Proyecto :
Detección de síndrome del ojo seco empleando atributos y variables medicas

- Greyler Jose Chacon Chaparro (E2)
- Jonk Keyler Sanchez Pabon (E2)
- Samuel David Traslaviña Mateus (F1)

Contenido de la presentación

01 Presentación del problema

02 Preprocesamiento de los datos

03 Modelos y su evaluación

04 Curvas de aprendizaje

01

Presentación del problema

Dataset

Se cuenta con el dataset *Dry Eye Disease* , que se encuentra disponible en kaggle. El *dataset* contiene información sobre los hábitos de sueño, actividad física y factores de salud de una muestra de persona, con el objetivo de analizar su relación con la enfermedad del ojo seco (*Dry Eye Disease*).

# Average screen time	# Blue-light filter	# Discomfort Eye-strain	# Redness in eye	# Itchiness/Irritation in eye	# Dry Eye Disease
Missing: 0 (0%) Distinct: 91 (<1%)	Missing: 0 (0%) Distinct: 2 (<1%)	Missing: 0 (0%) Distinct: 2 (<1%)	Missing: 0 (0%) Distinct: 2 (<1%)	Missing: 0 (0%) Distinct: 2 (<1%)	Missing: 0 (0%) Distinct: 2 (<1%)
Min 1.0 Max 10.0	Min 0 Max 1	Min 0 Max 1	Min 0 Max 1	Min 0 Max 1	Min 0 Max 1
9.6	1	0	0	1	1
4.0	0	1	0	0	0
7.6	0	1	0	1	0
3.5	0	1	1	0	1
2.6	1	1	0	1	1
5.4	1	0	0	1	1
4.5	1	1	0	0	1
3.9	1	0	0	1	0
5.7	0	0	1	1	1
7.9	1	0	1	1	1
10.0	1	0	1	0	1
6.3	1	1	0	1	1
6.5	0	1	1	0	1
7.0	1	1	1	0	1
6.2	1	1	0	0	1
6.3	1	0	1	0	0
6.1	0	1	0	1	1
4.5	1	1	1	0	1
3.1	0	0	1	0	1

Problema

La enfermedad puede tener diferentes factores de riesgos que pueden estar relacionados a los hábitos que tiene la persona , como tiempo de exposición a pantallas , actividad física , edad , antecedentes médicos ,entre otros. Se cuenta con el *dataset* con un total de 26 *features* , entonces , se decidió entrenar modelos de *Machine learning* para **clasificar** si un paciente , dadas todas las variables , tiene o no la enfermedad (**clasificación binaria**) bajo un **enfoque supervisado**.

02

Preprocesamiento de los datos

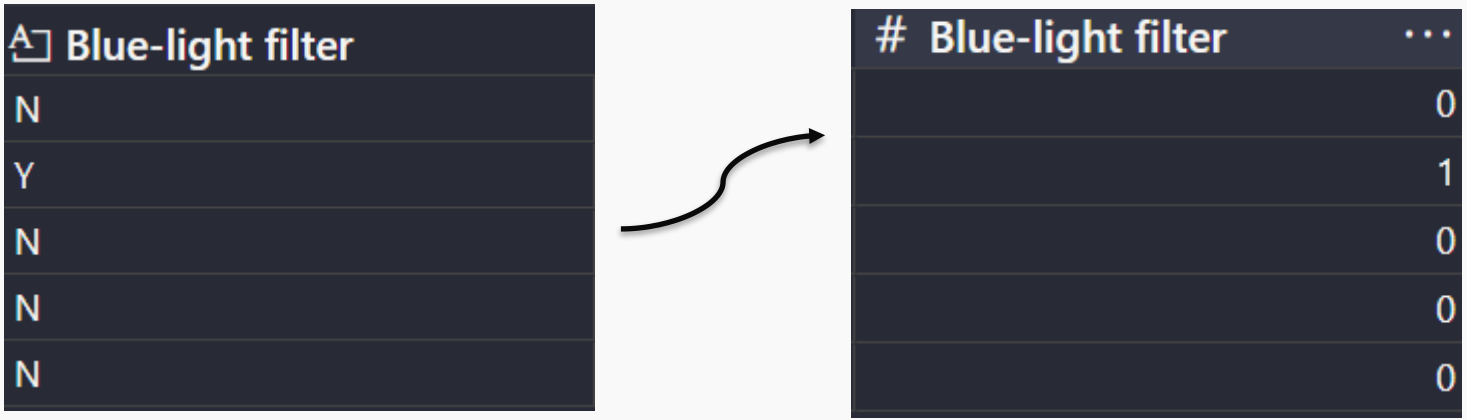
¿Por qué el preprocesamiento?

En la mayoría de las ocasiones que se cuenta con un proyecto de *machine learning* , los datos con los cuales se desea que el modelo aprenda y generalice para la tarea que se desea atacar , cuando están en bruto , pueden contener valores faltantes , algunas columnas se encuentran en un tipo de dato que los modelos no pueden utilizar , como texto y también existen modelos que son sensibles al escalado de los datos.

Para este proyecto se implementó la estrategia de ***label enconder*** y ***feature scaling***, ya que no se contaban con valores faltantes en los datos.

Label Enconder

Esta transformación identifica todos los distintos valores que se tienen para esa *feature* ,que se denominan clases, y luego se les asignan un número entero entre el rango de [0 - #clases-1].

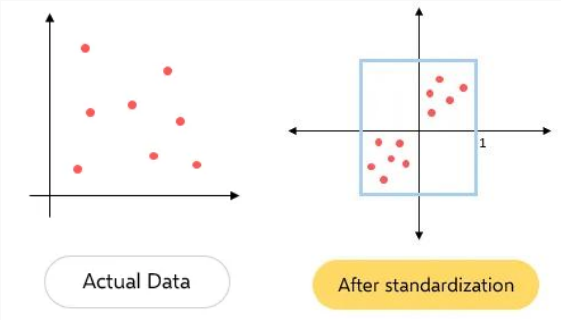


Estandarización

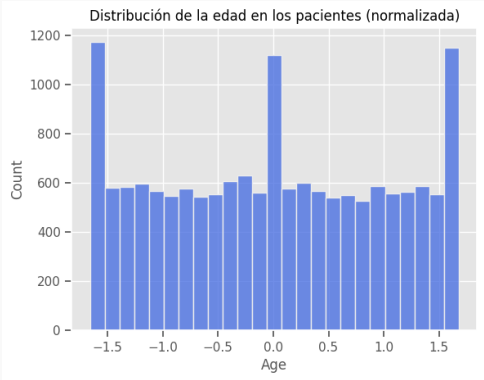
Con muy pocas excepciones , los algoritmos de *machine learning* , no funcionan bien cuando los datos numéricos se encuentran en diferentes escalas , La estandarizacion se hace necesaria cuando los conjuntos de datos contienen diferentes unidades de medición, rangos y magnitud , Si se usa, el modelo conducirá a un funcionamiento sesgado , La estandarización se define así :

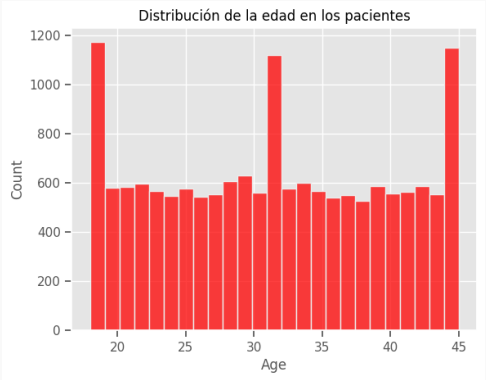
$$z_i = \frac{x_i - \overline{x_i}}{s_i}, i = 1, \ldots, n$$

Donde $\overline{x_i}$, es la media de la i-ésima *feature* y s_i su respectiva desviación



Realmente para los modelos que se usaron , solo la *SVM* , se le hace necesario la estandarización de las *features* antes de ajustar el modelo , ya que , sin una escala homogénea, características de gran varianza sesgan el espacio de características y el hiperplano óptimo.



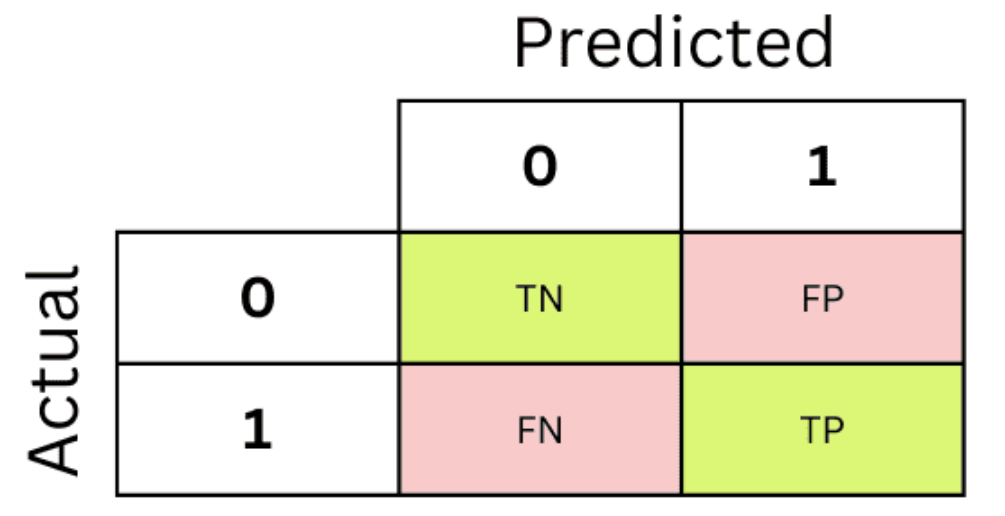


03

Modelos y su evaluación

Métricas

La **matriz de confusión** , mas que ser una métrica en si , es una herramienta que permite contar el número de veces que las instancias de la clase A se clasifican como clase B, para todos los pares A/B



El ***accuracy*** es una métrica fundamental para evaluar el rendimiento de un modelo de clasificación, proporcionando una vista rápida de qué tan bien se está desempeñando el modelo en términos de predicciones correctas. Se calcula como la relación de predicciones correctas al número total de muestras de entrada :

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Se debe tener la precaución cuando se trabaja con clases desbalanceadas.

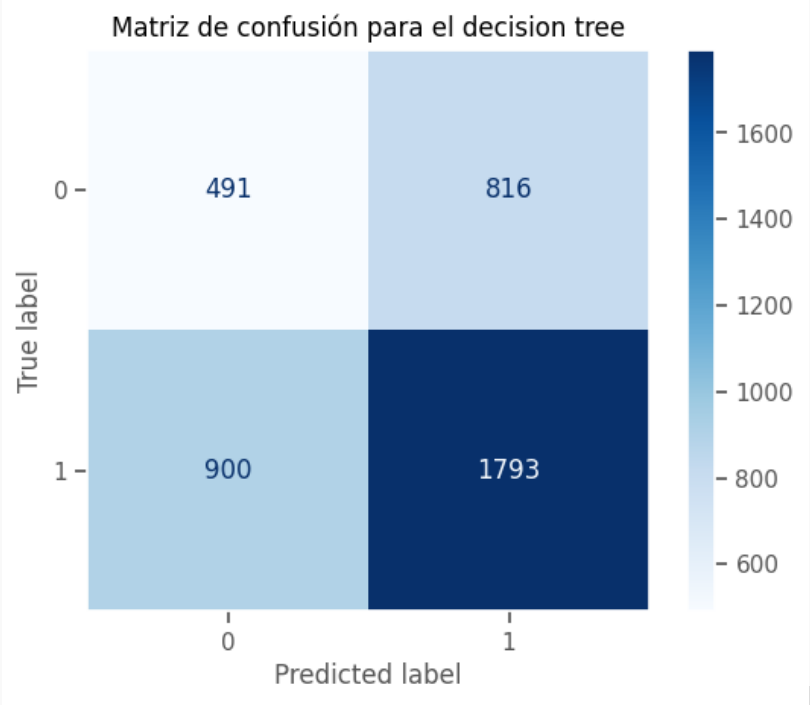
Otra métrica que fue empleada fue el **recall** , en el problema de detección del síndrome del ojo seco, porque se tratando de identificar correctamente a los pacientes que sí tienen la condición, incluso si eso significa que el modelo ocasionalmente prediga algunos falsos positivos.

$$recall = \frac{TP}{TP + FN}$$

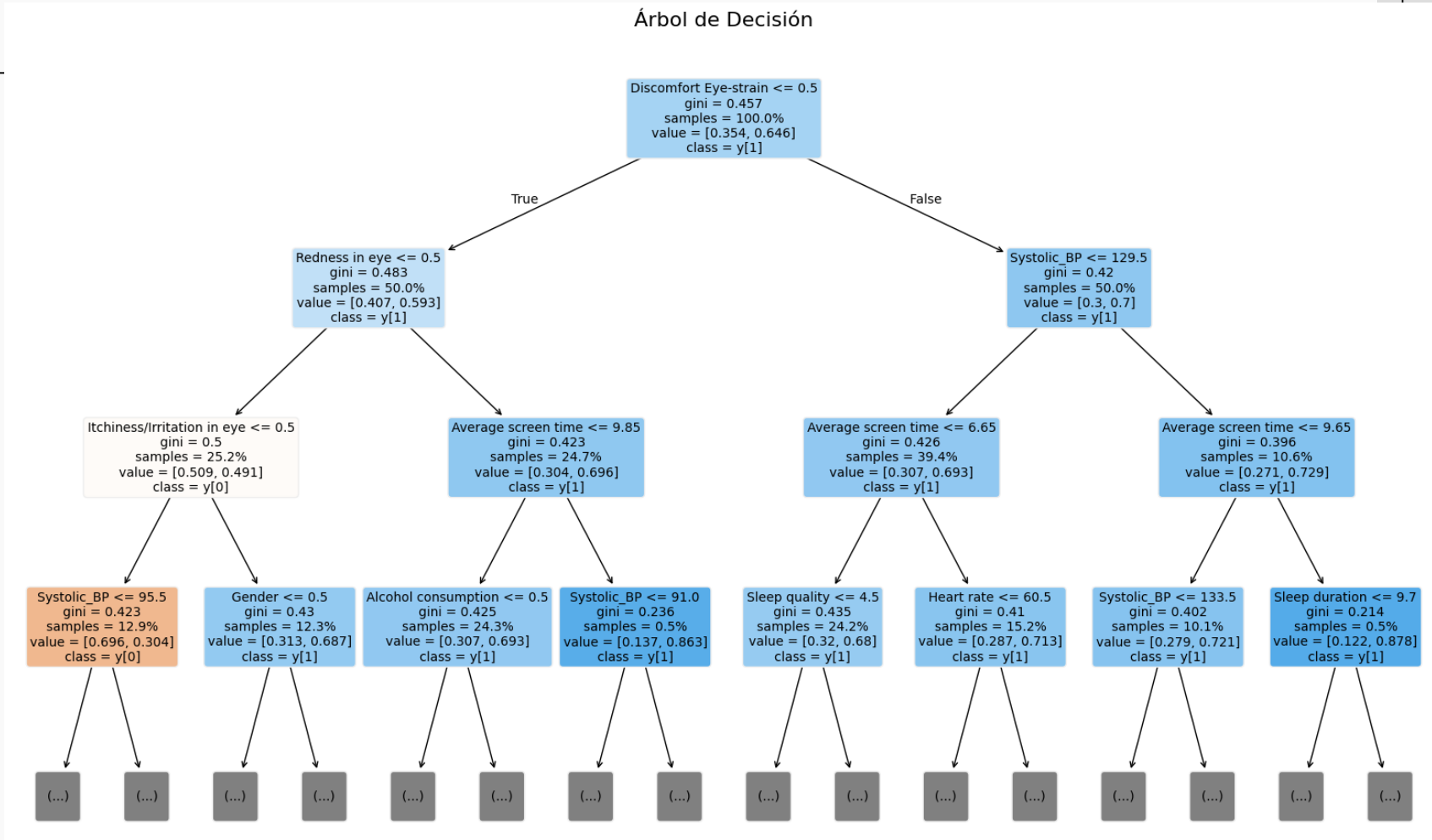
Gracias a la implementación de *scikit-learn* de `sklearn.metrics.classification_report` , nos permite también conocer la especificidad o tasa de verdaderos negativos :

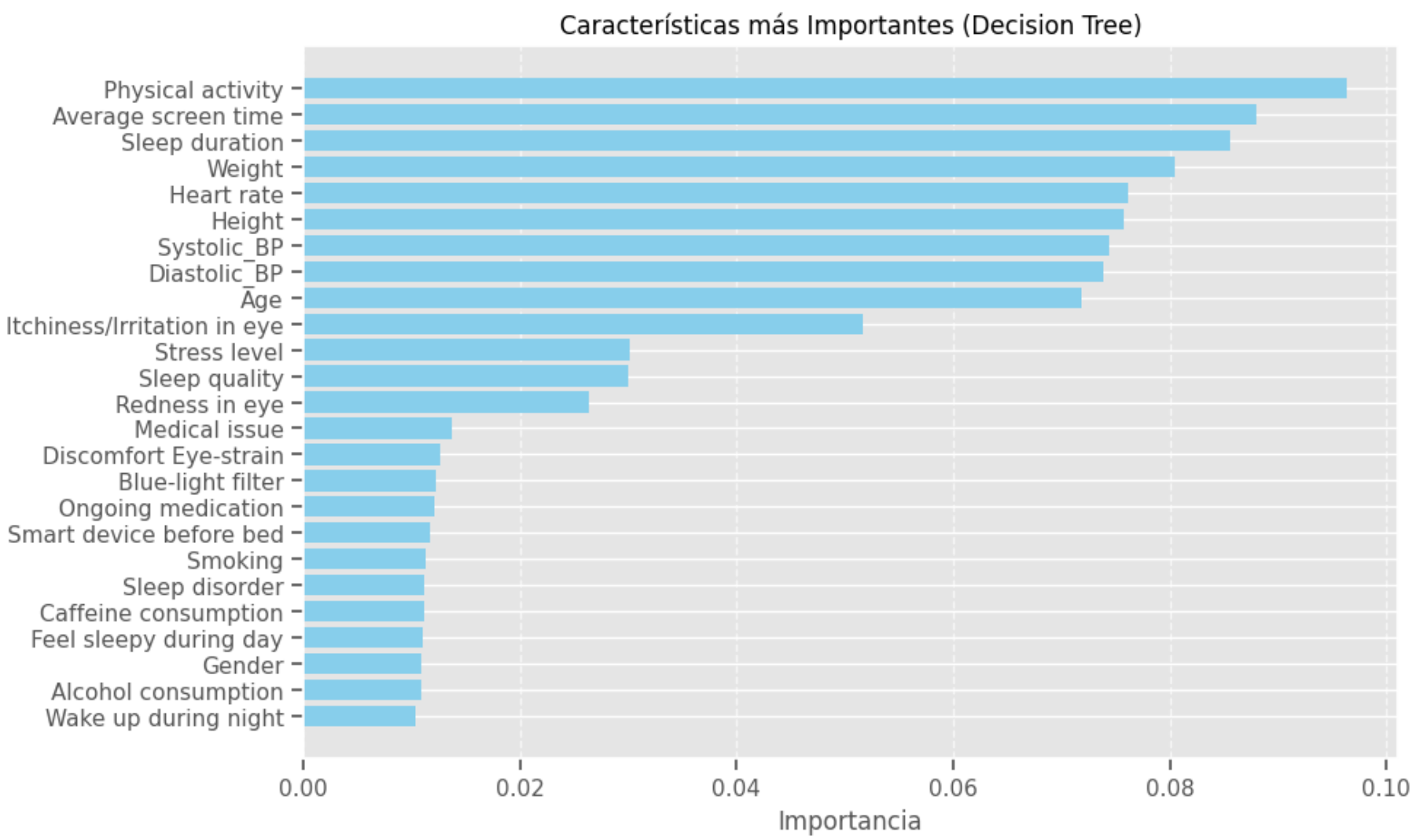
$$specificity = \frac{TN}{TN + FP}$$

Desicion Tree

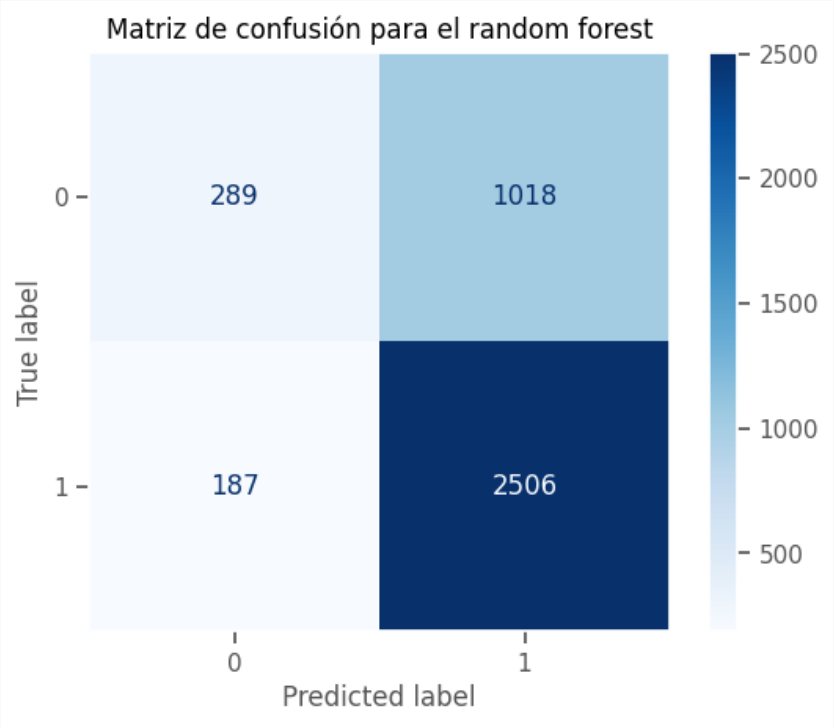


Accuracy: 0.571 , *Recall* : 0.38 , Specificity : 0.38

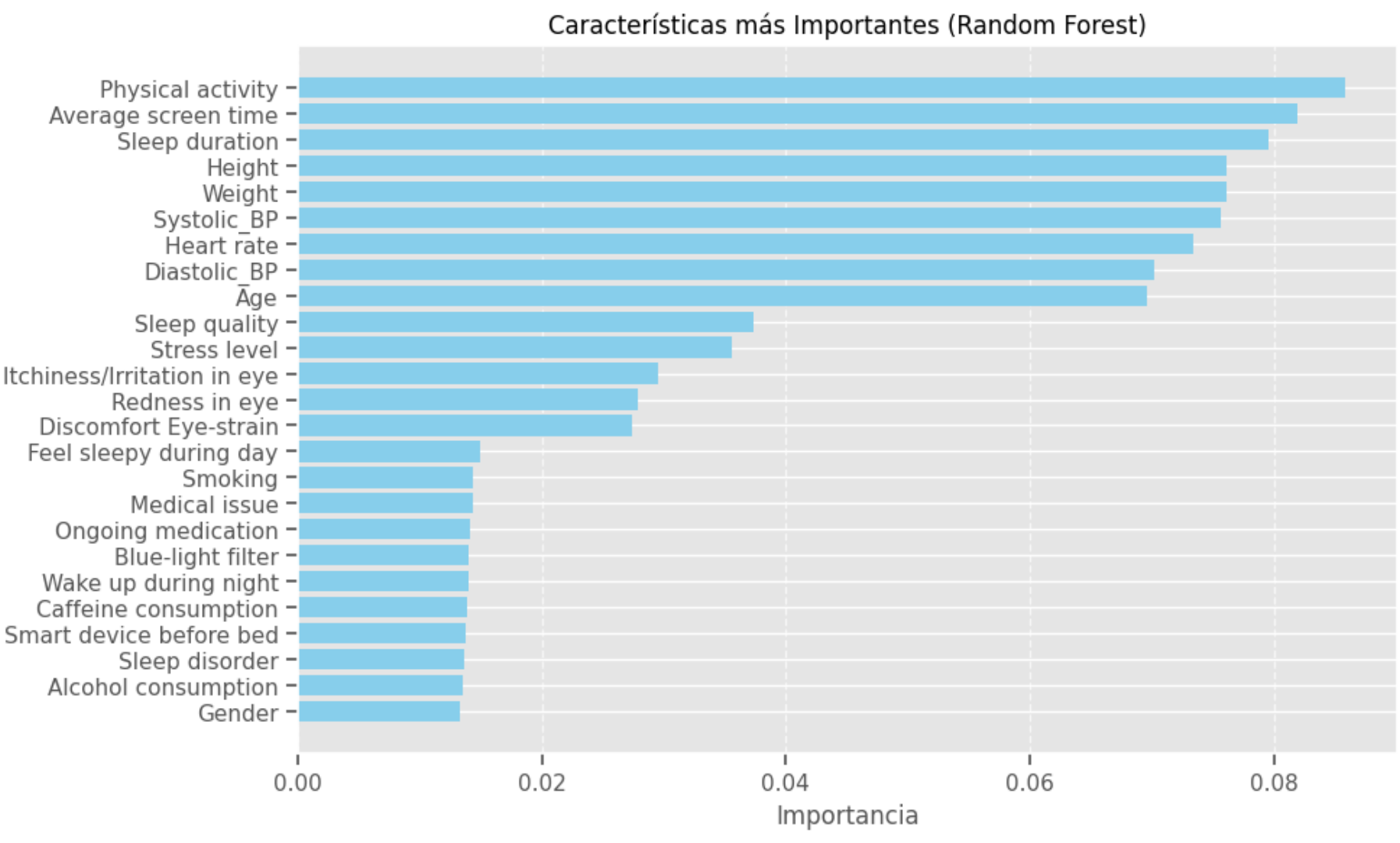

Árbol de Decisión
Discomfort Eye-strain <= 0.5
gini = 0.457
samples = 100.0%
value = [0.354, 0.646]
class = y[1]
True
False
Redness in eye <= 0.5
gini = 0.483
samples = 50.0%
value = [0.407, 0.593]
class = y[1]
Systolic_BP <= 129.5
gini = 0.42
samples = 50.0%
value = [0.3, 0.7]
class = y[1]
Itchiness/Irritation in eye <= 0.5
gini = 0.5
samples = 25.2%
value = [0.509, 0.491]
class = y[0]
Average screen time <= 9.85
gini = 0.423
samples = 24.7%
value = [0.304, 0.696]
class = y[1]
Average screen time <= 6.65
gini = 0.426
samples = 39.4%
value = [0.307, 0.693]
class = y[1]
Average screen time <= 9.65
gini = 0.396
samples = 10.6%
value = [0.271, 0.729]
class = y[1]
Systolic_BP <= 95.5
gini = 0.423
samples = 12.9%
value = [0.696, 0.304]
class = y[0]
Gender <= 0.5
gini = 0.43
samples = 12.3%
value = [0.313, 0.687]
class = y[1]
Alcohol consumption <= 0.5
gini = 0.425
samples = 24.3%
value = [0.307, 0.693]
class = y[1]
Systolic_BP <= 91.0
gini = 0.236
samples = 0.5%
value = [0.137, 0.863]
class = y[1]
Sleep quality <= 4.5
gini = 0.435
samples = 24.2%
value = [0.32, 0.68]
class = y[1]
Heart rate <= 60.5
gini = 0.41
samples = 15.2%
value = [0.287, 0.713]
class = y[1]
Systolic_BP <= 133.5
gini = 0.402
samples = 10.1%
value = [0.279, 0.721]
class = y[1]
Sleep duration <= 9.7
gini = 0.214
samples = 0.5%
value = [0.122, 0.878]
class = y[1]
(...)


Características más Importantes (Decision Tree)
Physical activity
Average screen time
Sleep duration
Weight
Heart rate
Height
Systolic_BP
Diastolic_BP
Age
Itchiness/Irritation in eye
Stress level
Sleep quality
Redness in eye
Medical issue
Discomfort Eye-strain
Blue-light filter
Ongoing medication
Smart device before bed
Smoking
Sleep disorder
Caffeine consumption
Feel sleepy during day
Gender
Alcohol consumption
Wake up during night
0.00
0.02
0.04
0.06
0.08
0.10
Importancia

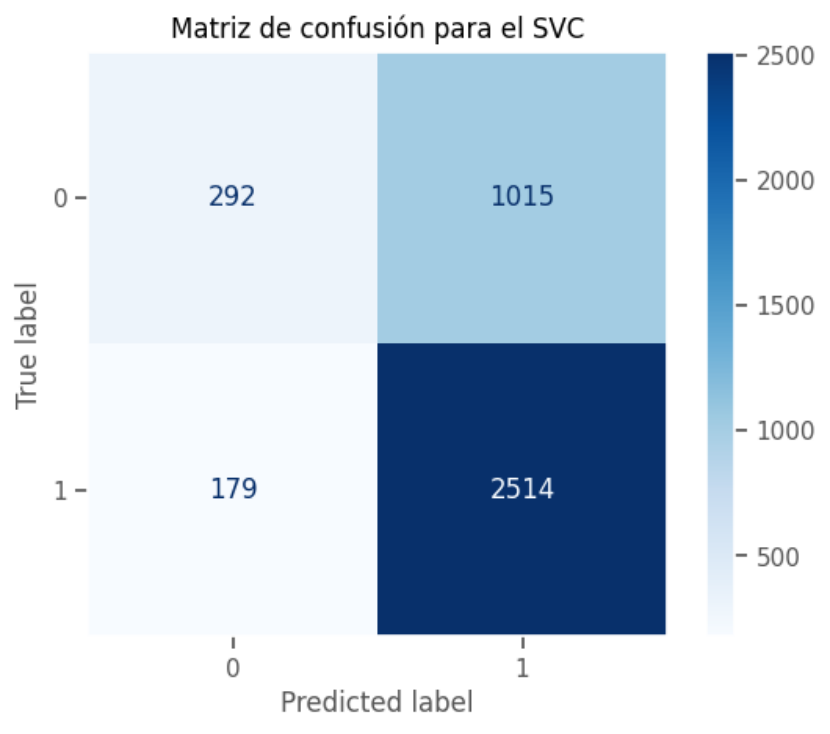
Random Forest



Accuracy: 0.699 , *Recall* : 0.93, Specificity : 0.22


Características más Importantes (Random Forest)
Physical activity
Average screen time
Sleep duration
Height
Weight
Systolic_BP
Heart rate
Diastolic_BP
Age
Sleep quality
Stress level
Itchiness/Irritation in eye
Redness in eye
Discomfort Eye-strain
Feel sleepy during day
Smoking
Medical issue
Ongoing medication
Blue-light filter
Wake up during night
Caffeine consumption
Smart device before bed
Sleep disorder
Alcohol consumption
Gender
0.00
0.02
0.04
0.06
0.08
Importancia

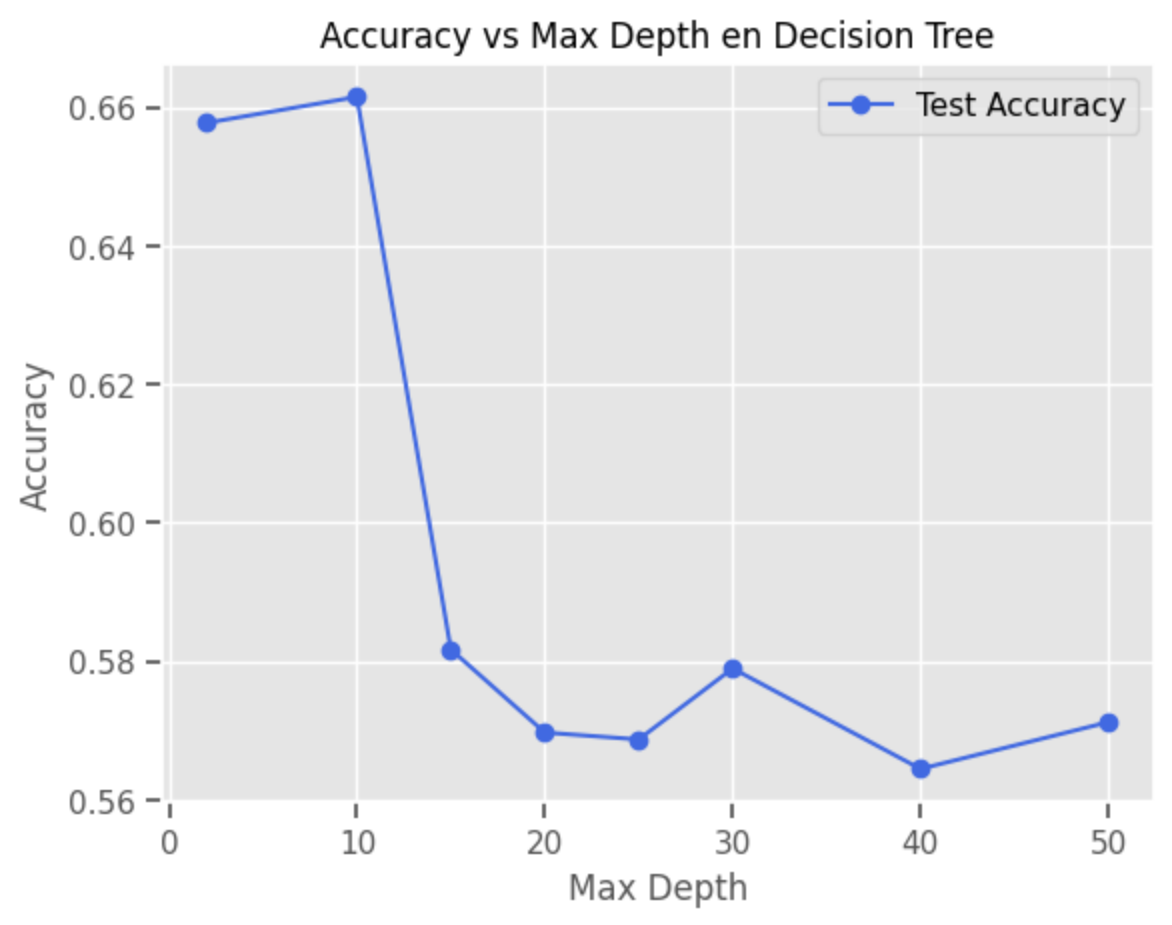
Support Vector Machine

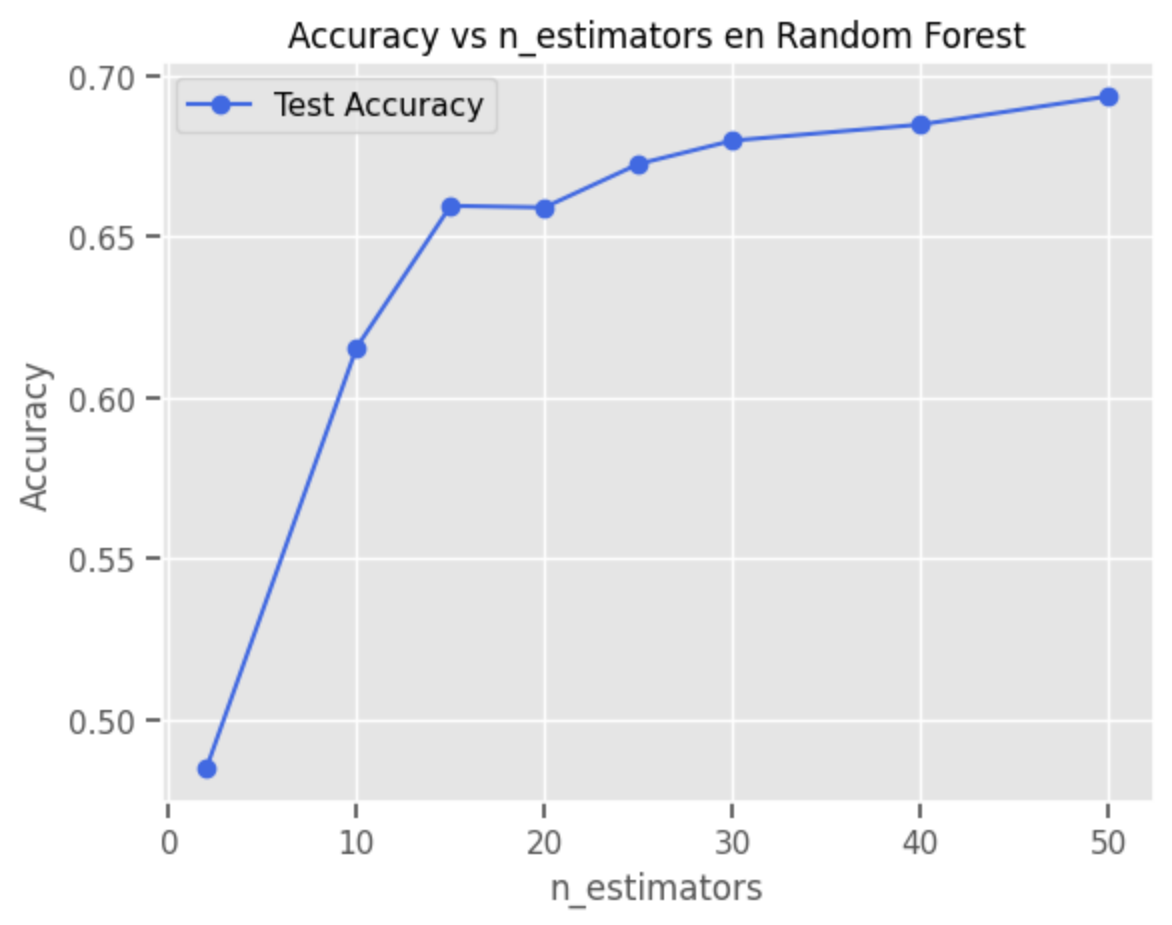


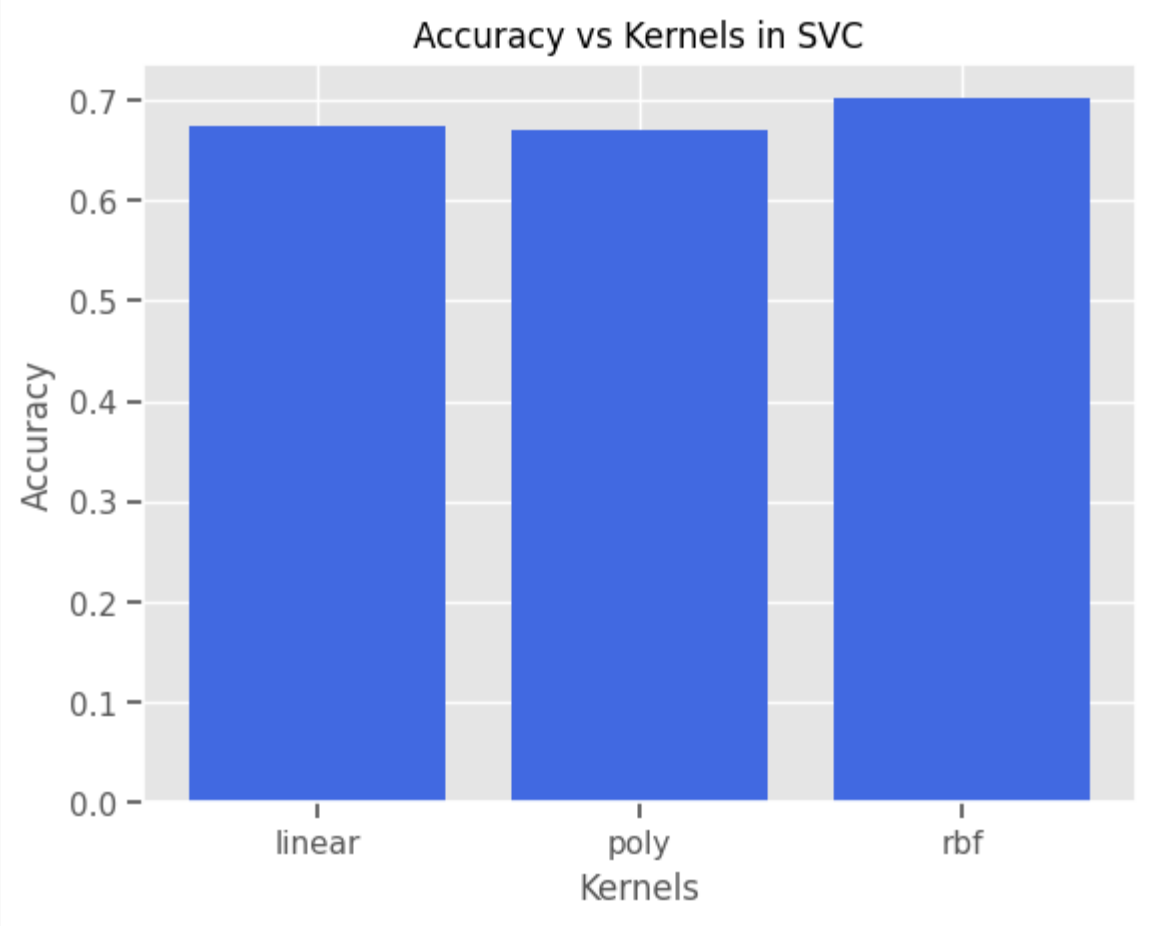
Accuracy: 0.70, *Recall* : 0.93, Specificity : 0.22

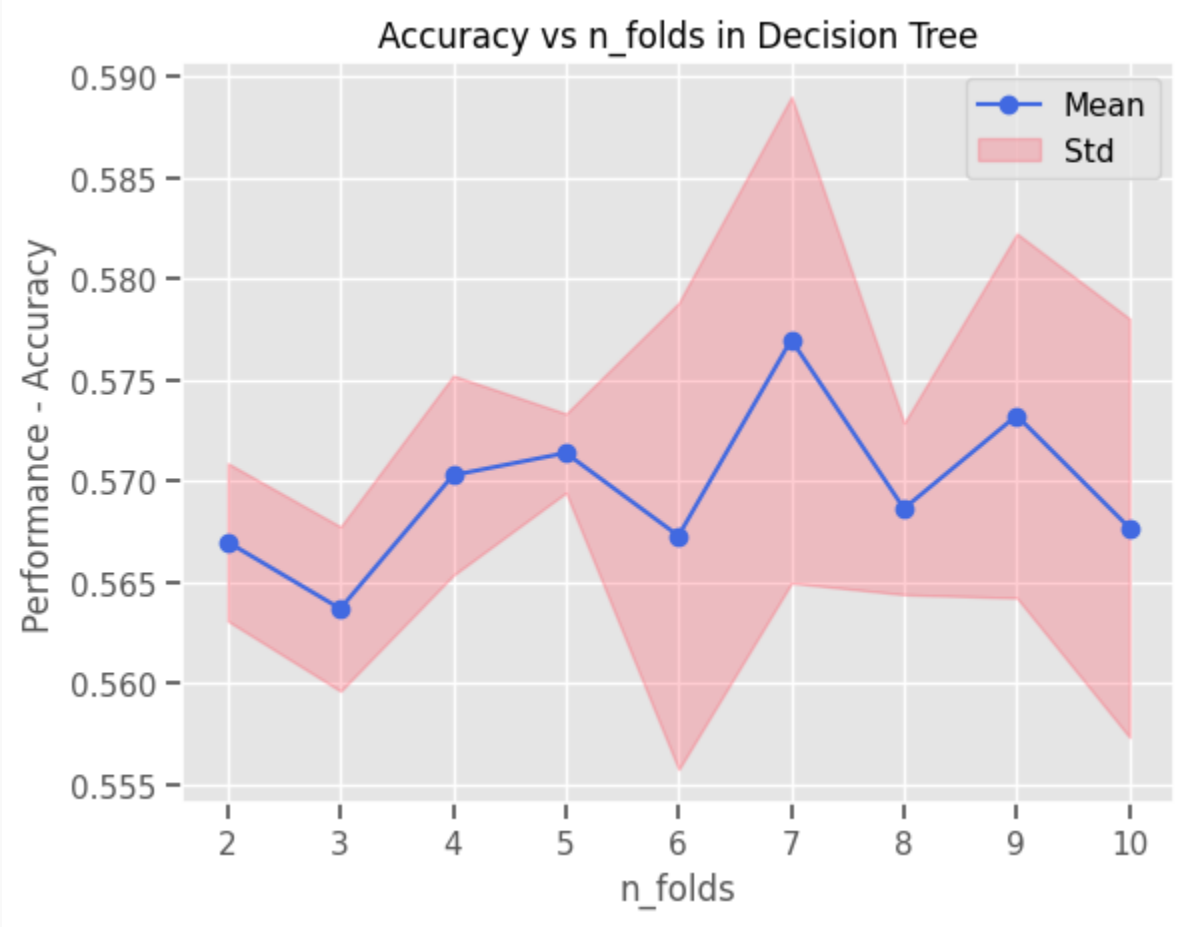
04

Curvas de aprendizaje


Accuracy vs Max Depth en Decision Tree
Test Accuracy
Accuracy
0.66
0.64
0.62
0.60
0.58
0.56
0
10
20
30
40
50
Max Depth


Accuracy vs n_estimators en Random Forest
Test Accuracy
Accuracy
0.70
0.65
0.60
0.55
0.50
0
10
20
30
40
50
n_estimators


Accuracy vs Kernels in SVC
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Accuracy
linear
poly
rbf
Kernels

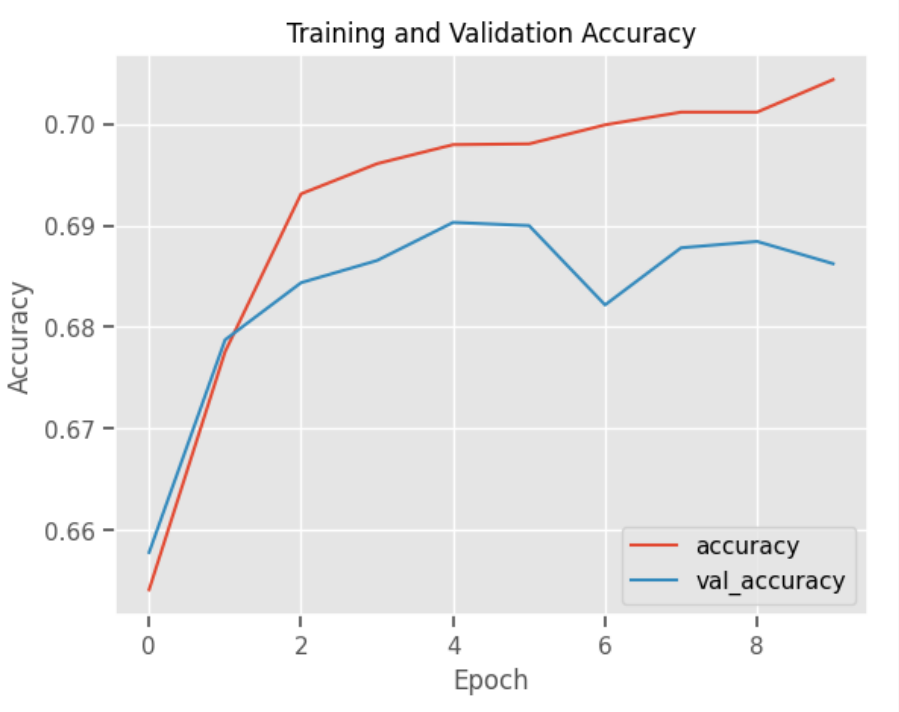

Accuracy vs n_folds in Decision Tree
Mean
Std
Performance - Accuracy
0.590
0.585
0.580
0.575
0.570
0.565
0.560
0.555
2
3
4
5
6
7
8
9
10
n_folds

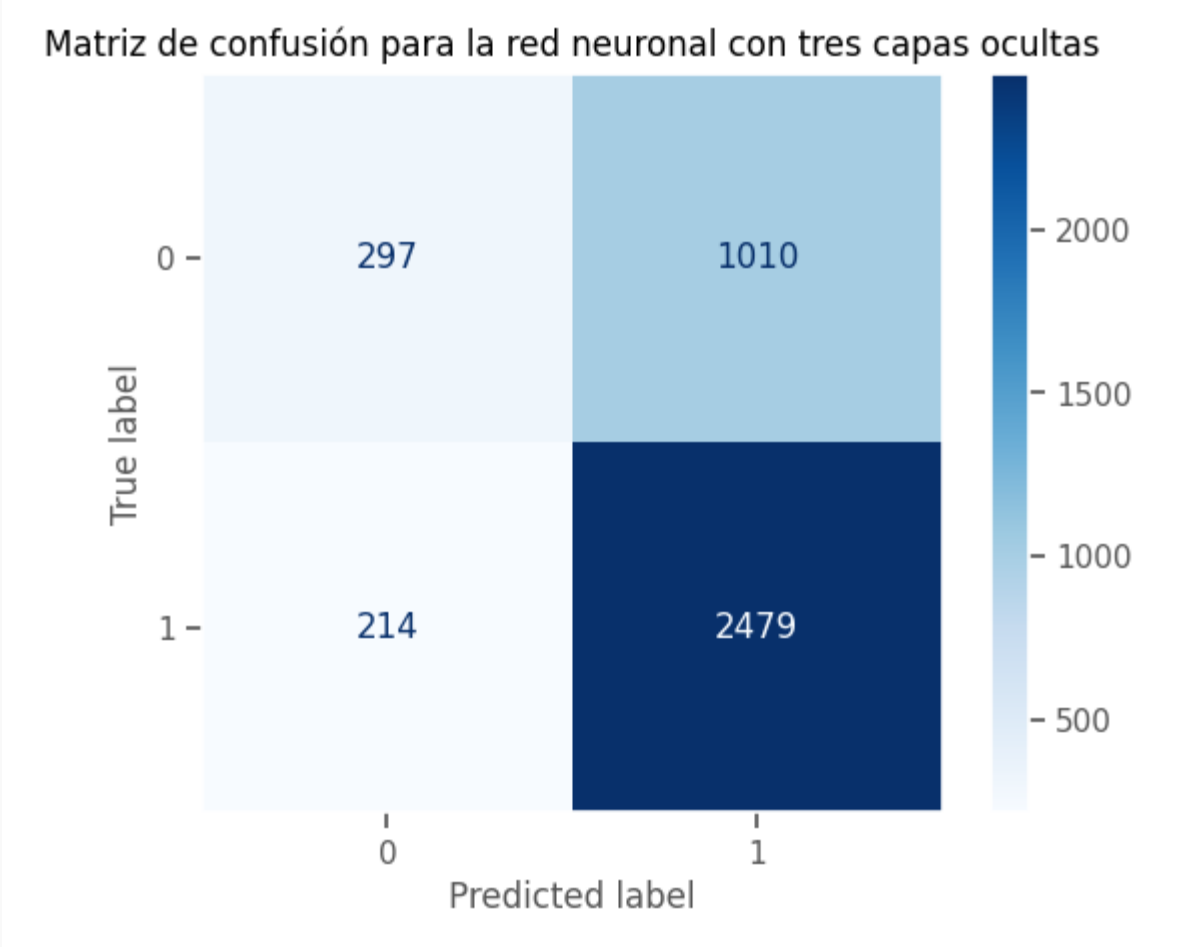
05

Deep Learning

La primera configuración fue :

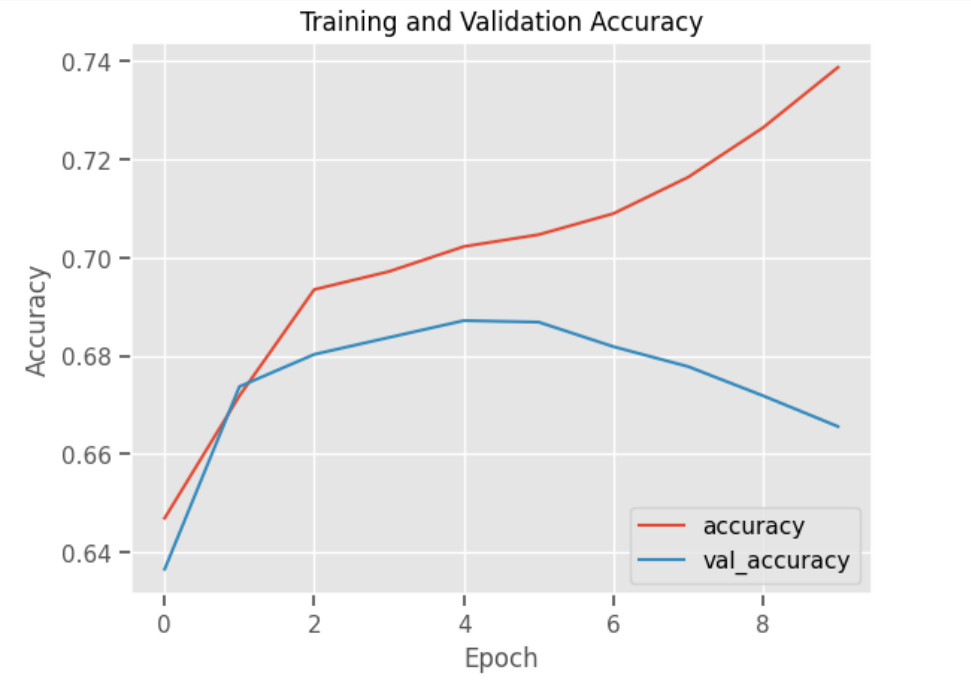
1. Entrada (n_registros,25)
2. Tres capas ocultas (128 neuronas)
3. Una capa de salida – Función de activación sigmoide
4. Optimizador : Adam – lr : 1e-4
5. Loss : binary cross entropy
6. 10 Epocas , batch-size : 16

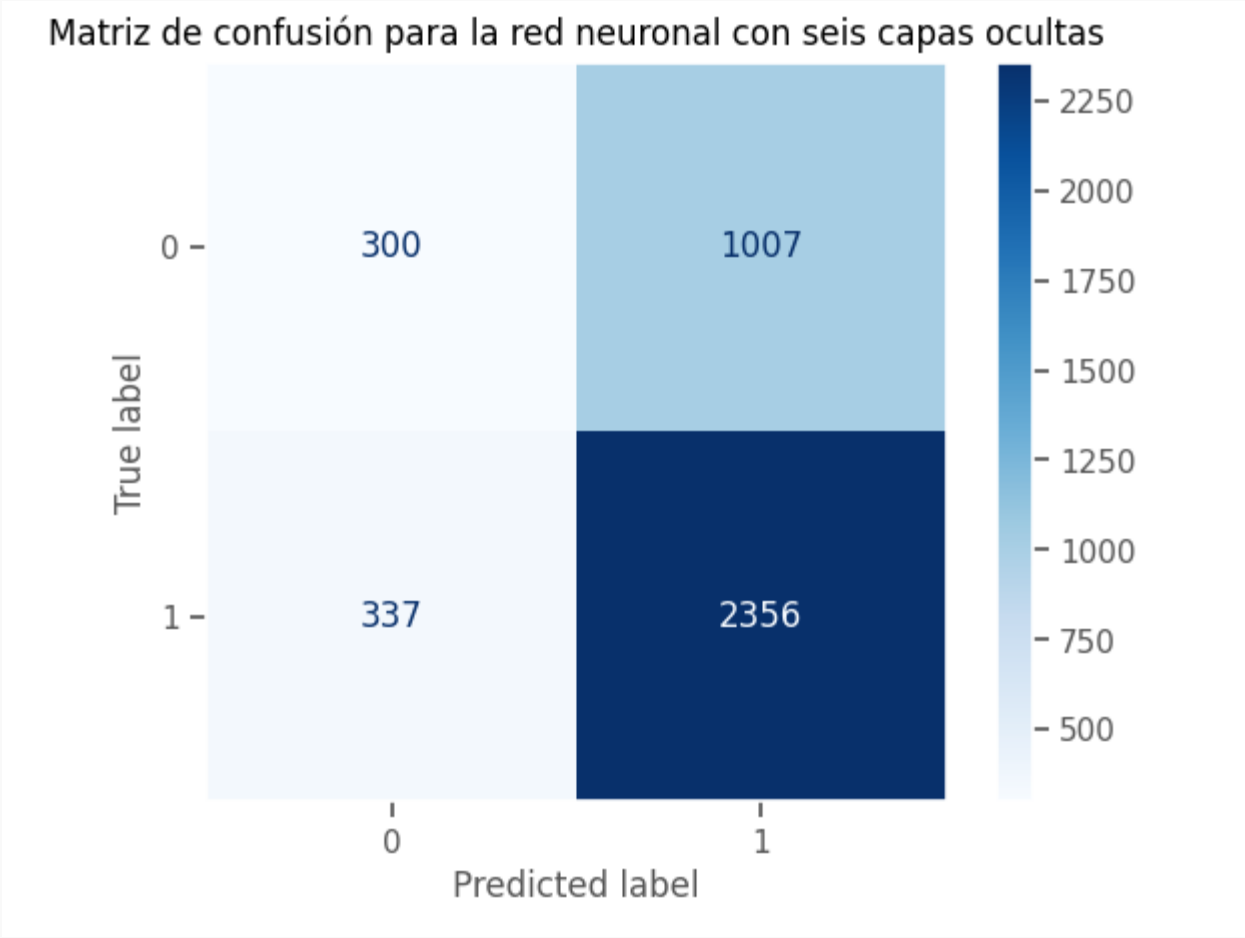



Matriz de confusión para la red neuronal con tres capas ocultas
297
1010
214
2479
0
1
True label
0
1
Predicted label
2000
1500
1000
500

La segunda configuración fue :

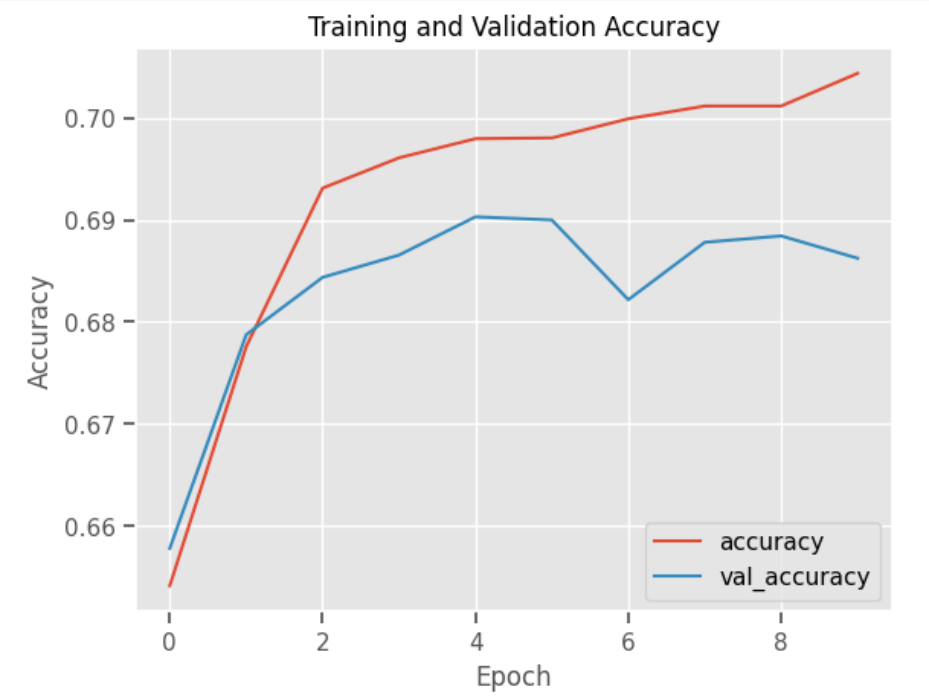
1. Entrada (n_registros,25)
2. Seis capas ocultas (128 neuronas)
3. Una capa de salida – Función de activación sigmoide
4. Optimizador : Adam – lr : 1e-4
5. Loss : binary cross entropy
6. 10 Epocas , batch-size : 32

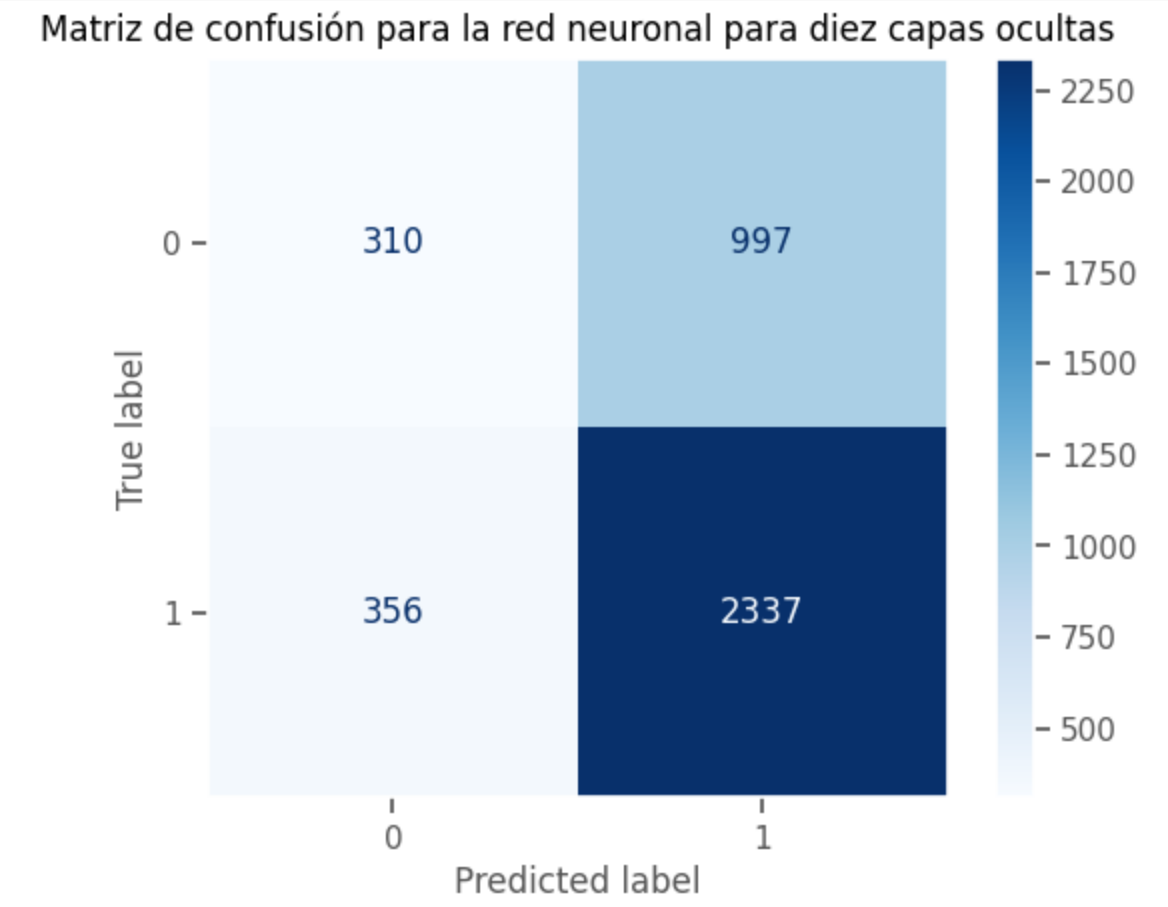



Matriz de confusión para la red neuronal con seis capas ocultas
300
1007
337
2356
0
1
True label
0
1
Predicted label
2250
2000
1750
1500
1250
1000
750
500

La segunda configuración fue :

1. Entrada (n_registros,25)
2. Diez capas ocultas (128 neuronas)
3. Una capa de salida – Función de activación sigmoide
4. Optimizador : Adam – lr : 1e-4
5. Loss : binary cross entropy
6. 10 Epocas , batch-size : 32

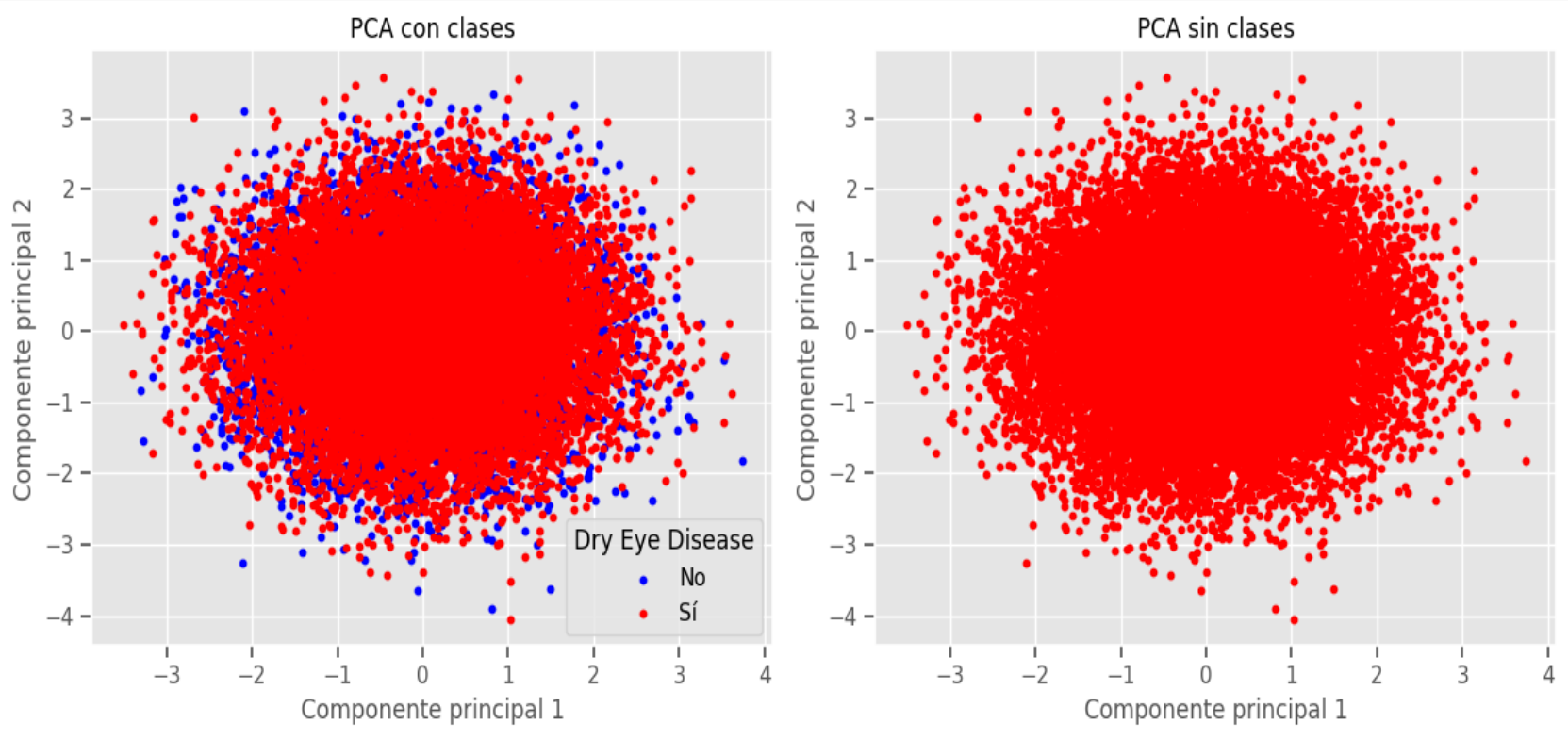



Matriz de confusión para la red neuronal para diez capas ocultas
310
997
356
2337
0
1
True label
0
1
Predicted label
2250
2000
1750
1500
1250
1000
750
500

