

# Laboratorio 7

Rodrigo Mansilla

Brandon Reyes

## Inicialización de Librerías

## Carga del Dataset

Out[ ]:	<b>Pregnancies</b>	<b>Glucose</b>	<b>BloodPressure</b>	<b>SkinThickness</b>	<b>Insulin</b>	<b>BMI</b>	<b>DiabetesPedigreeFunc</b>
0	6	148	72	35	0	33.6	0
1	1	85	66	29	0	26.6	0
2	8	183	64	0	0	23.3	0
3	1	89	66	23	94	28.1	0
4	0	137	40	35	168	43.1	1

## Variables del dataset ( diabetes.csv ):

- **Pregnancies:** Número de veces que la paciente ha estado embarazada.
- **BloodPressure:** Presión arterial diastólica (mm Hg).
- **Skin:** Grosor del pliegue cutáneo del tríceps (mm).
- **Insulin:** Nivel de insulina sérica a las 2 horas (mu U/ml).
- **BMI:** Índice de masa corporal (peso en kg / altura en m<sup>2</sup>).
- **DiabetesPedigreeFunction:** Función de historial familiar de diabetes (mide predisposición hereditaria).
- **Age:** Edad de la persona (en años).
- **Outcome:** Variable objetivo.
  - 0 : El paciente **no presenta diabetes**.
  - 1 : El paciente **presenta diabetes**.

## Contexto

El dataset busca predecir la presencia de **diabetes tipo II** en función de variables clínicas y características personales.

La columna **Outcome** es el objetivo de clasificación binaria:

- **0 = No diabetes,**
- **1 = Con diabetes.**

# Análisis Exploratorio de Datos (EDA):

Se realiza un análisis exploratorio para comprender mejor las características del dataset:

1. Estadísticas descriptivas.
2. Distribuciones de las variables (histogramas y boxplots).
3. Presencia de valores nulos y atípicos.
4. Balance de clases de la variable *Outcome*.
5. Correlaciones entre variables y mapa de calor.

## 1. Estadísticas descriptivas

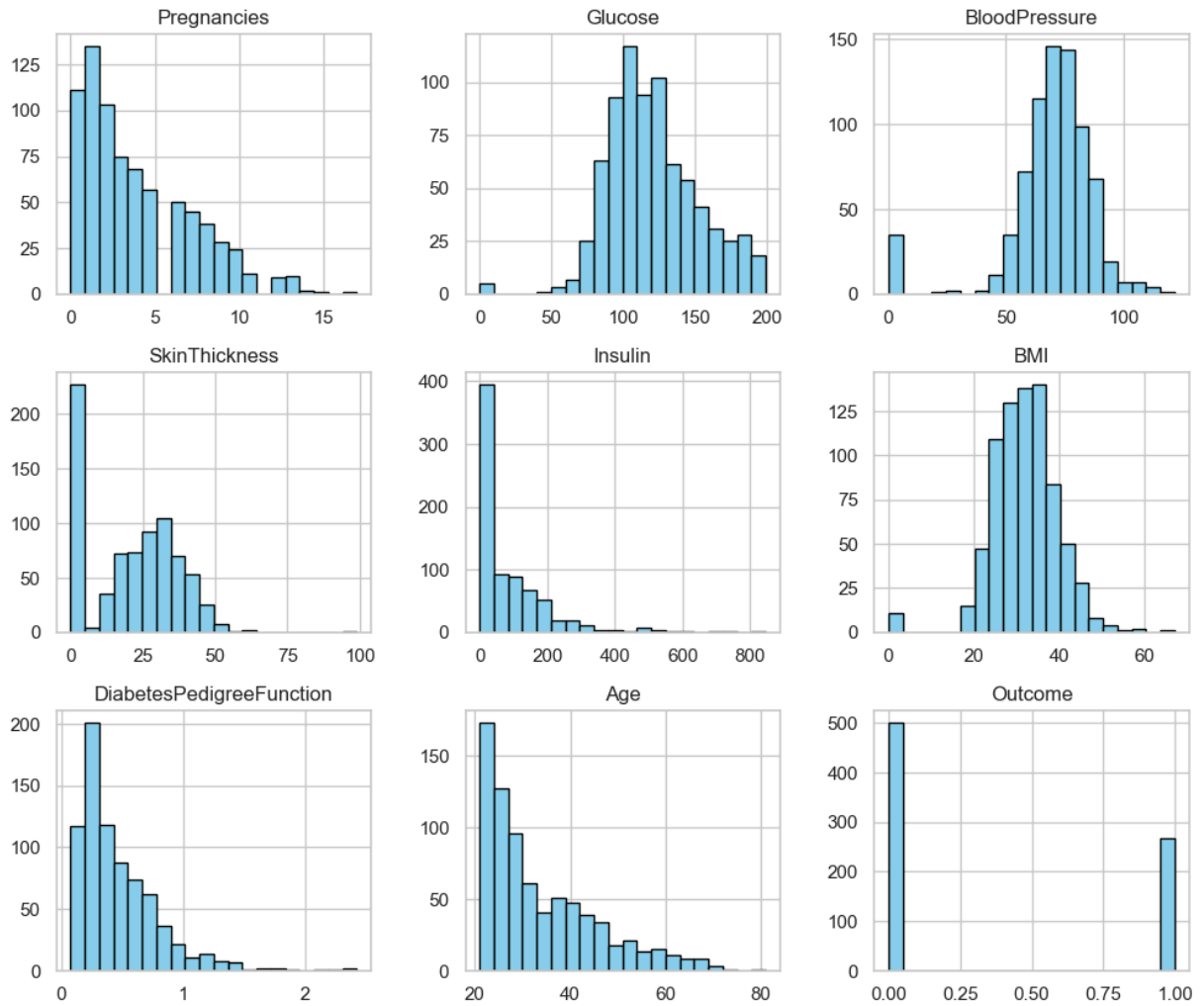
Dimensiones del dataset: (768, 9)

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabe
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	

- El dataset contiene **768 registros**.
- Variables como **Glucose, BloodPressure, SkinThickness, Insulin y BMI** presentan valores mínimos de **0**, que no son fisiológicamente válidos.
- La media de **Glucose** (~121) y de **BMI** (~32) reflejan una población con riesgo metabólico elevado.
- **Insulin** muestra dispersión (máximo 846, std ~115), indicando presencia de outliers.

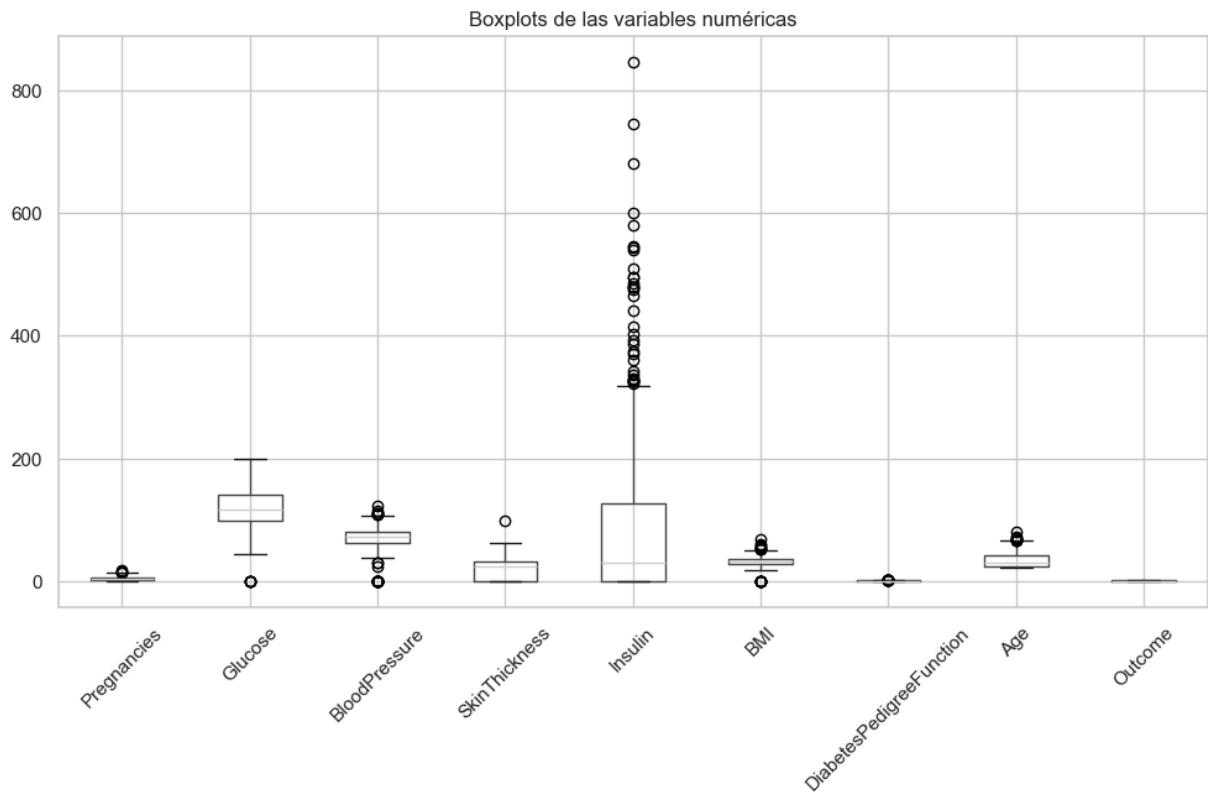
## 2. Distribuciones con histogramas

## Distribución de variables numéricas



- **Glucose** tiene una distribución aproximadamente normal centrada en ~120.
- **BloodPressure** es simétrica alrededor de 70 mmHg.
- **SkinThickness** e **Insulin** presentan fuerte sesgo hacia valores bajos, con muchos ceros.
- **BMI** se concentra en 25–40 (predominio de sobrepeso/obesidad).
- **Pregnancies** y **Age** muestran distribuciones asimétricas a la izquierda .

## 3.Detección de outliers



- Se identifican **outliers extremos**, principalmente en **Insulin** y en menor grado en **SkinThickness** y **Glucose**.
- La mayoría de variables tienen rango intercuartílico estable, salvo **Insulin** que presenta alta variabilidad.

### 3. Valores nulos

Valores nulos en el dataset:

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

Cantidad de ceros en variables críticas:

```
Glucose      5
BloodPressure 35
SkinThickness 227
Insulin      374
BMI          11
dtype: int64
```

Valores faltantes después de reemplazo de ceros:

```
Pregnancies      0
Glucose          5
BloodPressure    35
SkinThickness    227
Insulin          374
BMI              11
DiabetesPedigreeFunction  0
Age              0
Outcome          0
dtype: int64
```

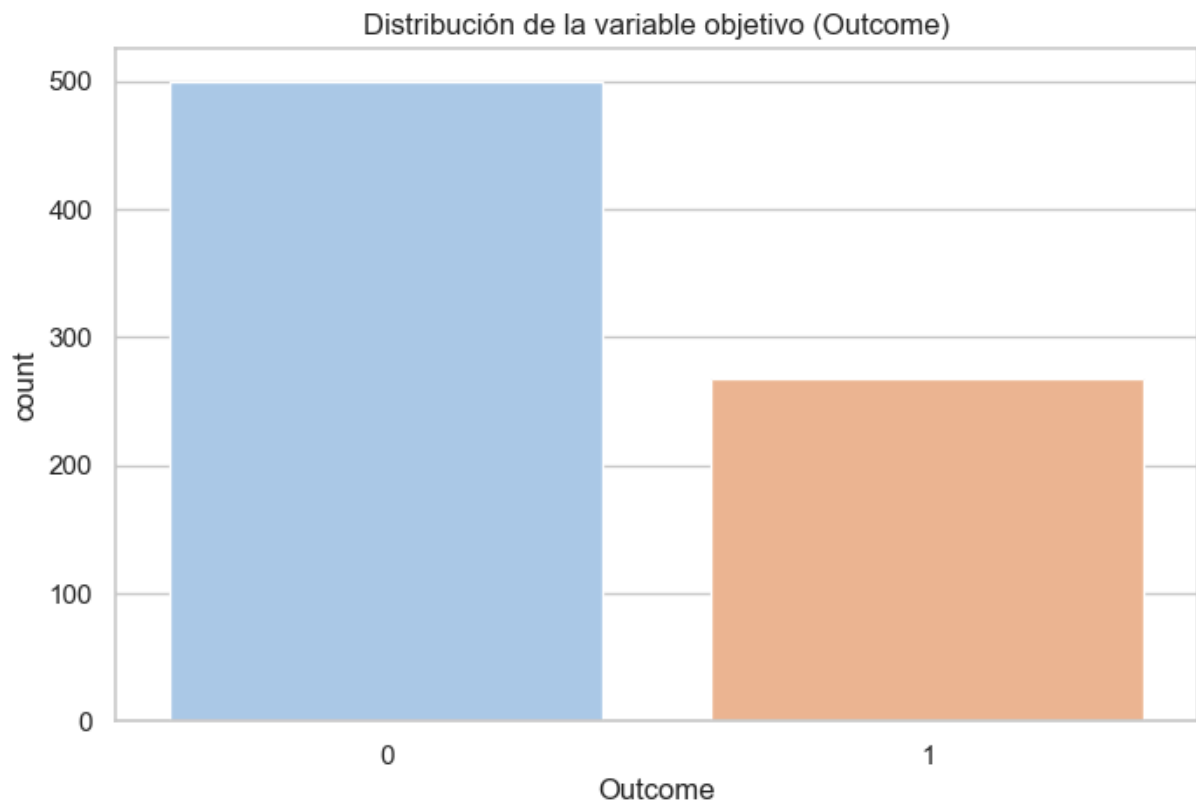
- No existen valores `NaN` explícitos, pero sí **ceros inválidos**:
  - 374 en **Insulin**
  - 227 en **SkinThickness**
  - 35 en **BloodPressure**
  - 11 en **BMI**
  - 5 en **Glucose**
- Estos ceros deben tratarse como *missing values* e imputarse o dejar que AutoGluon los maneje automáticamente.

## 4. Balance de clases

```
C:\Users\rodri\AppData\Local\Temp\ipykernel_27256\4063261591.py:1: FutureWarning:
```

```
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.
```

```
sns.countplot(x="Outcome", data=df, palette="pastel")
```



Balance de clases:

Outcome

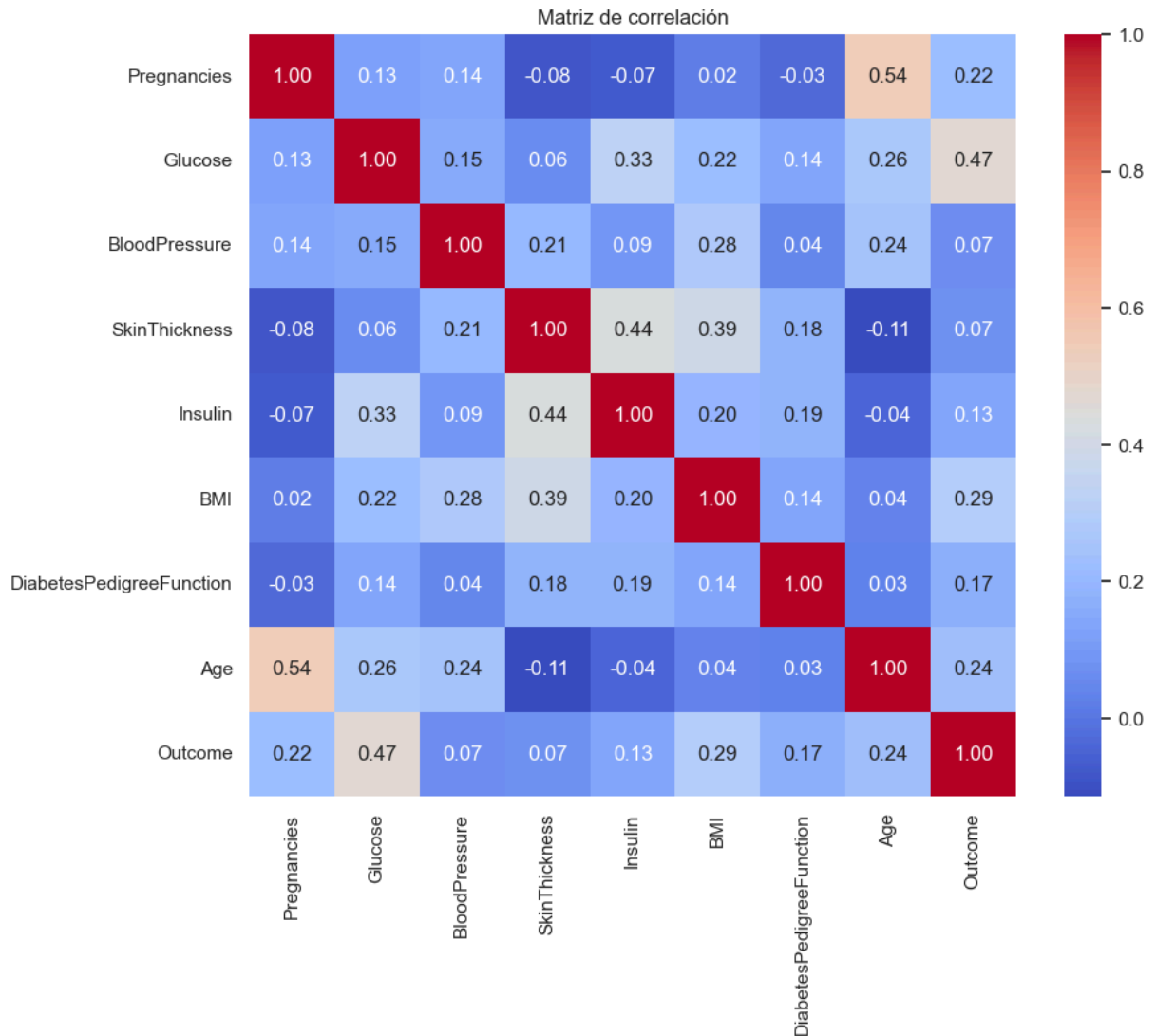
0 0.651042

1 0.348958

Name: proportion, dtype: float64

- El dataset está **desbalanceado**:
  - 0 (no diabetes): ~65%
  - 1 (diabetes): ~35%

## 5. Correlación y mapa de calor



- **Glucose** es la variable más correlacionada con **Outcome** .
- **BMI**, **Age** y **Pregnancies** también muestran correlaciones positivas relevantes.
- **Insulin** y **SkinThickness** presentan correlaciones moderadas pero con alta proporción de valores faltantes.
- No se observan correlaciones muy altas entre variables independientes → baja multicolinealidad.

## Preprocesamiento

De acuerdo con el análisis exploratorio:

- Se identificaron ceros inválidos en variables fisiológicas, tratados como valores faltantes.
- Se imputaron estos valores con la **mediana** de cada columna.
- Se generó un nuevo dataset `df_prepared` listo para entrenamiento con AutoGluon.

Valores faltantes después del preprocesamiento:

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

```
Out[ ]:      Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin  BMI  DiabetesPedigreeFunc
0           6    148.0         72.0         35.0    125.0  33.6
1           1     85.0         66.0         29.0    125.0  26.6
2           8    183.0         64.0         29.0    125.0  23.3
3           1     89.0         66.0         23.0     94.0  28.1
4           0    137.0         40.0         35.0    168.0  43.1
```

## Entrenamiento del modelo con AutoGluon

### Separación en Train/Test

Tamaño entrenamiento: (537, 9)

Tamaño prueba: (231, 9)

### Entrenamiento



```

No path specified. Models will be saved in: "AutogluonModels\ag-20250929_030411"
Verbosity: 2 (Standard Logging)
===== System Info =====
AutoGluon Version: 1.4.0
Python Version: 3.12.10
Operating System: Windows
Platform Machine: AMD64
Platform Version: 10.0.26100
CPU Count: 8
Memory Avail: 1.86 GB / 11.78 GB (15.8%)
Disk Space Avail: 177.44 GB / 475.67 GB (37.3%)
=====
Presets specified: ['best_quality']
Setting dynamic_stacking from 'auto' to True. Reason: Enable dynamic_stacking when use_bag_holdout is disabled. (use_bag_holdout=False)
Stack configuration (auto_stack=True): num_stack_levels=1, num_bag_folds=8, num_bag_sets=1
DyStack is enabled (dynamic_stacking=True). AutoGluon will try to determine whether the input data is affected by stacked overfitting and enable or disable stacking as a consequence.

    This is used to identify the optimal `num_stack_levels` value. Copies of AutoGluon will be fit on subsets of the data. Then holdout validation data is used to detect stacked overfitting.

    Running DyStack for up to 200s of the 800s of remaining time (25%).
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\autogluon\tabular\predictor\predictor.py:1444: UserWarning: Failed to use ray for memory safe fits. Falling back to normal fit. Error: ImportError('ray is required to train folds in parallel for TabularPredictor or HPO for MultiModalPredictor. A quick tip is to install via `pip install "ray>=2.10.0,<2.45.0"'')
    stacked_overfitting = self._sub_fit_memory_save_wrapper(
        Context path: "c:\Users\rodri\Documents\Data Science\Lab7-DataScience\AutogluonModels\ag-20250929_030411\ds_sub_fit\sub_fit_ho"
Running DyStack sub-fit ...
Beginning AutoGluon training ... Time limit = 200s
AutoGluon will save models to "c:\Users\rodri\Documents\Data Science\Lab7-DataScience\AutogluonModels\ag-20250929_030411\ds_sub_fit\sub_fit_ho"
Train Data Rows: 477
Train Data Columns: 8
Label Column: Outcome
Problem Type: binary
Preprocessing data ...
Selected class <--> label mapping: class 1 = 1, class 0 = 0
Using Feature Generators to preprocess the data ...
Fitting AutoMLPipelineFeatureGenerator...
    Available Memory: 1890.96 MB
    Train Data (Original) Memory Usage: 0.03 MB (0.0% of available memory)
    Inferring data type of each feature based on column values. Set feature_metadata_in to manually specify special dtypes of the features.
    Stage 1 Generators:
        Fitting AsTypeFeatureGenerator...
    Stage 2 Generators:
        Fitting FillNaFeatureGenerator...
    Stage 3 Generators:
        Fitting IdentityFeatureGenerator...
    Stage 4 Generators:

```

```

    Fitting DropUniqueFeatureGenerator...
Stage 5 Generators:
    Fitting DropDuplicatesFeatureGenerator...
Types of features in original data (raw dtype, special dtypes):
    ('float', []) : 6 | ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', ...]
    ('int', [])   : 2 | ['Pregnancies', 'Age']
Types of features in processed data (raw dtype, special dtypes):
    ('float', []) : 6 | ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', ...]
    ('int', [])   : 2 | ['Pregnancies', 'Age']
0.1s = Fit runtime
8 features in original data used to generate 8 features in processed data.
Train Data (Processed) Memory Usage: 0.03 MB (0.0% of available memory)
Data preprocessing and feature engineering runtime = 0.19s ...
AutoGluon will gauge predictive performance using evaluation metric: 'accuracy'
To change this, specify the eval_metric parameter of Predictor()
User-specified model hyperparameters to be fit:
{
    'GBM': [{}],
    'CAT': [{}],
    'XGB': [{}],
    'RF': [{}],
    'XT': [{}],
    'KNN': [{}],
    'NN_TORCH': [{}],
    'LR': [{}],
}
AutoGluon will fit 2 stack levels (L1 to L2) ...
Fitting 8 L1 models, fit_strategy="sequential" ...
Fitting model: KNeighbors_BAG_L1 ... Training model for up to 133.17s of the 199.80s of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
    0.74      = Validation score    (accuracy)
    0.01s    = Training    runtime
    0.26s    = Validation runtime
Fitting model: LightGBM_BAG_L1 ... Training model for up to 132.81s of the 199.44s of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFittingStrategy (sequential: cpus=4, gpus=0)
    Warning: Exception caused LightGBM_BAG_L1 to fail during training (ImportError)... Skipping this model.
        `import lightgbm` failed. A quick tip is to install via `pip install autogluon.tabular[lightgbm]==1.4.0`.
Fitting model: RandomForest_BAG_L1 ... Training model for up to 132.38s of the 199.01s of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
    0.7631    = Validation score    (accuracy)
    1.07s    = Training    runtime
    0.18s    = Validation runtime
Fitting model: CatBoost_BAG_L1 ... Training model for up to 130.99s of the 197.61s of remaining time.

```

```

Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=4, gpus=0)
Warning: Exception caused CatBoost_BAG_L1 to fail during training (ImportErr
or)... Skipping this model.
`import catboost` failed. A quick tip is to install via `pip install
autogluon.tabular[catboost]==1.4.0`.
Fitting model: ExtraTrees_BAG_L1 ... Training model for up to 130.50s of the 197.13s
of remaining time.
Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
0.7673 = Validation score (accuracy)
1.25s = Training runtime
0.3s = Validation runtime
Fitting model: XGBoost_BAG_L1 ... Training model for up to 128.81s of the 195.44s of
remaining time.
Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=4, gpus=0)
Warning: Exception caused XGBoost_BAG_L1 to fail during training (ImportErro
r)... Skipping this model.
`import xgboost` failed. A quick tip is to install via `pip install
autogluon.tabular[xgboost]==1.4.0`.
Fitting model: LinearModel_BAG_L1 ... Training model for up to 128.33s of the 194.96
s of remaining time.
Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=8, gpus=0)
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklear
n\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the t
otal number of samples (417). n_quantiles is set to n_samples.
warnings.warn(
Time limit exceeded... Skipping LinearModel_BAG_L1.
Fitting model: NeuralNetTorch_BAG_L1 ... Training model for up to 111.42s of the 17
8.04s of remaining time.
Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=4, gpus=0)
Warning: Exception caused NeuralNetTorch_BAG_L1 to fail during training (Imp
ortError)... Skipping this model.
Unable to import dependency torch
A quick tip is to install via `pip install torch`.
The minimum torch version is currently 2.2.
Fitting model: WeightedEnsemble_L2 ... Training model for up to 199.81s of the 177.6
8s of remaining time.
Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
Ensemble Weights: {'RandomForest_BAG_L1': 0.455, 'ExtraTrees_BAG_L1': 0.364,
'KNeighbors_BAG_L1': 0.182}
0.782 = Validation score (accuracy)
0.06s = Training runtime
0.01s = Validation runtime

```

```

Fitting 7 L2 models, fit_strategy="sequential" ...
Fitting model: LightGBM_BAG_L2 ... Training model for up to 177.54s of the 177.53s of
remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=4, gpus=0)
    Warning: Exception caused LightGBM_BAG_L2 to fail during training (ImportErr
or)... Skipping this model.
        `import lightgbm` failed. A quick tip is to install via `pip install
autogluon.tabular[lightgbm]==1.4.0`.
Fitting model: RandomForest_BAG_L2 ... Training model for up to 177.21s of the 177.2
0s of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    0.7547 = Validation score (accuracy)
    1.17s = Training runtime
    0.17s = Validation runtime
Fitting model: CatBoost_BAG_L2 ... Training model for up to 175.78s of the 175.77s o
f remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=4, gpus=0)
    Warning: Exception caused CatBoost_BAG_L2 to fail during training (ImportErr
or)... Skipping this model.
        `import catboost` failed. A quick tip is to install via `pip install
autogluon.tabular[catboost]==1.4.0`.
Fitting model: ExtraTrees_BAG_L2 ... Training model for up to 175.44s of the 175.43s
of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    0.7757 = Validation score (accuracy)
    1.09s = Training runtime
    0.16s = Validation runtime
Fitting model: XGBoost_BAG_L2 ... Training model for up to 174.12s of the 174.11s of
remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=4, gpus=0)
    Warning: Exception caused XGBoost_BAG_L2 to fail during training (ImportErro
r)... Skipping this model.
        `import xgboost` failed. A quick tip is to install via `pip install
autogluon.tabular[xgboost]==1.4.0`.
Fitting model: LinearModel_BAG_L2 ... Training model for up to 173.78s of the 173.76
s of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=8, gpus=0)
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklear
n\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the t
otal number of samples (417). n_quantiles is set to n_samples.
    warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklear

```

```

n\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (417). n_quantiles is set to n_samples.
  warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (417). n_quantiles is set to n_samples.
  warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (417). n_quantiles is set to n_samples.
  warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (417). n_quantiles is set to n_samples.
  warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (418). n_quantiles is set to n_samples.
  warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (418). n_quantiles is set to n_samples.
  warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (418). n_quantiles is set to n_samples.
  warnings.warn(
0.7694 = Validation score (accuracy)
36.4s = Training runtime
0.14s = Validation runtime
Fitting model: NeuralNetTorch_BAG_L2 ... Training model for up to 137.10s of the 137.09s of remaining time.
Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFittingStrategy (sequential: cpus=4, gpus=0)
Warning: Exception caused NeuralNetTorch_BAG_L2 to fail during training (ImportError)... Skipping this model.
Unable to import dependency torch
A quick tip is to install via `pip install torch`.
The minimum torch version is currently 2.2.
Fitting model: WeightedEnsemble_L3 ... Training model for up to 199.81s of the 136.65s of remaining time.
Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
Ensemble Weights: {'ExtraTrees_BAG_L2': 0.833, 'LinearModel_BAG_L2': 0.167}
0.7841 = Validation score (accuracy)
0.08s = Training runtime
0.0s = Validation runtime
AutoGluon training complete, total runtime = 63.5s ... Best model: WeightedEnsemble_L3 | Estimated inference throughput: 233.7 rows/s (60 batch size)
Disabling decision threshold calibration for metric `accuracy` due to having fewer than 10000 rows of validation data for calibration, to avoid overfitting (477 rows).
`accuracy` is generally not improved through threshold calibration. Force calibration via specifying `calibrate_decision_threshold=True`.
TabularPredictor saved. To load, use: predictor = TabularPredictor.load("c:\Users\ro

```

```

dri\Documents\Data Science\Lab7-DataScience\AutogluonModels\ag-20250929_030411\ds_sub_fit\sub_fit_ho")
Deleting DyStack predictor artifacts (clean_up_fits=True) ...
Leaderboard on holdout data (DyStack):
      model score_holdout score_val eval_metric pred_time_test pred_t
ime_val fit_time pred_time_test_marginal pred_time_val_marginal fit_time_marginal
al stack_level can_infer fit_order
0 RandomForest_BAG_L2      0.816667  0.754717 accuracy      0.708297
0.909560  3.501394      0.192164      0.169729      1.173
145      2      True      5
1 WeightedEnsemble_L3      0.783333  0.784067 accuracy      1.264089
1.043701  39.899482      0.034713      0.001001      0.083
637      3      True      8
2 WeightedEnsemble_L2      0.766667  0.781971 accuracy      0.559906
0.745855  2.384361      0.043773      0.006023      0.056
111      2      True      4
3 ExtraTrees_BAG_L2      0.766667  0.775681 accuracy      0.692727
0.899151  3.414614      0.176594      0.159319      1.086
364      2      True      6
4 LinearModel_BAG_L2      0.766667  0.769392 accuracy      1.052781
0.883381  38.729481      0.536648      0.143549      36.401
231      2      True      7
5 ExtraTrees_BAG_L1      0.750000  0.767296 accuracy      0.213319
0.301081  1.251873      0.213319      0.301081      1.251
873      1      True      3
6 RandomForest_BAG_L1      0.733333  0.763103 accuracy      0.199410
0.180122  1.069811      0.199410      0.180122      1.069
811      1      True      2
7 KNeighbors_BAG_L1      0.716667  0.740042 accuracy      0.103404
0.258628  0.006566      0.103404      0.258628      0.006
566      1      True      1
      1      = Optimal num_stack_levels (Stacked Overfitting Occurred: False)
      66s      = DyStack runtime | 734s      = Remaining runtime
Starting main fit with num_stack_levels=1.
For future fit calls on this dataset, you can skip DyStack to save time: `pr
edictor.fit(..., dynamic_stacking=False, num_stack_levels=1)`
Beginning AutoGluon training ... Time limit = 734s
AutoGluon will save models to "c:\Users\rodri\Documents\Data Science\Lab7-DataScienc
e\AutogluonModels\ag-20250929_030411"
Train Data Rows: 537
Train Data Columns: 8
Label Column: Outcome
Problem Type: binary
Preprocessing data ...
Selected class <--> label mapping: class 1 = 1, class 0 = 0
Using Feature Generators to preprocess the data ...
Fitting AutoMLPipelineFeatureGenerator...
Available Memory: 1119.67 MB
Train Data (Original) Memory Usage: 0.03 MB (0.0% of available memory)
Inferring data type of each feature based on column values. Set feature_meta
data_in to manually specify special dtypes of the features.
Stage 1 Generators:
Fitting AsTypeFeatureGenerator...
Stage 2 Generators:
Fitting FillNaFeatureGenerator...
Stage 3 Generators:

```

```

    Fitting IdentityFeatureGenerator...
Stage 4 Generators:
    Fitting DropUniqueFeatureGenerator...
Stage 5 Generators:
    Fitting DropDuplicatesFeatureGenerator...
Types of features in original data (raw dtype, special dtypes):
    ('float', []) : 6 | ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', ...]
    ('int', [])   : 2 | ['Pregnancies', 'Age']
Types of features in processed data (raw dtype, special dtypes):
    ('float', []) : 6 | ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', ...]
    ('int', [])   : 2 | ['Pregnancies', 'Age']
0.1s = Fit runtime
8 features in original data used to generate 8 features in processed data.
Train Data (Processed) Memory Usage: 0.03 MB (0.0% of available memory)
Data preprocessing and feature engineering runtime = 0.16s ...
AutoGluon will gauge predictive performance using evaluation metric: 'accuracy'
    To change this, specify the eval_metric parameter of Predictor()
User-specified model hyperparameters to be fit:
{
    'GBM': [{}],
    'CAT': [{}],
    'XGB': [{}],
    'RF': [{}],
    'XT': [{}],
    'KNN': [{}],
    'NN_TORCH': [{}],
    'LR': [{}],
}
AutoGluon will fit 2 stack levels (L1 to L2) ...
Fitting 8 L1 models, fit_strategy="sequential" ...
Fitting model: KNeighbors_BAG_L1 ... Training model for up to 488.90s of the 733.53s of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
    0.7207 = Validation score (accuracy)
    0.01s = Training runtime
    0.03s = Validation runtime
Fitting model: LightGBM_BAG_L1 ... Training model for up to 488.79s of the 733.42s of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFittingStrategy (sequential: cpus=4, gpus=0)
    Warning: Exception caused LightGBM_BAG_L1 to fail during training (ImportError)... Skipping this model.
    `import lightgbm` failed. A quick tip is to install via `pip install autogluon.tabular[lightgbm]==1.4.0`.
Fitting model: RandomForest_BAG_L1 ... Training model for up to 488.40s of the 733.03s of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
    0.7598 = Validation score (accuracy)
    1.4s = Training runtime
    0.2s = Validation runtime

```







```
warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (470). n_quantiles is set to n_samples.
warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (470). n_quantiles is set to n_samples.
warnings.warn(
    0.7672 = Validation score (accuracy)
    0.65s = Training runtime
    0.09s = Validation runtime
Fitting model: NeuralNetTorch_BAG_L1 ... Training model for up to 483.44s of the 728.07s of remaining time.
Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFittingStrategy (sequential: cpus=4, gpus=0)
Warning: Exception caused NeuralNetTorch_BAG_L1 to fail during training (ImportError)... Skipping this model.
Unable to import dependency torch
A quick tip is to install via `pip install torch`.
The minimum torch version is currently 2.2.
Fitting model: WeightedEnsemble_L2 ... Training model for up to 360.00s of the 727.70s of remaining time.
Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
Ensemble Weights: {'ExtraTrees_BAG_L1': 1.0}
    0.7896 = Validation score (accuracy)
    0.09s = Training runtime
    0.0s = Validation runtime
Fitting 7 L2 models, fit_strategy="sequential" ...
Fitting model: LightGBM_BAG_L2 ... Training model for up to 727.55s of the 727.54s of remaining time.
Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFittingStrategy (sequential: cpus=4, gpus=0)
Warning: Exception caused LightGBM_BAG_L2 to fail during training (ImportError)... Skipping this model.
`import lightgbm` failed. A quick tip is to install via `pip install autogluon.tabular[lightgbm]==1.4.0`.
Fitting model: RandomForest_BAG_L2 ... Training model for up to 727.20s of the 727.19s of remaining time.
Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
    0.7784 = Validation score (accuracy)
    1.37s = Training runtime
    0.18s = Validation runtime
Fitting model: CatBoost_BAG_L2 ... Training model for up to 725.58s of the 725.57s of remaining time.
Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFittingStrategy (sequential: cpus=4, gpus=0)
Warning: Exception caused CatBoost_BAG_L2 to fail during training (ImportError)
```



```

otal number of samples (470). n_quantiles is set to n_samples.
  warnings.warn(
    0.797    = Validation score    (accuracy)
    0.59s    = Training    runtime
    0.09s    = Validation runtime
Fitting model: NeuralNetTorch_BAG_L2 ... Training model for up to 721.38s of the 72
1.37s of remaining time.
  Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
  Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=4, gpus=0)
  Warning: Exception caused NeuralNetTorch_BAG_L2 to fail during training (Imp
ortError)... Skipping this model.
    Unable to import dependency torch
A quick tip is to install via `pip install torch`.
The minimum torch version is currently 2.2.
Fitting model: WeightedEnsemble_L3 ... Training model for up to 360.00s of the 721.0
0s of remaining time.
  Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
  Ensemble Weights: {'LinearModel_BAG_L2': 0.941, 'ExtraTrees_BAG_L1': 0.059}
  0.8007    = Validation score    (accuracy)
  0.08s     = Training    runtime
  0.0s      = Validation runtime
AutoGluon training complete, total runtime = 12.85s ... Best model: WeightedEnsemble
_L3 | Estimated inference throughput: 284.9 rows/s (68 batch size)
Disabling decision threshold calibration for metric `accuracy` due to having fewer t
han 10000 rows of validation data for calibration, to avoid overfitting (537 rows).
  `accuracy` is generally not improved through threshold calibration. Force ca
libration via specifying `calibrate_decision_threshold=True`.
TabularPredictor saved. To load, use: predictor = TabularPredictor.load("c:\Users\ro
dri\Documents\Data Science\Lab7-DataScience\AutogluonModels\ag-20250929_030411")
No path specified. Models will be saved in: "AutogluonModels\ag-20250929_030531"
Verbosity: 2 (Standard Logging)
===== System Info =====
AutoGluon Version: 1.4.0
Python Version: 3.12.10
Operating System: Windows
Platform Machine: AMD64
Platform Version: 10.0.26100
CPU Count: 8
Memory Avail: 1.08 GB / 11.78 GB (9.2%)
Disk Space Avail: 177.42 GB / 475.67 GB (37.3%)
=====
Presets specified: ['best_quality']
Setting dynamic_stacking from 'auto' to True. Reason: Enable dynamic_stacking when u
se_bag_holdout is disabled. (use_bag_holdout=False)
Stack configuration (auto_stack=True): num_stack_levels=1, num_bag_folds=8, num_bag_
sets=1
DyStack is enabled (dynamic_stacking=True). AutoGluon will try to determine whether
the input data is affected by stacked overfitting and enable or disable stacking as
a consequence.
  This is used to identify the optimal `num_stack_levels` value. Copies of Aut
oGluon will be fit on subsets of the data. Then holdout validation data is used to d
etect stacked overfitting.
    Running DyStack for up to 625s of the 2500s of remaining time (25%).

```

```

c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\autogluon\tabular\predictor\predictor.py:1444: UserWarning: Failed to use ray for memory safe fits. Falling back to normal fit. Error: ImportError('ray is required to train folds in parallel for TabularPredictor or HPO for MultiModalPredictor. A quick tip is to install via `pip install "ray>=2.10.0,<2.45.0"'`)
  stacked_overfitting = self._sub_fit_memory_save_wrapper(
    Context path: "c:\Users\rodri\Documents\Data Science\Lab7-DataScience\AutogluonModels\ag-20250929_030531\ds_sub_fit\sub_fit_ho"
Running DyStack sub-fit ...
Beginning AutoGluon training ... Time limit = 625s
AutoGluon will save models to "c:\Users\rodri\Documents\Data Science\Lab7-DataScience\AutogluonModels\ag-20250929_030531\ds_sub_fit\sub_fit_ho"
Train Data Rows:      477
Train Data Columns: 8
Label Column:         Outcome
Problem Type:          binary
Preprocessing data ...
Selected class <--> label mapping:  class 1 = 1, class 0 = 0
Using Feature Generators to preprocess the data ...
Fitting AutoMLPipelineFeatureGenerator...
  Available Memory:              1109.46 MB
  Train Data (Original) Memory Usage: 0.03 MB (0.0% of available memory)
  Inferring data type of each feature based on column values. Set feature_metadata_in to manually specify special dtypes of the features.
  Stage 1 Generators:
    Fitting AsTypeFeatureGenerator...
  Stage 2 Generators:
    Fitting FillNaFeatureGenerator...
  Stage 3 Generators:
    Fitting IdentityFeatureGenerator...
  Stage 4 Generators:
    Fitting DropUniqueFeatureGenerator...
  Stage 5 Generators:
    Fitting DropDuplicatesFeatureGenerator...
  Types of features in original data (raw dtype, special dtypes):
    ('float', []) : 6 | ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', ...]
    ('int', [])   : 2 | ['Pregnancies', 'Age']
  Types of features in processed data (raw dtype, special dtypes):
    ('float', []) : 6 | ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', ...]
    ('int', [])   : 2 | ['Pregnancies', 'Age']
  0.1s = Fit runtime
  8 features in original data used to generate 8 features in processed data.
  Train Data (Processed) Memory Usage: 0.03 MB (0.0% of available memory)
Data preprocessing and feature engineering runtime = 0.12s ...
AutoGluon will gauge predictive performance using evaluation metric: 'accuracy'
  To change this, specify the eval_metric parameter of Predictor()
User-specified model hyperparameters to be fit:
{
  'GBM': [{}],
  'CAT': [{}],
  'XGB': [{}],
  'RF': [{}],
  'XT': [{}],
  'KNN': [{}],

```

```

'NN_TORCH': [{}],
'LR': [{}],
}
AutoGluon will fit 2 stack levels (L1 to L2) ...
Fitting 8 L1 models, fit_strategy="sequential" ...
Fitting model: KNeighbors_BAG_L1 ... Training model for up to 416.48s of the 624.86s
of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    0.74      = Validation score    (accuracy)
    0.02s     = Training    runtime
    0.02s     = Validation runtime
Fitting model: LightGBM_BAG_L1 ... Training model for up to 416.35s of the 624.73s o
f remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=4, gpus=0)
    Warning: Exception caused LightGBM_BAG_L1 to fail during training (ImportErr
or)... Skipping this model.
    `import lightgbm` failed. A quick tip is to install via `pip install
autogluon.tabular[lightgbm]==1.4.0`.
Fitting model: RandomForest_BAG_L1 ... Training model for up to 415.97s of the 624.3
5s of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    0.7631    = Validation score    (accuracy)
    1.43s     = Training    runtime
    0.19s     = Validation runtime
Fitting model: CatBoost_BAG_L1 ... Training model for up to 414.24s of the 622.62s o
f remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=4, gpus=0)
    Warning: Exception caused CatBoost_BAG_L1 to fail during training (ImportErr
or)... Skipping this model.
    `import catboost` failed. A quick tip is to install via `pip install
autogluon.tabular[catboost]==1.4.0`.
Fitting model: ExtraTrees_BAG_L1 ... Training model for up to 413.88s of the 622.26s
of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    0.7673    = Validation score    (accuracy)
    1.41s     = Training    runtime
    0.25s     = Validation runtime
Fitting model: XGBoost_BAG_L1 ... Training model for up to 412.09s of the 620.48s of
remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=4, gpus=0)
    Warning: Exception caused XGBoost_BAG_L1 to fail during training (ImportErro
r)... Skipping this model.
    `import xgboost` failed. A quick tip is to install via `pip install
autogluon.tabular[xgboost]==1.4.0`.

```

Fitting model: LinearModel\_BAG\_L1 ... Training model for up to 411.74s of the 620.12s of remaining time.

Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`

Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFittingStrategy (sequential: cpus=8, gpus=0)

c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\\_data.py:2846: UserWarning: n\_quantiles (1000) is greater than the total number of samples (417). n\_quantiles is set to n\_samples.

warnings.warn(

c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\\_data.py:2846: UserWarning: n\_quantiles (1000) is greater than the total number of samples (417). n\_quantiles is set to n\_samples.

warnings.warn(

c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\\_data.py:2846: UserWarning: n\_quantiles (1000) is greater than the total number of samples (417). n\_quantiles is set to n\_samples.

warnings.warn(

c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\\_data.py:2846: UserWarning: n\_quantiles (1000) is greater than the total number of samples (417). n\_quantiles is set to n\_samples.

warnings.warn(

c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\\_data.py:2846: UserWarning: n\_quantiles (1000) is greater than the total number of samples (417). n\_quantiles is set to n\_samples.

warnings.warn(

c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\\_data.py:2846: UserWarning: n\_quantiles (1000) is greater than the total number of samples (418). n\_quantiles is set to n\_samples.

warnings.warn(

c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\\_data.py:2846: UserWarning: n\_quantiles (1000) is greater than the total number of samples (418). n\_quantiles is set to n\_samples.

warnings.warn(

c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\\_data.py:2846: UserWarning: n\_quantiles (1000) is greater than the total number of samples (418). n\_quantiles is set to n\_samples.

warnings.warn(

0.782 = Validation score (accuracy)

0.68s = Training runtime

0.15s = Validation runtime

Fitting model: NeuralNetTorch\_BAG\_L1 ... Training model for up to 410.75s of the 619.13s of remaining time.

Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`

Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFittingStrategy (sequential: cpus=4, gpus=0)

Warning: Exception caused NeuralNetTorch\_BAG\_L1 to fail during training (ImportError)... Skipping this model.

Unable to import dependency torch

A quick tip is to install via `pip install torch`.

The minimum torch version is currently 2.2.

Fitting model: WeightedEnsemble\_L2 ... Training model for up to 360.00s of the 618.71s of remaining time.

Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`



```

    Ensemble Weights: {'LinearModel_BAG_L1': 0.778, 'ExtraTrees_BAG_L1': 0.167,
'KNeighbors_BAG_L1': 0.056}
    0.7904 = Validation score (accuracy)
    0.13s = Training runtime
    0.0s = Validation runtime
Fitting 7 L2 models, fit_strategy="sequential" ...
Fitting model: LightGBM_BAG_L2 ... Training model for up to 618.51s of the 618.50s of
remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=4, gpus=0)
    Warning: Exception caused LightGBM_BAG_L2 to fail during training (ImportErr
or)... Skipping this model.
    `import lightgbm` failed. A quick tip is to install via `pip install
autogluon.tabular[lightgbm]==1.4.0`.
Fitting model: RandomForest_BAG_L2 ... Training model for up to 618.13s of the 618.1
2s of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    0.7589 = Validation score (accuracy)
    1.45s = Training runtime
    0.26s = Validation runtime
Fitting model: CatBoost_BAG_L2 ... Training model for up to 616.31s of the 616.30s of
remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=4, gpus=0)
    Warning: Exception caused CatBoost_BAG_L2 to fail during training (ImportErr
or)... Skipping this model.
    `import catboost` failed. A quick tip is to install via `pip install
autogluon.tabular[catboost]==1.4.0`.
Fitting model: ExtraTrees_BAG_L2 ... Training model for up to 615.66s of the 615.65s
of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    0.7841 = Validation score (accuracy)
    1.55s = Training runtime
    0.17s = Validation runtime
Fitting model: XGBoost_BAG_L2 ... Training model for up to 613.85s of the 613.85s of
remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=4, gpus=0)
    Warning: Exception caused XGBoost_BAG_L2 to fail during training (ImportErro
r)... Skipping this model.
    `import xgboost` failed. A quick tip is to install via `pip install
autogluon.tabular[xgboost]==1.4.0`.
Fitting model: LinearModel_BAG_L2 ... Training model for up to 613.51s of the 613.50
s of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=8, gpus=0)

```

```

c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (417). n_quantiles is set to n_samples.
  warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (417). n_quantiles is set to n_samples.
  warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (417). n_quantiles is set to n_samples.
  warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (417). n_quantiles is set to n_samples.
  warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (418). n_quantiles is set to n_samples.
  warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (418). n_quantiles is set to n_samples.
  warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (418). n_quantiles is set to n_samples.
  warnings.warn(
    0.7652 = Validation score (accuracy)
    0.52s = Training runtime
    0.11s = Validation runtime
Fitting model: NeuralNetTorch_BAG_L2 ... Training model for up to 612.75s of the 612.75s of remaining time.
  Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
  Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFittingStrategy (sequential: cpus=4, gpus=0)
  Warning: Exception caused NeuralNetTorch_BAG_L2 to fail during training (ImportError)... Skipping this model.
  Unable to import dependency torch
A quick tip is to install via `pip install torch`.
The minimum torch version is currently 2.2.
Fitting model: WeightedEnsemble_L3 ... Training model for up to 360.00s of the 612.40s of remaining time.
  Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
  Ensemble Weights: {'ExtraTrees_BAG_L2': 1.0}
    0.7841 = Validation score (accuracy)
    0.11s = Training runtime
    0.0s = Validation runtime
AutoGluon training complete, total runtime = 12.78s ... Best model: WeightedEnsemble_L2 | Estimated inference throughput: 318.9 rows/s (60 batch size)

```



Disabling decision threshold calibration for metric `accuracy` due to having fewer than 10000 rows of validation data for calibration, to avoid overfitting (477 rows).

`accuracy` is generally not improved through threshold calibration. Force calibration via specifying `calibrate\_decision\_threshold=True`.

TabularPredictor saved. To load, use: predictor = TabularPredictor.load("c:\Users\rodri\Documents\Data Science\Lab7-DataScience\AutogluonModels\ag-20250929\_030531\ds\_sub\_fit\sub\_fit\_ho")

Deleting DyStack predictor artifacts (clean\_up\_fits=True) ...

Leaderboard on holdout data (DyStack):

	model	score_holdout	score_val	eval_metric	pred_time_test	pred_time_val
ime_val	fit_time	pred_time_test_marginal	pred_time_val_marginal	fit_time_marginal		
1	stack_level	can_infer	fit_order			
0	ExtraTrees_BAG_L2	0.783333	0.784067	accuracy	1.276899	
0.791894	5.089104		0.243662		0.173539	1.5543
39	2	True	7			
1	WeightedEnsemble_L3	0.783333	0.784067	accuracy	1.320133	
0.793895	5.200247		0.043235		0.002000	0.1111
43	3	True	9			
2	LinearModel_BAG_L2	0.783333	0.765199	accuracy	2.196986	
0.729573	4.052490		1.163749		0.111218	0.5177
25	2	True	8			
3	LinearModel_BAG_L1	0.766667	0.781971	accuracy	0.553040	
0.153446	0.680997		0.553040		0.153446	0.6809
97	1	True	4			
4	WeightedEnsemble_L2	0.766667	0.790356	accuracy	0.883640	
0.429525	2.233234		0.033443		0.003416	0.1264
72	2	True	5			
5	ExtraTrees_BAG_L1	0.750000	0.767296	accuracy	0.192103	
0.251897	1.406409		0.192103		0.251897	1.4064
09	1	True	3			
6	RandomForest_BAG_L2	0.750000	0.758910	accuracy	1.216244	
0.882919	4.986523		0.183007		0.264563	1.4517
59	2	True	6			
7	RandomForest_BAG_L1	0.733333	0.763103	accuracy	0.183040	
0.192246	1.428004		0.183040		0.192246	1.4280
04	1	True	2			
8	KNeighbors_BAG_L1	0.716667	0.740042	accuracy	0.105053	
0.020766	0.019355		0.105053		0.020766	0.0193
55	1	True	1			

1 = Optimal num\_stack\_levels (Stacked Overfitting Occurred: False)

17s = DyStack runtime | 2483s = Remaining runtime

Starting main fit with num\_stack\_levels=1.

For future fit calls on this dataset, you can skip DyStack to save time: `predictor.fit(..., dynamic\_stacking=False, num\_stack\_levels=1)`

Beginning AutoGluon training ... Time limit = 2483s

AutoGluon will save models to "c:\Users\rodri\Documents\Data Science\Lab7-DataScience\AutogluonModels\ag-20250929\_030531"

Train Data Rows: 537

Train Data Columns: 8

Label Column: Outcome

Problem Type: binary

Preprocessing data ...

Selected class <--> label mapping: class 1 = 1, class 0 = 0

Using Feature Generators to preprocess the data ...

Fitting AutoMLPipelineFeatureGenerator...

Available Memory: 1059.02 MB

```

Train Data (Original) Memory Usage: 0.03 MB (0.0% of available memory)
Inferring data type of each feature based on column values. Set feature_meta
data_in to manually specify special dtypes of the features.
Stage 1 Generators:
    Fitting AsTypeFeatureGenerator...
Stage 2 Generators:
    Fitting FillNaFeatureGenerator...
Stage 3 Generators:
    Fitting IdentityFeatureGenerator...
Stage 4 Generators:
    Fitting DropUniqueFeatureGenerator...
Stage 5 Generators:
    Fitting DropDuplicatesFeatureGenerator...
Types of features in original data (raw dtype, special dtypes):
    ('float', []) : 6 | ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', ...]
    ('int', [])   : 2 | ['Pregnancies', 'Age']
Types of features in processed data (raw dtype, special dtypes):
    ('float', []) : 6 | ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', ...]
    ('int', [])   : 2 | ['Pregnancies', 'Age']
0.1s = Fit runtime
8 features in original data used to generate 8 features in processed data.
Train Data (Processed) Memory Usage: 0.03 MB (0.0% of available memory)
Data preprocessing and feature engineering runtime = 0.13s ...
AutoGluon will gauge predictive performance using evaluation metric: 'accuracy'
To change this, specify the eval_metric parameter of Predictor()
User-specified model hyperparameters to be fit:
{
    'GBM': [{}],
    'CAT': [{}],
    'XGB': [{}],
    'RF': [{}],
    'XT': [{}],
    'KNN': [{}],
    'NN_TORCH': [{}],
    'LR': [{}],
}
AutoGluon will fit 2 stack levels (L1 to L2) ...
Fitting 8 L1 models, fit_strategy="sequential" ...
Fitting model: KNeighbors_BAG_L1 ... Training model for up to 1655.10s of the 2483.2
6s of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    0.7207 = Validation score (accuracy)
    0.01s = Training runtime
    0.09s = Validation runtime
Fitting model: LightGBM_BAG_L1 ... Training model for up to 1654.85s of the 2483.02s
of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFittingStrategy (sequential: cpus=4, gpus=0)
    Warning: Exception caused LightGBM_BAG_L1 to fail during training (ImportError)... Skipping this model.
    `import lightgbm` failed. A quick tip is to install via `pip install

```

```

autogluon.tabular[lightgbm]==1.4.0`.
Fitting model: RandomForest_BAG_L1 ... Training model for up to 1654.43s of the 248
2.60s of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    0.7598 = Validation score (accuracy)
    1.37s = Training runtime
    0.2s = Validation runtime
Fitting model: CatBoost_BAG_L1 ... Training model for up to 1652.74s of the 2480.91s
of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=4, gpus=0)
    Warning: Exception caused CatBoost_BAG_L1 to fail during training (ImportErr
or)... Skipping this model.
    `import catboost` failed. A quick tip is to install via `pip install
autogluon.tabular[catboost]==1.4.0`.
Fitting model: ExtraTrees_BAG_L1 ... Training model for up to 1652.38s of the 2480.5
4s of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    0.7896 = Validation score (accuracy)
    1.25s = Training runtime
    0.2s = Validation runtime
Fitting model: XGBoost_BAG_L1 ... Training model for up to 1650.82s of the 2478.98s
of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=4, gpus=0)
    Warning: Exception caused XGBoost_BAG_L1 to fail during training (ImportErro
r)... Skipping this model.
    `import xgboost` failed. A quick tip is to install via `pip install
autogluon.tabular[xgboost]==1.4.0`.
Fitting model: LinearModel_BAG_L1 ... Training model for up to 1650.44s of the 2478.
60s of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=8, gpus=0)
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklear
n\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the t
otal number of samples (469). n_quantiles is set to n_samples.
    warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklear
n\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the t
otal number of samples (470). n_quantiles is set to n_samples.
    warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklear
n\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the t
otal number of samples (470). n_quantiles is set to n_samples.
    warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklear
n\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the t
otal number of samples (470). n_quantiles is set to n_samples.

```

```

warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (470). n_quantiles is set to n_samples.
warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (470). n_quantiles is set to n_samples.
warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (470). n_quantiles is set to n_samples.
warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (470). n_quantiles is set to n_samples.
warnings.warn(
0.7672 = Validation score (accuracy)
0.51s = Training runtime
0.08s = Validation runtime
Fitting model: NeuralNetTorch_BAG_L1 ... Training model for up to 1649.73s of the 2477.90s of remaining time.
Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFittingStrategy (sequential: cpus=4, gpus=0)
Warning: Exception caused NeuralNetTorch_BAG_L1 to fail during training (ImportError)... Skipping this model.
Unable to import dependency torch
A quick tip is to install via `pip install torch`.
The minimum torch version is currently 2.2.
Fitting model: WeightedEnsemble_L2 ... Training model for up to 360.00s of the 2477.54s of remaining time.
Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
Ensemble Weights: {'ExtraTrees_BAG_L1': 1.0}
0.7896 = Validation score (accuracy)
0.09s = Training runtime
0.0s = Validation runtime
Fitting 7 L2 models, fit_strategy="sequential" ...
Fitting model: LightGBM_BAG_L2 ... Training model for up to 2477.40s of the 2477.38s of remaining time.
Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFittingStrategy (sequential: cpus=4, gpus=0)
Warning: Exception caused LightGBM_BAG_L2 to fail during training (ImportError)... Skipping this model.
`import lightgbm` failed. A quick tip is to install via `pip install autogluon.tabular[lightgbm]==1.4.0`.
Fitting model: RandomForest_BAG_L2 ... Training model for up to 2477.02s of the 2477.00s of remaining time.
Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
0.7784 = Validation score (accuracy)
1.44s = Training runtime

```

```

    0.62s    = Validation runtime
Fitting model: CatBoost_BAG_L2 ... Training model for up to 2474.81s of the 2474.80s
of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=4, gpus=0)
    Warning: Exception caused CatBoost_BAG_L2 to fail during training (ImportErr
or)... Skipping this model.
        `import catboost` failed. A quick tip is to install via `pip install
autogluon.tabular[catboost]==1.4.0`.
Fitting model: ExtraTrees_BAG_L2 ... Training model for up to 2474.44s of the 2474.4
3s of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    0.7821   = Validation score    (accuracy)
    1.36s    = Training    runtime
    0.23s    = Validation runtime
Fitting model: XGBoost_BAG_L2 ... Training model for up to 2472.78s of the 2472.76s
of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=4, gpus=0)
    Warning: Exception caused XGBoost_BAG_L2 to fail during training (ImportErro
r)... Skipping this model.
        `import xgboost` failed. A quick tip is to install via `pip install
autogluon.tabular[xgboost]==1.4.0`.
Fitting model: LinearModel_BAG_L2 ... Training model for up to 2472.38s of the 2472.
36s of remaining time.
    Failed to import torch or check CUDA availability!Please ensure you have the
correct version of PyTorch installed by running `pip install -U torch`
    Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFitti
ngStrategy (sequential: cpus=8, gpus=0)
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklear
n\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the t
otal number of samples (469). n_quantiles is set to n_samples.
    warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklear
n\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the t
otal number of samples (470). n_quantiles is set to n_samples.
    warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklear
n\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the t
otal number of samples (470). n_quantiles is set to n_samples.
    warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklear
n\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the t
otal number of samples (470). n_quantiles is set to n_samples.
    warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklear
n\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the t
otal number of samples (470). n_quantiles is set to n_samples.
    warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklear
n\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the t
otal number of samples (470). n_quantiles is set to n_samples.
    warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklear
n\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the t

```

```

otal number of samples (470). n_quantiles is set to n_samples.
  warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (470). n_quantiles is set to n_samples.
  warnings.warn(
c:\Users\rodri\Documents\Data Science\Lab7-DataScience\venv\Lib\site-packages\sklearn\preprocessing\_data.py:2846: UserWarning: n_quantiles (1000) is greater than the total number of samples (470). n_quantiles is set to n_samples.
  warnings.warn(
    0.797    = Validation score    (accuracy)
    0.93s    = Training    runtime
    0.15s    = Validation runtime
Fitting model: NeuralNetTorch_BAG_L2 ... Training model for up to 2471.12s of the 2471.10s of remaining time.
  Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
  Fitting 8 child models (S1F1 - S1F8) | Fitting with SequentialLocalFoldFittingStrategy (sequential: cpus=4, gpus=0)
  Warning: Exception caused NeuralNetTorch_BAG_L2 to fail during training (ImportError)... Skipping this model.
    Unable to import dependency torch
A quick tip is to install via `pip install torch`.
The minimum torch version is currently 2.2.
Fitting model: WeightedEnsemble_L3 ... Training model for up to 360.00s of the 2470.71s of remaining time.
  Failed to import torch or check CUDA availability!Please ensure you have the correct version of PyTorch installed by running `pip install -U torch`
  Ensemble Weights: {'LinearModel_BAG_L2': 0.941, 'ExtraTrees_BAG_L1': 0.059}
    0.8007    = Validation score    (accuracy)
    0.07s    = Training    runtime
    0.0s     = Validation runtime
AutoGluon training complete, total runtime = 12.84s ... Best model: WeightedEnsemble_L3 | Estimated inference throughput: 230.6 rows/s (68 batch size)
Disabling decision threshold calibration for metric `accuracy` due to having fewer than 10000 rows of validation data for calibration, to avoid overfitting (537 rows).
  `accuracy` is generally not improved through threshold calibration. Force calibration via specifying `calibrate_decision_threshold=True`.
TabularPredictor saved. To load, use: predictor = TabularPredictor.load("c:\Users\rodri\Documents\Data Science\Lab7-DataScience\AutogluonModels\ag-20250929_030531")

```

	model	score_test	score_val	eval_metric	pred_time_test	pred_time_val	fi
0	WeightedEnsemble_L3	0.753247	0.800745	accuracy	1.665514	0.728731	4.1
1	ExtraTrees_BAG_L1	0.748918	0.789572	accuracy	0.203488	0.204077	1.2
2	WeightedEnsemble_L2	0.748918	0.789572	accuracy	0.237428	0.206102	1.3
3	LinearModel_BAG_L2	0.748918	0.797020	accuracy	1.631587	0.727735	4.0
4	RandomForest_BAG_L1	0.740260	0.759777	accuracy	0.224081	0.199665	1.3
5	ExtraTrees_BAG_L2	0.740260	0.782123	accuracy	1.285525	0.808797	4.5
6	LinearModel_BAG_L1	0.735931	0.767225	accuracy	0.545752	0.083755	0.5
7	RandomForest_BAG_L2	0.735931	0.778399	accuracy	1.292126	1.200939	4.5
8	KNeighbors_BAG_L1	0.696970	0.720670	accuracy	0.119790	0.092068	0.0

Modelos entrenados: ['WeightedEnsemble\_L3', 'ExtraTrees\_BAG\_L1', 'WeightedEnsemble\_L2', 'LinearModel\_BAG\_L2', 'RandomForest\_BAG\_L1', 'ExtraTrees\_BAG\_L2', 'LinearModel\_BAG\_L1', 'RandomForest\_BAG\_L2', 'KNeighbors\_BAG\_L1']

- El **mejor modelo fue el ensamble de nivel 3 (WeightedEnsemble\_L3)** con **accuracy de 0.753 en test** y 0.801 en validación.
- Los modelos base como **ExtraTrees y RandomForest** lograron entre 0.735–0.749 de accuracy, lo cual es competitivo pero menor al ensamble.
- **KNeighbors** obtuvo el peor desempeño (0.697), confirmando que técnicas basadas en distancia son menos efectivas en este dataset con alta dimensionalidad y escalas heterogéneas.
- Los **modelos lineales** tuvieron desempeño intermedio (~0.736–0.749), lo esperado en datasets donde las relaciones no son completamente lineales.

## Baseline vs Modelos

Accuracy – Baseline (LogisticRegression): 0.7446  
Accuracy – AutoGluon (mejor ensamble/modelo): 0.7532  
Diferencia (AG - Baseline): +0.0087

- La **regresión logística** (modelo base) alcanzó un **accuracy = 0.7446**.
- El **mejor ensamble de AutoGluon (WeightedEnsemble\_L3)** logró un **accuracy = 0.7532**, una mejora de **+0.0087**.
- La ganancia es modesta, pero confirma que el ensamble automático de varios clasificadores puede superar al modelo lineal tradicional.
- En problemas clínicos, incluso mejoras pequeñas pueden ser relevantes si implican menos falsos negativos.

## Feature Importance

Computing feature importance via permutation shuffling for 8 features using 231 rows with 5 shuffle sets...

25.7s = Expected runtime (5.14s per shuffle set)

3.77s = Actual runtime (Completed 5 of 5 shuffle sets)

	importance	stddev	p_value	n	p99_high	p99_low
<b>Glucose</b>	0.083117	0.017954	0.000246	5	0.120084	0.046150
<b>Pregnancies</b>	0.017316	0.018110	0.049650	5	0.054604	-0.019972
<b>BMI</b>	0.017316	0.028387	0.122140	5	0.075766	-0.041134
<b>SkinThickness</b>	0.008658	0.005302	0.010871	5	0.019575	-0.002259
<b>Age</b>	0.008658	0.010604	0.070964	5	0.030491	-0.013175
<b>DiabetesPedigreeFunction</b>	0.001732	0.010426	0.364591	5	0.023198	-0.019735
<b>Insulin</b>	-0.000866	0.009385	0.576678	5	0.018458	-0.020190
<b>BloodPressure</b>	-0.006926	0.003872	0.991935	5	0.001046	-0.014899

Top importancia de variables:

	importance
Glucose	0.083117
Pregnancies	0.017316
BMI	0.017316
SkinThickness	0.008658
Age	0.008658
DiabetesPedigreeFunction	0.001732
Insulin	-0.000866
BloodPressure	-0.006926

- **Glucose** es la variable más importante (0.083), consistente con la fisiopatología de la diabetes: la glucemia elevada es el predictor principal.
- **Pregnancies** y **BMI** también muestran relevancia positiva, coherente con factores de riesgo conocidos (obesidad, embarazos previos).
- **SkinThickness** y **Age** aportan información secundaria.
- **DiabetesPedigreeFunction** tiene baja importancia, lo que sugiere que el historial familiar aporta menos valor predictivo en este dataset.
- **Insulin** y **BloodPressure** resultaron con importancia cercana a cero o negativa, probablemente por ruido, outliers y alta proporción de datos imputados.

