Laboratorio 7

Elisa Samayoa y Julio Avila

1. Carga del Conjunto de Datos:
    a. Crear un DataFrame con el contenido del archivo `diabetes.csv

```
[4] diabetes = pd.read_csv('diabetes.csv')
    diabetes
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...

2. Análisis Exploratorio de Datos (EDA):

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Hay un total de 768 observaciones, con 9 variables, de las cuales 7 son discretas y 2 son continuas.

a. Obtener estadísticas descriptivas básicas del conjunto de datos.

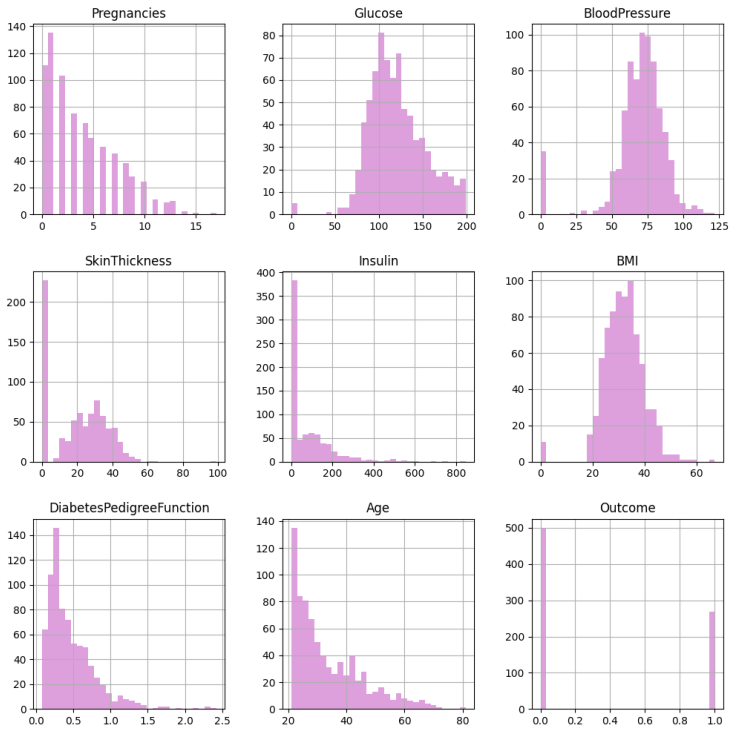
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin
count	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479
std	3.369578	31.972618	19.355807	15.952218	115.244002
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000
75%	6.000000	140.250000	80.000000	32.000000	127.250000
max	17.000000	199.000000	122.000000	99.000000	846.000000

	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000
mean	31.992578	0.471876	33.240885	0.348958
std	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.078000	21.000000	0.000000
25%	27.300000	0.243750	24.000000	0.000000
50%	32.000000	0.372500	29.000000	0.000000
75%	36.600000	0.626250	41.000000	1.000000
max	67.100000	2.420000	81.000000	1.000000

Como se puede observar, la edad promedio en los registros es de 33 años, el más joven es de 21 años y el más grande de 81. En promedio, se tienen entre 3 y 4 embarazos, con niveles de glucosa de 120 y una desviación del 31.97. La presión se encuentra, en promedio, en los 69, el máximo registro es de 122, pero el 50% de los datos se concentra en los 72. La mitad de los registros tiene un nivel de insulina del 30.50, mientras que el máximo es de

846. Para el espesor de la piel el promedio es de 20.53, con un máximo de 99 y mínimo de 0. El BMI se encuentra con un promedio de 31.99 y un máximo de 67. Para la variable Diabetespedigreefunction se tiene un mínimo de 0.07 y máximo de 2.42.

b. Visualizar la distribución de las variables.



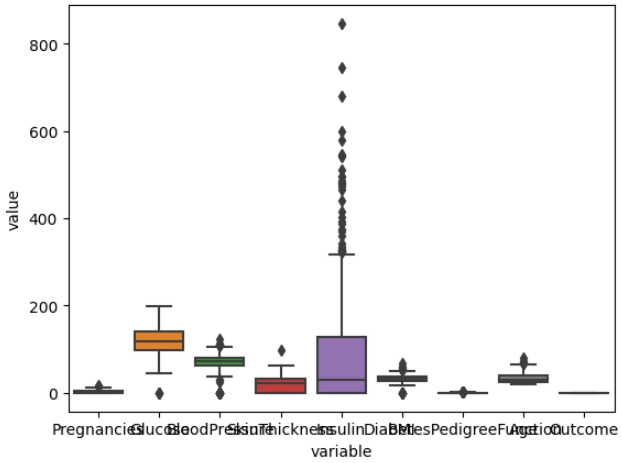
La variable pregnancies tiene un distribución asimétrica positiva, al igual que la de age, diabetespredigreefunction e insulin. La variable de glucose, bloodpressure, bmi y skinthickness tienen una distribución casi normal con unos datos atípicos hacia la izquierda.

c. Verificar la presencia de valores nulos o atípicos y decidir cómo manejarlo
   i. No se encontraron valores nulos en la base de datos

