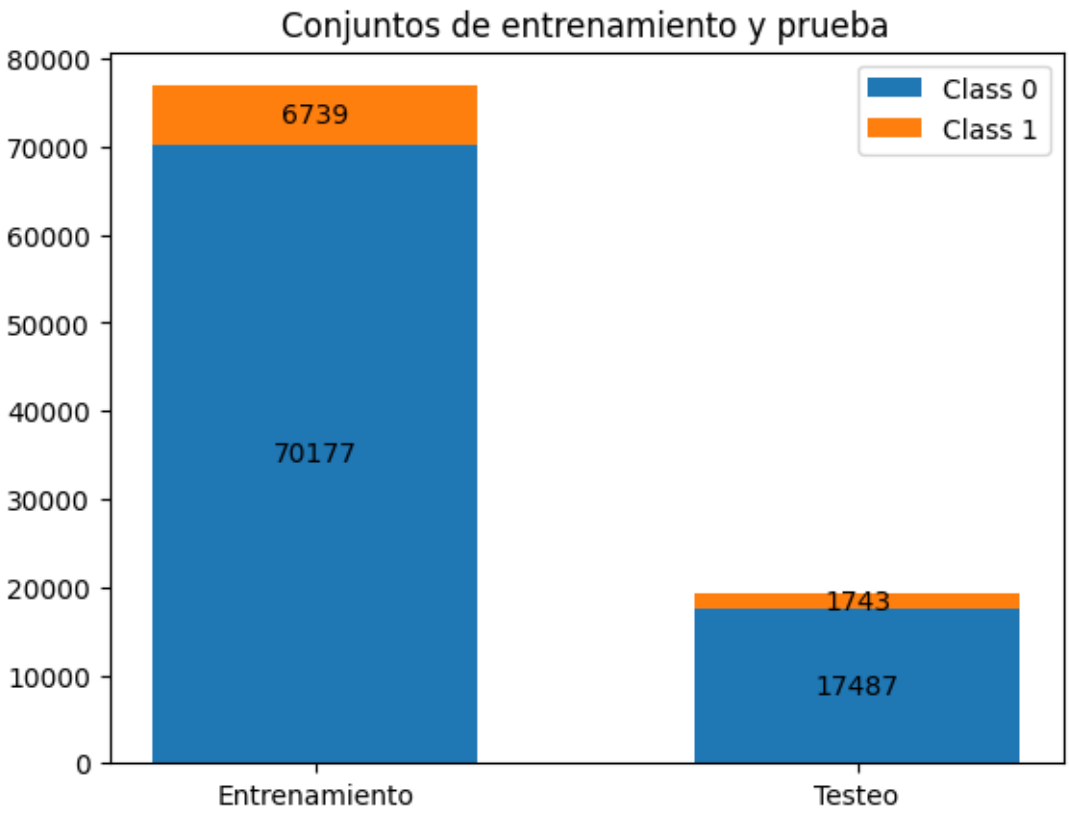
Esta selección de datos se realiza de forma aleatoria para evitar sesgos al momento de elegir los datos, por lo que los índices que se muestran en las cabeceras de los datos no están en algún orden particular.

Al elegir una proporción de 80 - 20 y tener un set de datos de 2011 filas, se espera que la longitud 1608 renglones de entrenamiento y 403 renglones de testeo.

En la gráfica se puede observar que se respeta la proporción de 80 - 20, mientras que se muestra el número de elementos que pertenecen a cada clase. Mientras cada clase tenga una cantidad similar de datos, el modelo está mejor balanceado. Balancear el número de clases puede ser una herramienta para mejorar el desempeño del modelo.