## Evaluación del modelo

### Tabla de Modelos

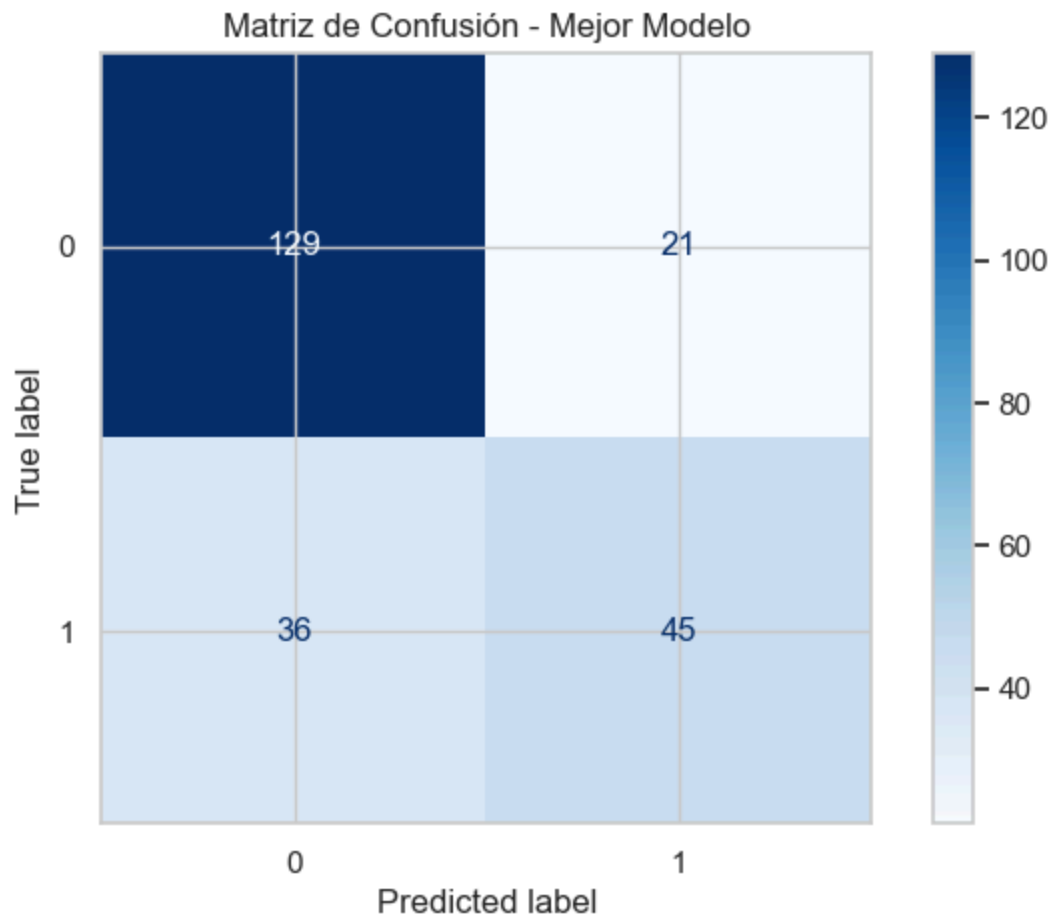


	model	score_test	score_val	eval_metric	pred_time_test	pred_time_val	fi
0	WeightedEnsemble_L3	0.753247	0.800745	accuracy	0.804070	0.728731	4.1
1	ExtraTrees_BAG_L1	0.748918	0.789572	accuracy	0.240000	0.204077	1.2
2	WeightedEnsemble_L2	0.748918	0.789572	accuracy	0.243528	0.206102	1.3
3	LinearModel_BAG_L2	0.748918	0.797020	accuracy	0.802070	0.727735	4.0
4	RandomForest_BAG_L1	0.740260	0.759777	accuracy	0.197995	0.199665	1.3
5	ExtraTrees_BAG_L2	0.740260	0.782123	accuracy	0.844939	0.808797	4.5
6	LinearModel_BAG_L1	0.735931	0.767225	accuracy	0.191545	0.083755	0.5
7	RandomForest_BAG_L2	0.735931	0.778399	accuracy	0.858677	1.200939	4.5
8	KNeighbors_BAG_L1	0.696970	0.720670	accuracy	0.064295	0.092068	0.0

Mejor modelo: WeightedEnsemble\_L3

- El mejor modelo fue el **WeightedEnsemble\_L3** con **accuracy = 0.753 en test** y **0.801 en validación**.
- Modelos como **ExtraTrees** y **LinearModel** alcanzaron valores similares , pero el ensamble logra una ligera ventaja al combinar clasificadores.
- **KNeighbors** obtuvo el peor desempeño, confirmando su menor capacidad en datasets tabulares.

## Confusion Matrix



- **Clase 0 (no diabetes):** 129 verdaderos negativos, 21 falsos positivos.
- **Clase 1 (diabetes):** 45 verdaderos positivos, 36 falsos negativos.
- El modelo clasifica mejor a los pacientes **sin diabetes**, pero pierde sensibilidad en detectar casos positivos.

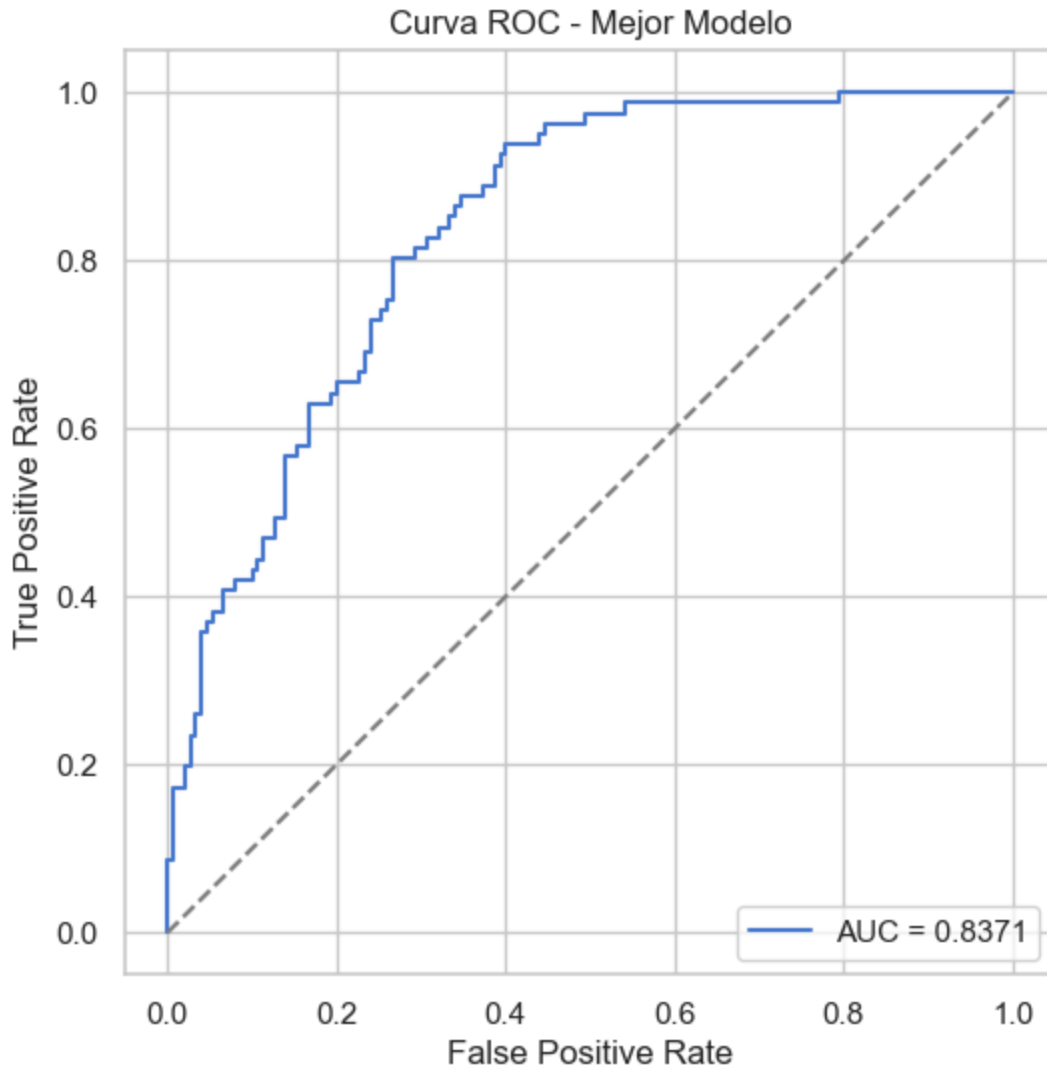
## Métricas adicionales

	precision	recall	f1-score	support
0	0.7818	0.8600	0.8190	150
1	0.6818	0.5556	0.6122	81
accuracy			0.7532	231
macro avg	0.7318	0.7078	0.7156	231
weighted avg	0.7468	0.7532	0.7465	231

- **Precisión (Clase 1):** 0.68 → de cada 10 predicciones positivas, ~7 son correctas.
- **Recall (Clase 1):** 0.56 → el modelo detecta solo un poco más de la mitad de los pacientes con diabetes.
- **F1-score (Clase 1):** 0.61 → equilibrio moderado entre precisión y recall.

- **Macro promedio:** 0.72 (precision), 0.71 (recall), 0.72 (F1).
- Esto refleja que el dataset está desbalanceado y el modelo tiende a favorecer la clase negativa.

## Curva ROC y AUC



- El **AUC = 0.837** indica buena capacidad discriminativa global entre clases.
- El modelo es capaz de separar adecuadamente positivos y negativos al variar el umbral de decisión.
- Sin embargo, el recall relativamente bajo sugiere que, en aplicaciones clínicas, se debería ajustar el umbral de predicción para priorizar la detección de casos positivos.
- AutoGluon logró un **mejor desempeño que la regresión logística base**, pero el **trade-off entre precisión y recall** debe considerarse.
- En salud, **minimizar falsos negativos** es prioritario, incluso a costa de aumentar falsos positivos.

- Ajustar el umbral de decisión o aplicar técnicas de balanceo de clases podría mejorar el rendimiento en la detección de diabetes.

## Reflexión

El uso de AutoGluon y AutoML permitió entrenar y comparar múltiples modelos de clasificación de manera rápida y automatizada, obteniendo un ensamble como mejor resultado sin necesidad de programar cada algoritmo desde cero. Esta capacidad ofrece una ventaja importante siendo esta el ahorro de tiempo que permite realizar una exploración sistemática de modelos y obtener desempeños competitivos sin necesidad de hiperajustes manuales extensivos.

Sin embargo el proceso funciona como una caja negra en la cual se pierde parte del control sobre las decisiones de modelado y coimplica la interpretabilidad de los resultados. Además, el tiempo de cómputo puede ser alto al usar presets de máxima calidad.

En este problema de predicción de diabetes, métricas como la precisión global son útiles, pero el recall y el F1-score de la clase positiva son más eficientes, ya que la prioridad clínica es minimizar falsos negativos. El **AUC-ROC** también es valioso para evaluar la capacidad discriminativa del modelo independientemente del umbral de decisión.

Al aplicar AutoML en salud, deben tomarse precauciones éticas y prácticas como:

- Validar los resultados con expertos clínicos.
- Evitar depender únicamente de métricas globales.
- Considerar la importancia de la interpretabilidad y la transparencia de los modelos.
- Usar estos sistemas como apoyo a la decisión, no como reemplazo del juicio médico.

En comparación con la construcción manual de un modelo, la experiencia con AutoGluon muestra que AutoML acelera el proceso exploratorio y logra modelos competitivos, pero el enfoque manual ofrece mayor control, comprensión del funcionamiento interno y capacidad de justificar decisiones. En este caso, el mejor balance surge al combinar ambas aproximaciones como usar AutoML para exploración inicial y ensamble, y complementar con modelos manuales más interpretables para aplicaciones críticas como la salud.