

Actividad 1. Momento 2: Análisis de Bases de Datos y Experimentación

Deep Learning

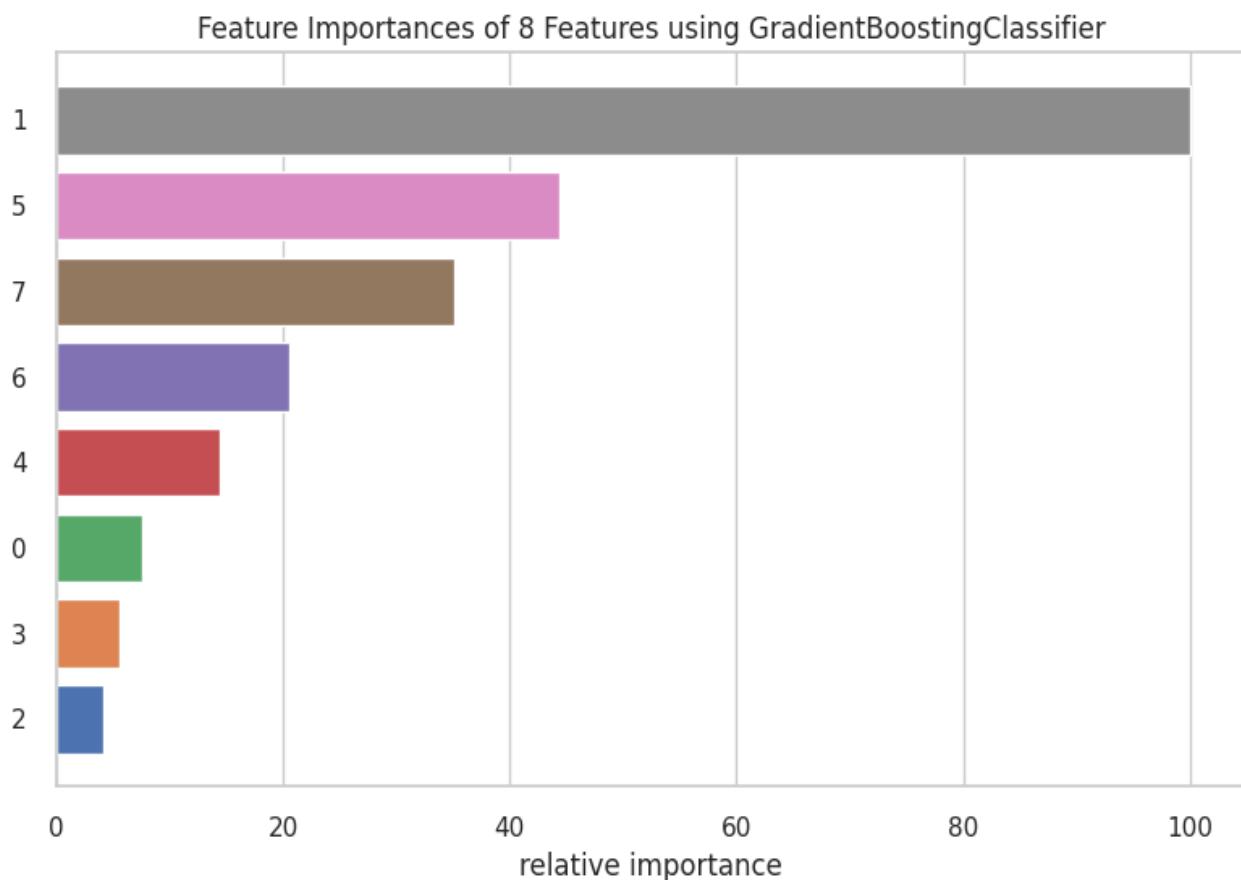
Análisis de bases de datos y experimentación con diferentes modelos de Machine Learning

- 1) La base de datos seleccionada para este ejercicio fue la que contiene información sobre pacientes con posibles casos de diabetes con variables como el número de embarazos de la paciente, el nivel de glucosa en sangre, la presión arterial diastólica, el grosor del pliegue cutáneo, el nivel de insulina en sangre, el Índice de Masa Corporal, la función que estima la probabilidad de diabetes según antecedentes familiares, la edad y la variable objetivo que en este caso es **Outcome** e indica si la paciente tiene diabetes (1) o no (0) por lo que será nuestra etiqueta en los distintos modelos.
- 2) Modelos:
 - a. **Decision Tree Classifier (DTC)**: el {árbol de decisión clasifica datos dividiéndolos en ramas según reglas. Cada nodo representa una pregunta y las hojas contienen la clase final. Usa criterios como Gini o Entropía para encontrar la mejor división. Es interpretable, pero propenso al sobreajuste si no se limita la profundidad o se aplica poda.
 - b. **Multilayer Perceptron (MLP)**: red neuronal con capas de entrada, ocultas y salida. Usa funciones de activación como ReLU y aprende con backpropagation. Es útil para problemas complejos, pero requiere muchos datos y normalización.
 - c. **K-Nearest Neighbors (KNN)**: clasifica observando los K vecinos más cercanos según una métrica de distancia. Es fácil de entender y no necesita entrenamiento previo, pero se vuelve lento con muchos datos y es sensible a la escala.
 - d. **Stochastic Gradient Descent Classifier (SGDC)**: optimiza modelos lineales como Regresión Logística mediante descenso de gradiente estocástico. Es eficiente en grandes datasets, pero puede ser inestable si no se ajusta bien la tasa de aprendizaje (learning rate) y requiere normalización.
 - e. **Extra Trees Classifier (ETC)**: variante de Random Forest que selecciona divisiones de forma aleatoria, reduciendo la varianza y acelerando el entrenamiento. Es eficiente y menos propenso al sobreajuste, aunque puede perder algo de precisión.
 - f. **Random Forest (RF)**: modelo de ensamble basado en la combinación de múltiples árboles de decisión entrenados con bagging. Mejora precisión y reduce sobreajuste, pero puede ser costoso computacionalmente con muchas características.

- g. **Gradient Boosting (GB)**: entrena árboles secuenciales donde cada uno corrige errores del anterior mediante descenso de gradiente. Es potente, pero más lento que Random Forest y propenso al sobreajuste si no se ajustan bien los hiperparámetros.

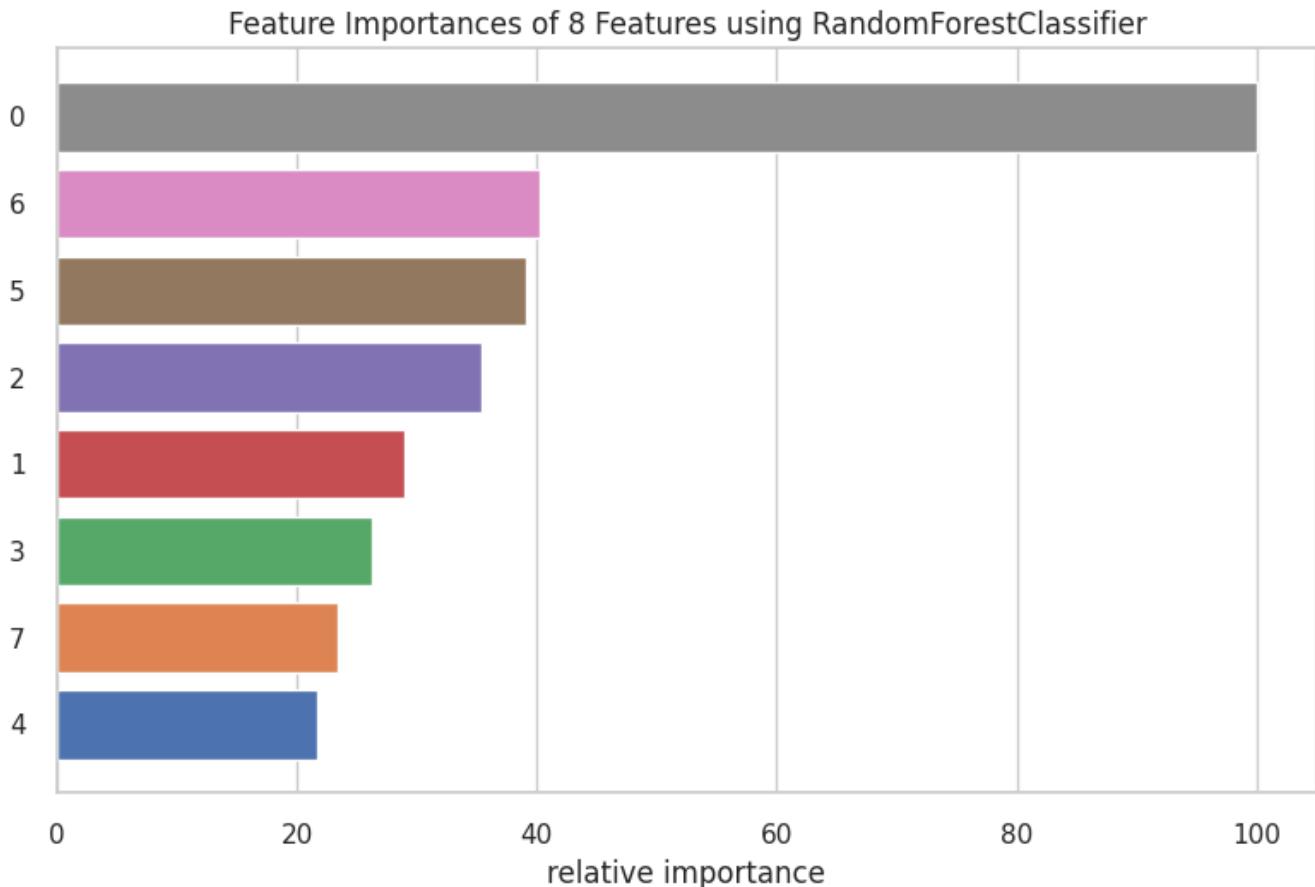
3) Resultados:

Any: En este caso los modelos con mayor Accuracy fueron ExtraTreesClassifier y GradientBoostingClassifier con el 75.97% seguidos de RandomForestClassifier con el 74.67%. El modelo que se decidió experimentar fue GB y los mejores hiperparámetros obtenidos fueron: learning_rate= 0.01, max_depth= 4, n_estimators= 200.



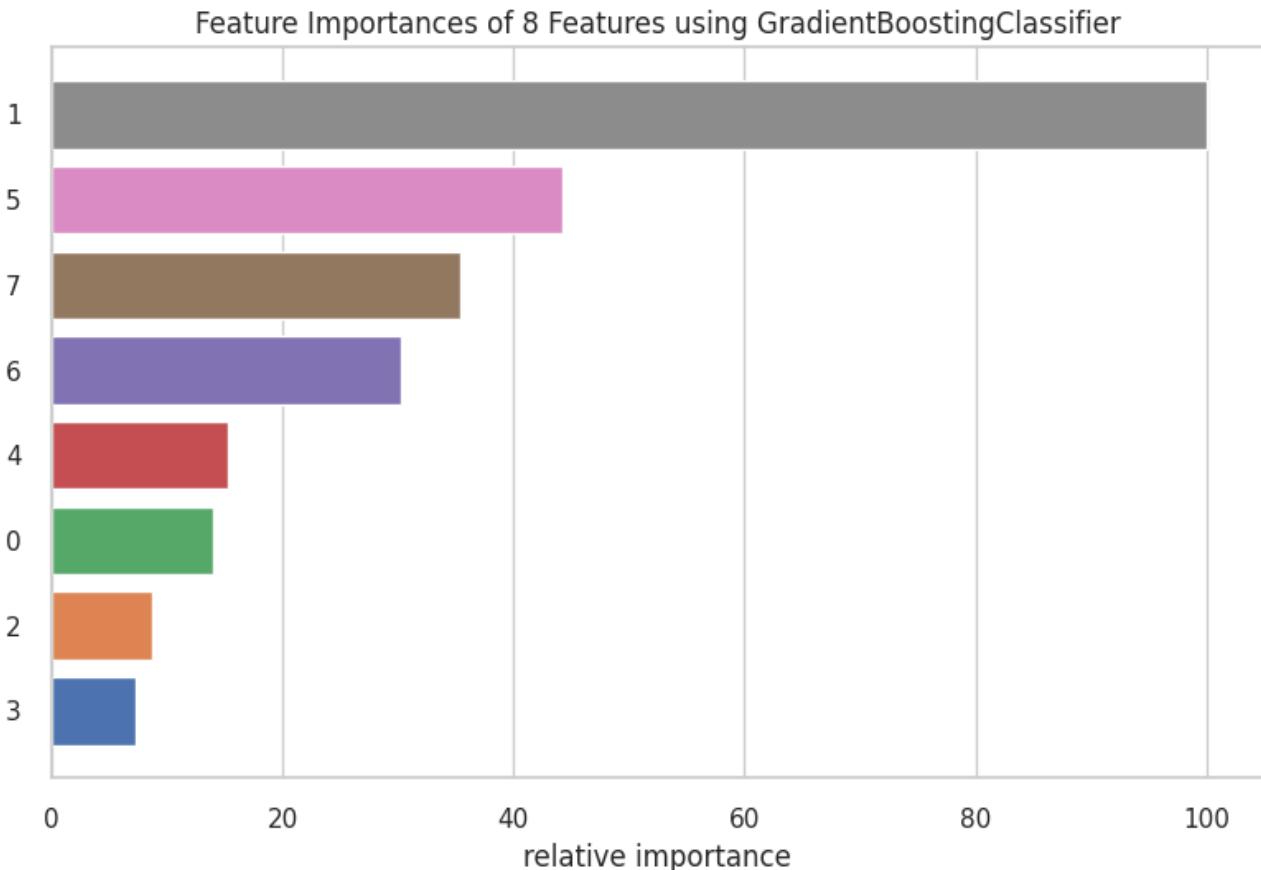
La gráfica revela un desequilibrio significativo, donde la característica 1 domina claramente con casi el 100% de importancia relativa, seguida por las características 5 y 7 con importancias moderadas (45% y 35% respectivamente), mientras que las demás tienen una influencia mínima por debajo del 20%. Esta distribución indica que el modelo depende excesivamente de una sola variable para sus predicciones, lo que podría señalar un posible sobreajuste, una variable con alto poder predictivo o incluso una potencial fuga de datos.

PCA Sin Normalizar: en este caso el modelo con mayor Accuracy fue RandomForestClassifier con el 77.27%, seguido de ExtraTreesClassifier y MLPClassifier con el 75.32% y también de GradientBoostingClassifier con el 74.67%, por lo que el modelo que se decidió experimentar fue RF y los mejores hiperparámetros obtenidos fueron: max_depth= 20, min_samples_leaf= 4, min_samples_split= 10, n_estimators= 50.



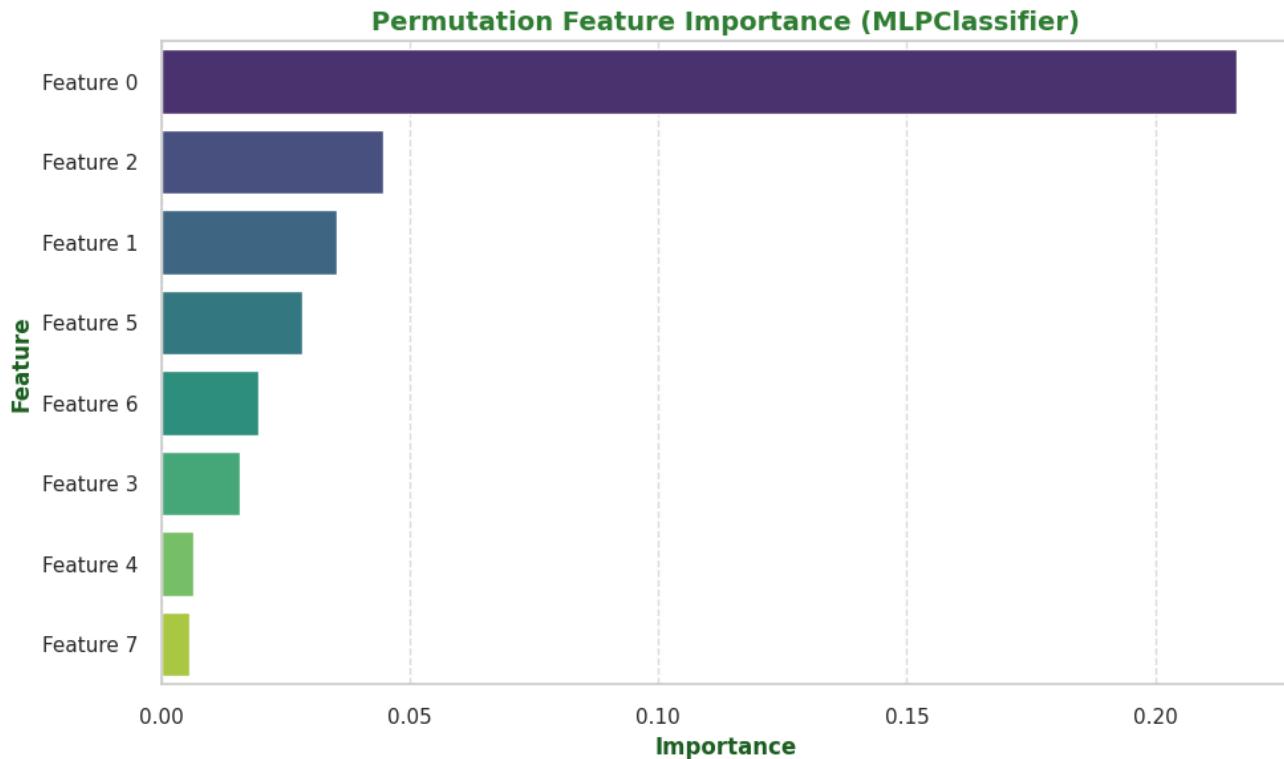
Se observa una distribución poco equilibrada pues la característica 0 tiene la mayor importancia (100%), seguida por la característica 6 (40%), mientras que las características 5 y 2 presentan valores significativamente menores (aproximadamente 40%). Esta distribución poco balanceada sugiere que el modelo RandomForest sigue la tendencia de que el modelo depende excesivamente de una sola variable para sus predicciones, lo que podría señalar un posible sobreajuste, una variable con alto poder predictivo o incluso una potencial fuga de datos.

Sin PCA Normalizado: en este caso los modelos con mayor Accuracy fueron ExtraTreesClassifier y GradientBoostingClassifier con el 75.97% seguidos de RandomForestClassifier con el 75.32%. El modelo que se decidió experimentar fue GB y los mejores hiperparámetros obtenidos fueron: learning_rate= 0.01, max_depth= 4, n_estimators= 200.



Se observa una distribución muy desigual incluso después del preprocesamiento de los datos, donde la característica 1 domina completamente con una importancia cercana al 100%, lo que indica que es el factor más determinante en las predicciones del modelo. La característica 5 ocupa el segundo lugar con aproximadamente un 45% de importancia, seguida por la característica 7 (35%) y la característica 6 (30%). Las características restantes tienen una importancia considerablemente menor: la característica 4 (15%), la característica 0 (15%), la característica 2 (10%) y la característica 3 (8%).

PCA Normalizada: en este caso el modelo con mayor Accuracy fue MLPClassifier con el 75.97% seguido de ExtraTreesClassifier con el 75.32% y RandomForestClassifier con el 74.67, por lo que el modelo que se decidió experimentar fue MLP y los mejores hiperparámetros obtenidos fueron: activation= tanh, hidden_layer_sizes= (50,), learning_rate_init= 0.001, max_iter= 200



En este caso la gráfica muestra la importancia de características mediante permutación para el modelo MLPClassifier y se observa una distribución muy desequilibrada donde la característica 0 domina completamente con una importancia de aproximadamente 0.20 (o 20%), siendo cuatro veces más importante que la siguiente característica en relevancia. La característica 2 que ocupa el segundo lugar con un valor cercano a 0.05, seguida por la característica 1 (0.04), la característica 5 (0.03), y la característica 6 (0.02). Las características 3, 4 y 7 tienen una importancia mínima (por debajo de 0.02). Este patrón de importancia revela que el modelo MLPClassifier depende fundamentalmente de la característica 0 para sus predicciones, mostrando un patrón similar al observado en los modelos anteriores donde una sola variable tiene un peso desproporcionadamente alto en el rendimiento predictivo.

Tras haber visto el comportamiento de los mejores modelos para cada caso podemos darnos cuenta que:

- En el GradientBoostingClassifier en la primera gráfica, la característica 1 era claramente dominante. Esto correspondería a la columna "Glucose", lo cual tiene sentido ya que el nivel de glucosa en sangre es uno de los indicadores más directos

para diagnosticar diabetes. Las características 5 y 7 (que serían "BMI" y "Age") aparecían como la segunda y tercera más importantes, también consistente con factores de riesgo conocidos para la diabetes.

- En la gráfica de RandomForestClassifier se mostraba la característica 0 ("Pregnancies") como la más importante, seguida de lejos por las características 6,5 y 2 ("DiabetesPedigreeFunction", "BMI", y "BloodPresure"). Esto es interesante porque sugiere que el RandomForest da más peso a los antecedentes de embarazos que el GradientBoosting, posiblemente captando la relación con la diabetes gestacional, pero también teniendo en cuenta la salud del corazón, el estado físico y la presión sanguínea como factores directos.
- Para el GradientBoostingClassifier sin PCA normalizado con 8 características, de manera similar, la característica 1 ("Glucose") dominaba, mientras que las características 5 y 7 ("BMI" y "Age") ocupaban el segundo y tercer lugar, lo que refleja edad y de condición física.
- Finalmente, para MLPClassifier (Red Neuronal) también se identificaba la característica 0 ("Pregnancies") como la más importante, seguida por la 2 ("BloodPressure"), lo que sugiere que la red neuronal encontró patrones diferentes en los datos, dando mayor relevancia a la presión arterial que otros modelos.

Es notable cómo diferentes algoritmos priorizan distintas variables para predecir la diabetes. Mientras que GradientBoosting se enfoca más en los niveles de glucosa, los modelos de RandomForest y MLP parecen dar mayor importancia al historial de embarazos. Esto demuestra la importancia de probar múltiples modelos, ya que cada uno puede captar diferentes relaciones en los datos.

Tabla:

Data	Model	Accuracy %	Precision %	Recall %	F1 %
Any	DTC	72.08	71.26	72.08	71.41
	MLP	71.43	71.29	71.43	67.80
	KNN	72.73	72.01	72.73	72.16
	SGDC	68.83	69.71	68.83	69.17
	ETC	75.97	75.43	75.97	74.87
	RF	74.68	74.01	74.68	74.07
PCA	GB	75.97	75.46	75.97	75.55
	DTC	68.18	67.56	68.18	67.80
	MLP	75.32	75.03	75.32	73.64
	KNN	70.78	69.76	70.78	69.88
	SGDC	71.43	70.33	71.43	70.24
	ETC	75.32	74.65	75.32	74.48
Standard Scaler	RF	77.27	76.76	77.27	76.58
	GB	74.68	74.10	74.68	74.23
	DTC	72.73	72.01	72.73	72.16

	MLP	72.73	72.15	72.73	72.32
	KNN	72.08	71.56	72.08	71.74
	SGDC	70.78	70.66	70.78	70.72
	ETC	75.97	75.43	75.97	74.87
	RF	75.32	74.68	75.32	74.65
	GB	75.97	75.46	75.97	75.55
PCA + StandardScaler	DTC	72.08	72.20	72.08	72.14
	MLP	75.97	75.39	75.97	75.40
	KNN	72.08	71.56	72.08	71.74
	SGDC	70.78	70.66	70.78	70.72
	ETC	75.32	74.65	75.32	74.48
	RF	74.68	73.97	74.68	73.52
	GB	73.38	72.75	73.38	72.90

Conclusiones:

- El análisis de los modelos muestra que los modelos Extra Trees Classifier (ETC) y Gradient Boosting (GB) ofrecen el mejor rendimiento general con un 75.97% de exactitud, además de un buen equilibrio entre precisión y recall. Esto los convierte en las opciones más confiables para la predicción de diabetes en este conjunto de datos. Por otro lado, Random Forest (RF) con PCA alcanza una exactitud del 77.27%, lo que sugiere que este modelo podría beneficiarse de la reducción de dimensionalidad en ciertos casos.
- La aplicación de PCA no mostró mejoras significativas para la mayoría de los modelos y, en algunos casos, redujo su desempeño. Decision Tree Classifier (DTC), por ejemplo, experimentó una caída en su exactitud al aplicar PCA, lo que indica que la reducción de dimensiones eliminó información relevante para su clasificación. Sin embargo, en el caso de Random Forest, PCA ayudó a mejorar su rendimiento, lo que sugiere que algunos modelos basados en árboles pueden beneficiarse de esta técnica.
- El uso de Standard Scaler mejoró el desempeño de modelos sensibles a la escala de los datos, como K-Nearest Neighbors (KNN), Stochastic Gradient Descent Classifier (SGDC) y Multi-Layer Perceptron (MLP). En particular, la combinación de PCA y Standard Scaler permitió que MLP alcanzara una precisión de 75.97%, lo que la hace una alternativa viable si se busca mejorar el desempeño de redes neuronales en este conjunto de datos.
- En general, ETC y GB con Standard Scaler son las mejores opciones si se busca un modelo estable y con buen rendimiento en la predicción de diabetes. Si se desea explorar técnicas de reducción de dimensiones, Random Forest con PCA es la opción más prometedora. Sin embargo, la combinación de PCA y Standard Scaler no aportó mejoras significativas en la mayoría de los modelos, por lo que su uso dependerá del enfoque particular del análisis.