[65]:
```
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import numpy as np
```

```
# Dividir entradas y salida
X = data.drop(["diabetes"], axis=1)
y = data["diabetes"]

# Dividir datos en train y test con un ratio de 80% - 20%
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Gráfica
# Datos de gráfica
vals = pd.concat([y_train.value_counts(), y_test.value_counts()], axis=1).
 ↪to_numpy()
splits = ('Entrenamiento', 'Testeo')
split_count = {
    'Class 0': vals[0],
    'Class 1': vals[1],
}

# Visualización
width = 0.6
fig, ax = plt.subplots()
bottom = np.zeros(2)
for c, c_count in split_count.items():
    p = ax.bar(splits, c_count, width, label=c, bottom=bottom)
    bottom += c_count

    ax.bar_label(p, label_type='center')
ax.set_title("Conjuntos de entrenamiento y prueba")
ax.legend()

plt.show()

print()
print("----------- Entrenamiento -----------")
print("Entrada: ")
print(X_train.head())
print("Salida: ")
print(y_train.head())
print("Número de renglones: ", X_train.shape[0])
print()
print("--------------- Testeo --------------")
print("Entrada: ")
print(X_test.head())
print("Salida: ")
print(y_test.head())
print("Número de renglones: ", X_test.shape[0], end='\n')
```




```
---------- Entrenamiento ----------
Entrada:
        age  hypertension  heart_disease    bmi  HbA1c_level  \
64182  80.0             0              1  23.36          4.0
99842  51.0             0              0  27.32          4.0
82761  53.0             0              0  27.32          5.7
37231  31.0             0              0  41.97          5.7
12492  25.0             0              0  27.32          6.1

       blood_glucose_level  gender_Female  gender_Male  gender_Other  \
64182                  160              1            0             0
99842                   80              0            1             0
82761                  130              0            1             0
37231                  126              1            0             0
12492                  159              0            1             0

       smoking_history_No Info  smoking_history_current  smoking_history_ever  \
64182                        0                        0                     0
99842                        1                        0                     0
82761                        0                        1                     0
```

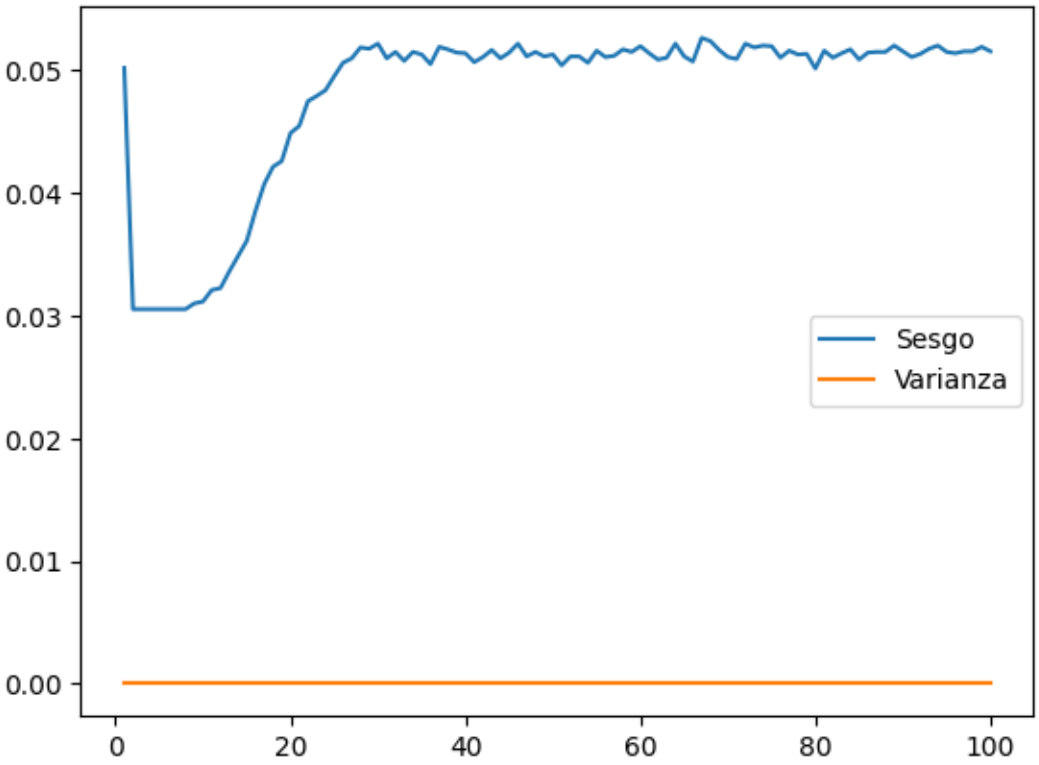
```
37231                         0                         1                         0
12492                         0                         0                         0

       smoking_history_former  smoking_history_never  \
64182                       0                      1
99842                       0                      0
82761                       0                      0
37231                       0                      0
12492                       0                      1

       smoking_history_not current
64182                            0
99842                            0
82761                            0
37231                            0
12492                            0
Salida:
64182    0
99842    0
82761    0
37231    0
12492    0
Name: diabetes, dtype: int64
Número de renglones:  76916

--------------- Testeo --------------
Entrada:
        age  hypertension  heart_disease    bmi  HbA1c_level  \
83417  69.0             0              0  27.32          6.0
12564  19.0             0              1  40.25          6.1
3000   53.0             0              0  20.97          6.0
17901  51.0             0              0  37.65          6.2
63287  36.0             0              0  22.83          6.5

       blood_glucose_level  gender_Female  gender_Male  gender_Other  \
83417                  158              0            1             0
12564                  140              1            0             0
3000                   140              1            0             0
17901                  155              1            0             0
63287                  200              1            0             0

       smoking_history_No Info  smoking_history_current  smoking_history_ever  \
83417                        0                        0                     0
12564                        0                        0                     0
3000                         1                        0                     0
17901                        0                        1                     0
63287                        0                        0                     0
```

```
       smoking_history_former  smoking_history_never  \
83417                       0                      1
12564                       0                      0
3000                        0                      0
17901                       0                      0
63287                       0                      1

       smoking_history_not current
83417                            0
12564                            1
3000                             0
17901                            0
63287                            0
Salida:
83417    0
12564    0
3000     0
17901    0
63287    0
Name: diabetes, dtype: int64
Número de renglones:  19230
```