06

Aprendizaje no supervisado

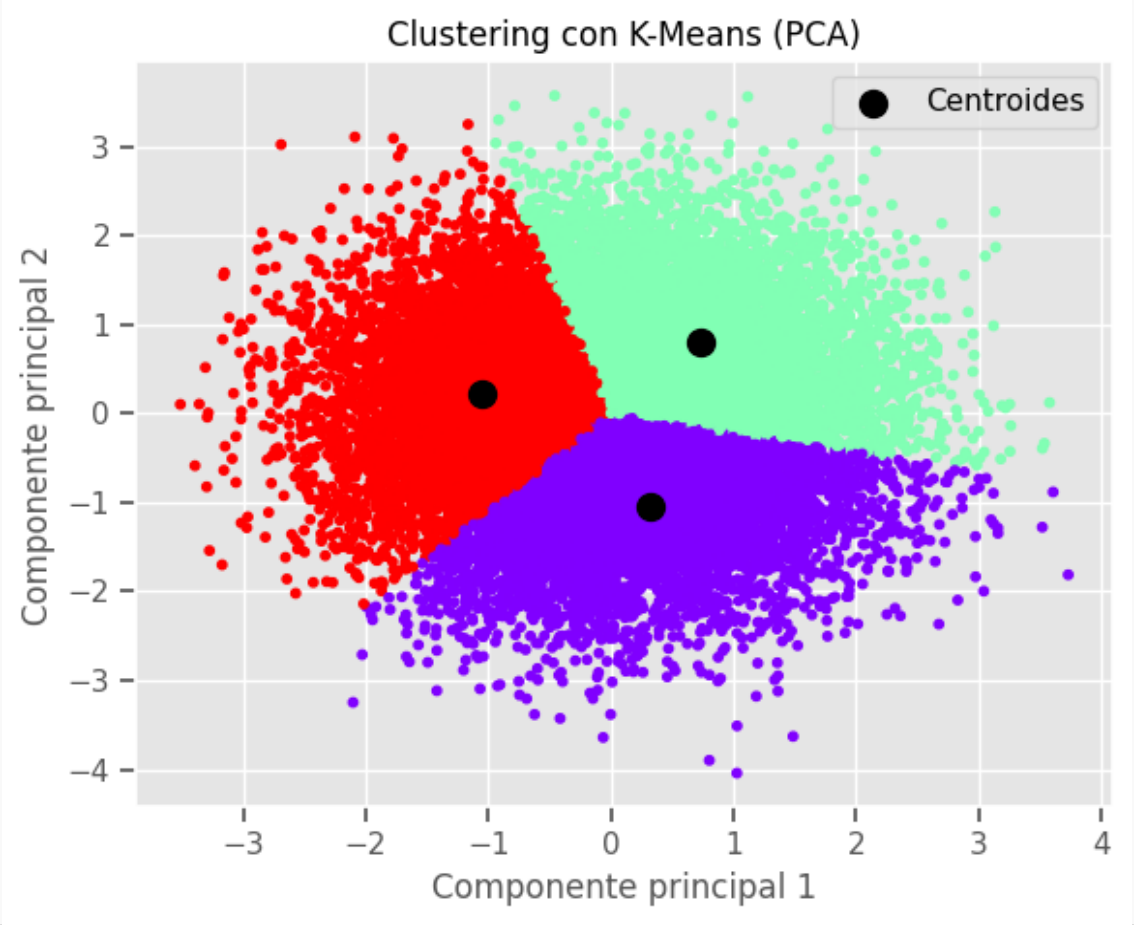
PCA


PCA con clases
Componente principal 2
Componente principal 1
Dry Eye Disease
No
Sí
PCA sin clases
Componente principal 2
Componente principal 1

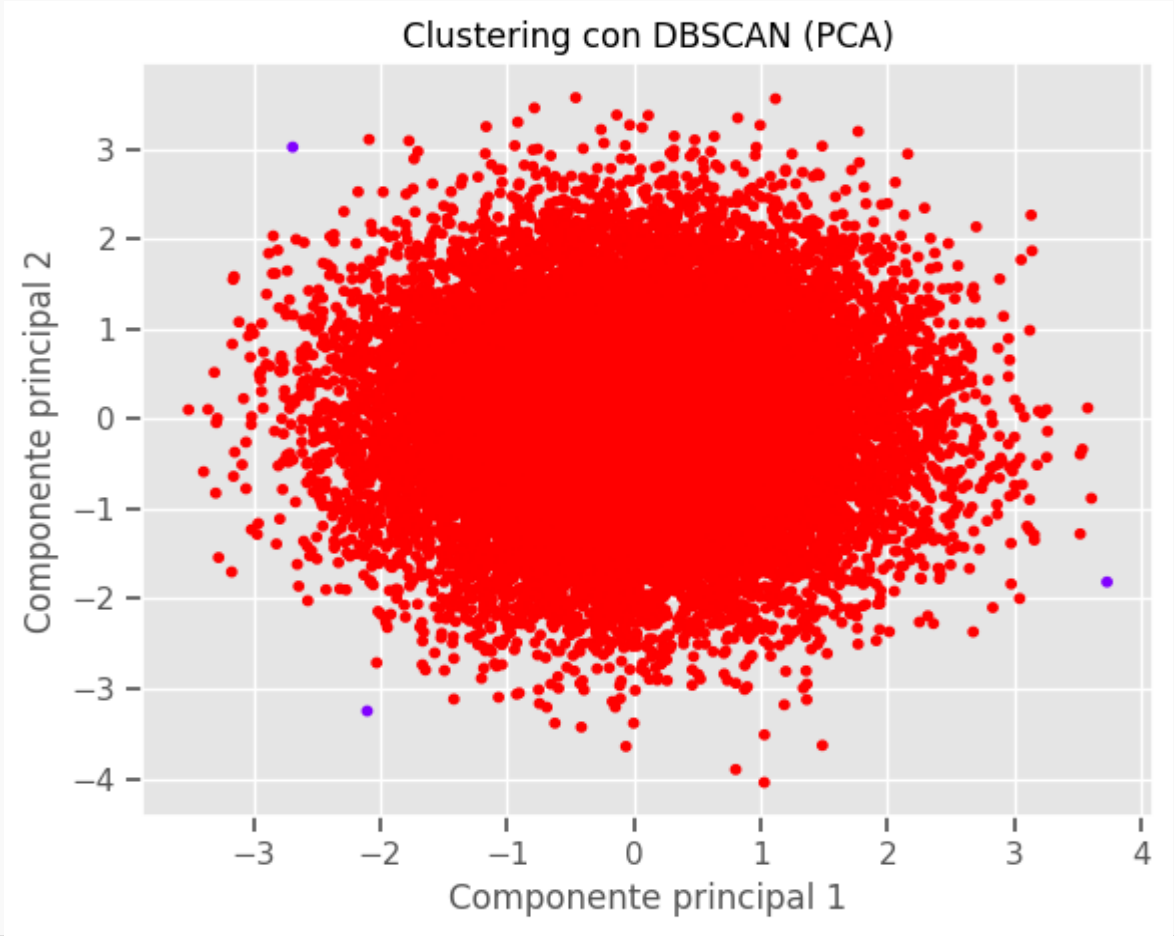
- Varianza explicada por componente:
- Componente 1: 0.0426 (4.26%)
- Componente 2: 0.0425 (4.25%)

- Varianza total explicada por los 2 componentes: 0.0851 (8.51%)

K-means


Clustering con K-Means (PCA)
Centroides
Componente principal 2
Componente principal 1
3
2
1
0
−1
−2
−3
−4
−3
−2
−1
0
1
2
3
4

DBSCAN


Clustering con DBSCAN (PCA)
Componente principal 2
Componente principal 1
3
2
1
0
−1
−2
−3
−4
−3
−2
−1
0
1
2
3
4

Conclusiones

- Los modelos de aprendizaje supervisado aplicados permiten identificar patrones relevantes entre variables relacionadas con la salud y la presencia del síndrome del ojo seco, demostrando que el enfoque basado en datos es viable para apoyar el diagnóstico preliminar.

- El modelo de SVM obtuvo el mejor rendimiento (70.1% de precisión), lo que sugiere que este algoritmo logra separar de forma más efectiva los casos positivos y negativos dentro del espacio de características del conjunto de datos.

- El Random Forest también mostró un rendimiento sólido (69.8%), validando la utilidad de técnicas de ensamblaje para mejorar la generalización y mitigar el sobreajuste, en comparación con modelos simples como el Árbol de Decisión (57.1%).

- El uso de preprocesamiento (label encoding y escalado) fue esencial para garantizar el correcto funcionamiento de los modelos, en especial SVM, que es sensible a las diferencias de escala entre variables.

- Un espacio de tan solo 2 componentes , proyectado a partir de los datos originales , no son suficientes para garantizar preservar una cantidad aceptable de varianza del espacio original , por lo cual , no se logra una gran distinción entre clases .

Gracias 👍