```
diabetes.isnull().sum()
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

ii. En la variable de Insuline es donde se pudieron observar más puntos atípicos.

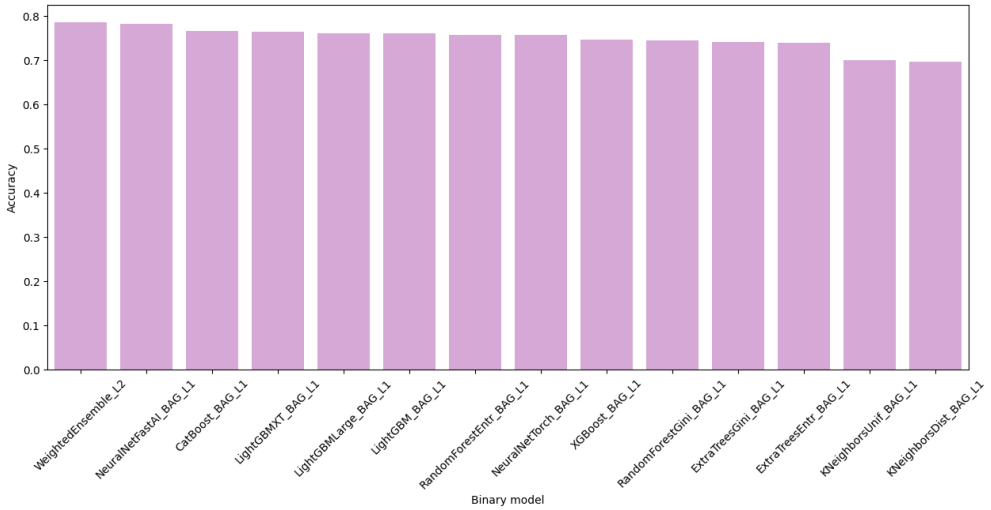


3. Entrenamiento con AutoGluon:
    a. Utilizar la opción `best_quality` de preset y la métrica `accuracy`.
    b. Entrenar modelos de clasificación con AutoGluon para predecir la columna “Outcome”

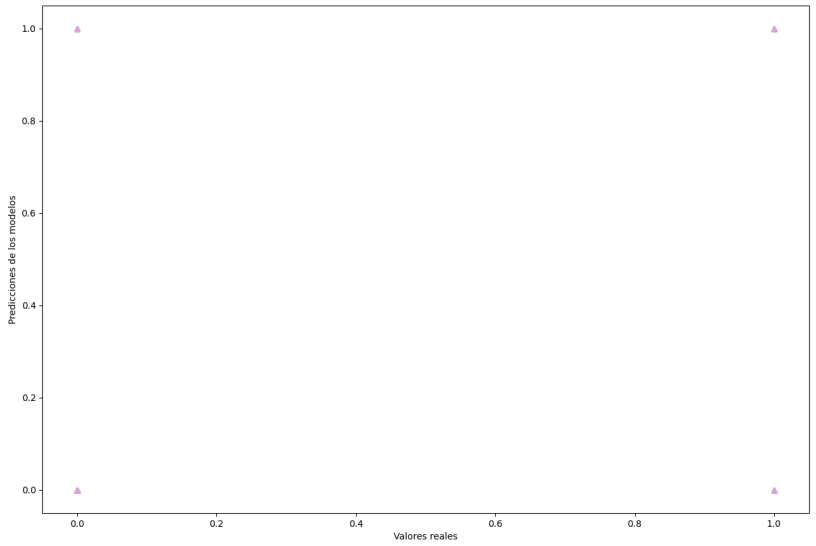
```
predictor = TabularPredictor(label="Outcome",
                             problem_type = 'binary',
                             eval_metric = 'accuracy').fit(train_data = X_entreno,
                                                           time_limit = 300,
                                                           presets = "best_quality")
```

4. Evaluación del modelo
    a. Desplegar una tabla de los mejores modelos (leaderboard) en función de la precisión y destacar el mejor modelo.

index	model	score_val	pred_time_val	fit_time	pred_time_val_marginal	fit_time_marginal	stack_level	can_infer
0	WeightedEnsemble_L2	0.7866449511400652	1.0015275478363037	330.7477402687073	0.004218339920043945	1.8440649509429932	2	true
1	NeuralNetFastAI_BAG_L1	0.7833876221498371	0.8035032749176025	155.32581114768982	0.8035032749176025	155.32581114768982	1	true
2	CatBoost_BAG_L1	0.7671009771986971	0.04957151412963867	76.56884670257568	0.04957151412963867	76.56884670257568	1	true
3	LightGBMXT_BAG_L1	0.7654723127035831	0.14423441886901855	97.00901746749878	0.14423441886901855	97.00901746749878	1	true
4	LightGBMLarge_BAG_L1	0.762214983713355	0.16687822341918945	93.30647540092468	0.16687822341918945	93.30647540092468	1	true
5	LightGBM_BAG_L1	0.760586319218241	0.10971236228942871	82.76700282096863	0.10971236228942871	82.76700282096863	1	true
6	RandomForestEntr_BAG_L1	0.757328990228013	0.25595927238464355	1.8068420886993408	0.25595927238464355	1.8068420886993408	1	true
7	NeuralNetTorch_BAG_L1	0.757328990228013	0.7261345386505127	125.50350689888	0.7261345386505127	125.50350689888	1	true
8	XGBoost_BAG_L1	0.747557003257329	0.4472329616546631	62.97437000274658	0.4472329616546631	62.97437000274658	1	true
9	RandomForestGini_BAG_L1	0.745928338762215	0.2919771671295166	1.5675878524780273	0.2919771671295166	1.5675878524780273	1	true



    b. Visualizar la matriz de confusión del mejor modelo




```
RMSE = 0.449
MSE = 0.2012987012987013
MAE = 0.2012987012987013
R2 = 0.05070590574666933
```

5. Redactar sus reflexiones sobre su experiencia en la última semana en este curso y su opinión sobre el paquete AutoGluon. ¿Qué ventajas y desventajas pueden verse en este tipo de herramientas de "AutoML"?

   El contenido de esta última semana ha sido bastante interesante. En el curso de Data Mining y Machine Learning pudimos ver un poco acerca de estas predicciones, pero en R, por lo que aprenderlas en python fue más ventajoso. El tema de la última clase para hacer los modelos también ha sido de gran utilidad, sobre todo porque

   muchas veces no sabemos cómo llevarlos en práctica más allá de lo visto en clase. En cuanto al paquete de AutoGluon es bastante fácil de implementar, solo es necesario conocer bien los datos que se tienen para definir cuáles son los parámetros que se desean y así obtener pronósticos más cercanos.

   Una de las principales ventajas de este tipo de herramientas es que nos podemos ahorrar bastante tiempo al implementarlas, que son fáciles de acceder y que, si en dado caso no sabemos qué parámetros son mejor para nuestro modelo, estas herramientas buscan la manera de adaptarse al mismo, por lo que evita ese proceso de estar “jugando con los resultados”

   Sin embargo, una gran desventaja es que si se desean personalizar más los modelos, esto no es posible, ya que hay pocas opciones que uno puede manejar. Por lo que, en ciertos casos, si algo no funciona o no llega como nos gustaría, lo mejor sería buscar otra opción. Además, se puede ofrecer opciones limitadas a modelos, algo que sería distinto si nosotros los programáramos de acuerdo a lo que se adaptara mejor.