3 3. Sesgo y varianza

[100]:
```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, f1_score,␣
 ↪confusion_matrix
from mlxtend.evaluate import bias_variance_decomp
import matplotlib.pyplot as plt

bias = np.zeros(100)
variance = np.zeros(100)
error = np.zeros(100)

# Obtener sesgo y varianza actualizando hiperparámetros
for i in range(100):
    # Árbol de decision
    tree = DecisionTreeClassifier(
        criterion="entropy",
        max_depth=i + 1
    )

    # Obtener el sesgo y varianza promedio de 200 corridas diferentes
    avg_expected_loss, avg_bias, avg_var = bias_variance_decomp(
            tree, X_train.to_numpy(), y_train.to_numpy(), X_test.to_numpy(),␣
 ↪y_test.to_numpy(),
```

```
            loss='0-1_loss',
            num_rounds=1,
            random_seed=123)

    bias[i] = avg_bias
    variance[i] = avg_var
    error[i] = avg_expected_loss

# Mostrar la relación del sesgo y varianza en 200 corridas diferentes
plt.plot(np.linspace(1,100, 100), bias, label = "Sesgo")
plt.plot(np.linspace(1,100, 100), variance, label = "Varianza")
plt.legend()
plt.show()

# Modelo previamente construido
tree = DecisionTreeClassifier(
    criterion="entropy",
    max_depth=8
)
# Obtener el sesgo y varianza promedio de 200 corrridas iguales
avg_expected_loss, avg_bias, avg_var = bias_variance_decomp(
    tree, X_train.to_numpy(), y_train.to_numpy(), X_test.to_numpy(), y_test.
 ↪to_numpy(),
    loss='0-1_loss',
    num_rounds=200,
    random_seed=123)

print()
print("Sesgo y varianza promedio de 200 entrenamientos")
print('Sesgo promedio: %.3f' % avg_bias)
print('Varianza promedio: %.3f' % avg_var)
```




```
Sesgo y varianza promedio de 200 entrenamientos
Sesgo promedio: 0.031
Varianza promedio: 0.001
```

4 5. Nivel de ajuste del modelo

[ ]:

5 6. Técnicas de regularización o ajuste de parámetros

[ ]: