UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

# MÉTODO PARA LA DETECCIÓN DE LA DISLEXIA EN NIÑOS DE PREESCOLAR MEDIANTE APRENDIZAJE COMPUTACIONAL

TESIS:

PARA OBTENER EL TÍTULO DE

## INGENIERO EN COMPUTACIÓN

*PRESENTA:*

**JOSÉ EDUARDO CASTELLANOS NARVÁEZ**

DIRECTOR DE TESIS:

**DR. CHRISTIAN EDUARDO MILLÁN HERNÁNDEZ**

CO-DIRECTOR

**M.T.C.A. ERIK GERMÁN RAMOS PÉREZ**

HUAJUAPAN DE LEÓN, OAXACA.

MAYO DE 2024

*Dedicado a mis papás, Eduardo y Patricia,*

*por el apoyo y la confianza.*

*Dedicado a mis hermanos*

*Uziel y Yamile.*

# Agradecimientos

A mis padres, por brindarme su apoyo durante estos años, por los sacrificios que han hecho, por proporcionarme una educación y motivarme a superarme. Gracias por sus consejos, su confianza, su cuidado, su amor.

A Lupita, por su apoyo, cariño y amor incondicional, por ser la persona que me motiva a ser mejor persona y a superar los obstáculos que se presentan. Por todo esto y muchas cosas más, gracias.

A mis abuelos, quienes desde mi infancia me enseñaron muchos valores, me brindaron consejos y me desearon éxito y superación en mi vida.
A mis amigos, quienes me han brindado su amistad, consejos, ánimo y compañía durante la carrera.

Mi más sincero agradecimiento a mi director de tesis, el Dr. Christian Eduardo Millán Hernández, por motivarme a realizar este proyecto. Gracias por su tiempo, orientación y consejos. Toda mi admiración y respeto hacia usted.

A mi asesor de tesis, el Dr. Christian Eduardo Millán Hernández y codirector el M.T.C.A. Erik Germán Ramos Pérez, a mis sinodales, Dra. Verónica Rodríguez López, Dr. Eduardo Sánchez Soto, y Dr. José Aníbal Arias Aguilar, gracias por los comentarios y observaciones realizaron para mejorar este trabajo.

A la Universidad Tecnológica de la Mixteca por darme la oportunidad de realizar mis estudios.

# Índice general

**1. Introducción** **11**

1.1. Antecedentes . . . . . 11

1.2. Planteamiento del Problema . . . . . 13

1.3. Justificación . . . . . 15

1.4. Hipótesis . . . . . 17

1.5. Objetivos . . . . . 17

1.5.1. Objetivo general . . . . . 17

1.5.2. Objetivo específicos . . . . . 17

1.6. Metas . . . . . 17

1.7. Estructura de la tesis . . . . . 18

**2. Marco teórico** **19**

2.1. Aprendizaje computacional . . . . . 19

2.1.1. Tipos de aprendizaje . . . . . 19

2.2. Etapas de entrenamiento . . . . . 20

2.2.1. Preprocesamiento de datos . . . . . 21

2.2.2. Sobre muestreo(Oversampling) . . . . . 27

2.2.3. Ajuste de hiperparámetros . . . . . 28

2.2.4. Validación cruzada . . . . . 28

2.3. Algoritmos de aprendizaje computacional supervisados . . . . . 29

2.3.1. Técnicas de aprendizaje estadístico . . . . . 30
2.3.2. Máquina de Soporte de Vectores . . . . . 33
2.3.3. Técnicas basadas en lógica . . . . . 34
2.3.4. Técnicas basadas en perceptrón . . . . . 37
2.3.5. Evaluación de resultados . . . . . 38
2.4. Trabajos relacionados . . . . . 39
2.4.1. Métodos basados en instrumentos especializados . . . . . 40
2.4.2. Métodos basados en software . . . . . 41
2.4.3. Resumen . . . . . 42

**3. Método propuesto** **44**
3.1. Descripción de alto nivel del método propuesto . . . . . 44
3.2. Marco de referencia de trabajo . . . . . 45
3.3. Resumen . . . . . 49

**4. Resultados** **50**
4.1. Conjunto de datos . . . . . 51
4.1.1. Análisis descriptivo de los atributos de la base de datos MusVis . 51
4.2. Preprocesamiento . . . . . 55
4.2.1. Limpieza de los datos . . . . . 55
4.2.2. Separación de datos en entrenamiento y prueba . . . . . 55
4.3. Experimento 1: Modelo tipo I con el conjunto de datos MusVis-Entrenamiento balanceado . . . . . 56
4.3.1. MusVis-Entrenamiento balanceado . . . . . 56
4.4. Experimento 2: Modelo tipo II con el conjunto de datos MusVis-Entrenamiento balanceado y con selección de características . . . . . 58

4.4.1. Entrenamiento MusVis balanceado y selección de características mediante Árboles Extremadamente Aleatorios . . . . . 59
4.4.2. Entrenamiento MusVis balanceado y selección de características ANOVA . . . . . 64
4.5. Experimento 3: Modelo tipo III con el conjunto de datos MusVis-Entrenamiento balanceado y con ensamble de modelos . . . . . 69
4.6. Evaluación de resultados en el conjunto de datos MusVis-Prueba . . . . . 73
4.6.1. Evaluación de resultados del modelo tipo I . . . . . 73
4.6.2. Evaluación de resultados del modelo tipo II . . . . . 75
4.6.3. Evaluación de resultados del modelo tipo III . . . . . 82
4.7. Busqueda en malla . . . . . 84
4.7.1. Modelo tipo I . . . . . 85
4.7.2. Modelo tipo II . . . . . 86
4.7.3. Modelo tipo III . . . . . 87

**5. Conclusiones** **89**
5.1. Aportaciones . . . . . 92
5.2. Trabajo futuro . . . . . 92

**Anexos** **93**
5.3. Selección de características . . . . . 93
5.3.1. Separación de datos . . . . . 93
5.4. Selección de características . . . . . 94
5.4.1. selección de características con ANOVA . . . . . 94
5.4.2. Entrenamiento MusVis balanceado y selección de características Random Forest . . . . . 112

5.4.3. Resultados del experimento 2: selección de características con bosque aleatorio . . . . . 114

**Referencias** **116**

# Índice de figuras

2.1. Ciclo de un proyecto de ML . . . . . 21
2.2. Escala Min-Max . . . . . 23
2.3. Escalado de la varianza . . . . . 24
2.4. Escala L2-normalización . . . . . 25
2.5. División de datos en la validación cruzada . . . . . 29
2.6. Función logística estándar . . . . . 32
2.7. Matriz de confusión . . . . . 38

3.1. Etapas del marco de trabajo propuesto para la detección de dislexia . . . 47

4.1. Resultados de la sensibilidad de los modelos obtenidos en el Experimento 1. 74
4.2. Sensibilidad del modelo 2.1 sin escalamiento . . . . . 75
4.3. Sensibilidad del modelo 2.1 con escalamiento normalizer . . . . . 76
4.4. Sensibilidad del modelo 2.1 con escalamiento min-max . . . . . 77
4.5. Sensibilidad del modelo 2.1 con escalamiento estandar . . . . . 78
4.6. Sensibilidad del modelo 2.2 sin escalamiento . . . . . 79
4.7. Sensibilidad del modelo 2.2 con escalamiento normalizer . . . . . 80
4.8. Sensibilidad del modelo 2.2 con escalamiento min-max . . . . . 81
4.9. Sensibilidad del modelo 2.2 con escalamiento estandar . . . . . 82
4.10. Sensibilidad del modelo 3 con escalamiento L2-Normalizer. . . . . 83
4.11. Sensibilidad del modelo 3 con escalamiento MinMax. . . . . 83

4.12. Sensibilidad del modelo 3 con escalamiento estandar. . . . . 84
4.13. Resultados de la sensibilidad del los modelos obtenidos en el Experimento 1 con búsqueda en malla . . . . . 85
4.14. Resultados de la sensibilidad del los modelos obtenidos en el Experimento 2.1 con búsqueda en malla . . . . . 86
4.15. Resultados de la sensibilidad del los modelos obtenidos en el Experimento 2.2 con búsqueda en malla . . . . . 87
4.16. Resultados de la sensibilidad del los modelos obtenidos en el Experimento 3 con búsqueda en malla . . . . . 88

5.1. Separación de datos en entrenamiento y prueba . . . . . 93
5.2. Sensibilidad del modelo 2.3 . . . . . 115

# Índice de tablas


4.1. *Información de los niños participantes en MusVis* . . . . . 52
4.2. Edades de los niños en el conjunto de datos MusVis . . . . . 52
4.3. Características visuales y auditivas del conjunto de datos MusVis . . . . 53
4.4. Características demográficas de los participantes . . . . . 54
4.5. Características eliminadas durante la limpieza. . . . . 55
4.6. Separación de datos en entrenamiento y prueba . . . . . 55
4.7. Validación cruzada kfold con un k = 5 en el conjunto de datos MusVis balanceado. . . . . 57
4.8. Validación cruzada con un k = 5 en el conjunto de datos MusVis balanceado por clase. . . . . 58
4.9. Resultados de F1 en MusVis balanceada sin escalamiento. . . . . 60
4.10. Resultados de F1 en MusVis balanceada con normalización. . . . . 60
4.11. Resultados de F1 en MusVis balanceada con escalamiento Min-Max. . . . 61
4.12. Resultados de F1 en MusVis balanceada con escalamiento estándar. . . . 61
4.13. Resultados de sensibilidad por clase en MusVis sin escalamiento. . . . . 62
4.14. Resultados de sensibilidad por clase en MusVis con escalador normalizer. 62
4.15. Resultados de sensibilidad por clase en MusVis con escalador Min-Max. . 63
4.16. Resultados de sensibilidad por clase en MusVis con escalador Estandar. . 63
4.17. Resultados de F1 en MusVis balanceada sin escalamiento. . . . . 65
4.18. Resultados de F1 en MusVis balanceada con normalización. . . . . 65

4.19. Resultados de F1 en MusVis balanceada con escalamiento Min-Max. . . . 66
4.20. Resultados de F1 en MusVis balanceada con escalamiento estándar. . . . 66
4.21. Resultados de sensibilidad por clase en MusVis sin escalamiento. . . . . 67
4.22. Resultados de sensibilidad por clase en MusVis con escalador normalizer. 68
4.23. Resultados de sensibilidad por clase en MusVis con escalador Min-Max. . 68
4.24. Resultados de sensibilidad por clase en MusVis con escalador Estandar. . 69
4.25. Resultados de ensambles con los datos balanceados, Validación cruzada (5 folders), escalamiento Normalizer y selección de características Anova. . . 70
4.26. Resultados de ensambles con los datos balanceados, Validación cruzada (5 folders), escalamiento MinMax y selección de características Anova. . . . 71
4.27. Resultados de ensambles con los datos balanceados, Validación cruzada (5 folders), escalamiento Estandar y selección de características Anova. . . . 71
4.28. Resultados de ensambles con los datos balanceados, Validación cruzada (5 folders), escalamiento Normalizer y selección de características Anova por clase. . . . . 72
4.29. Resultados de ensambles con los datos balanceados, Validación cruzada (5 folders), escalamiento Min-Max y selección de características Anova por clase. . . . . 72
4.30. Resultados de ensambles con los datos balanceados, Validación cruzada (5 folders), escalamiento Estandar y selección de características Anova por clase. . . . . 73

5.1. Porcentaje de percentiles seleccionados por Anova en cada modelo . . . . 94
5.2. Resultados de ensambles con los datos balanceados, Validación cruzada (5 folders), escalamiento y selección de características Bosque aleatorio . . . 112
5.3. Resultados con los datos balanceados, Validación cruzada (5 folders), escalamiento y selección de características Bosque aleatorio por clase. . . . 113

# Capítulo 1

# Introducción

## 1.1. Antecedentes

La International Dyslexia Association(IDA,2002), describe la dislexia como un "trastorno específico del aprendizaje de origen neurobiológico. Se caracteriza por dificultades en el reconocimiento preciso y/o fluido de las palabras y por una deficiente capacidad para deletrear y descodificar. Estas dificultades suelen ser el resultado de un déficit en el componente fonológico del lenguaje que a menudo resulta inesperado en relación con otras capacidades cognitivas y con la impartición de una enseñanza eficaz en el aula. Las consecuencias secundarias pueden incluir problemas en la comprensión lectora y una experiencia lectora reducida que puede impedir el crecimiento del vocabulario y los conocimientos previos".

Normalmente, el trastorno de dislexia se diagnóstica clínicamente después de que el niño va por primera vez a la escuela y tiene que lidiar con aprender a leer, es entonces, cuando el niño descubre su discapacidad (Hulme y Snowling, 2009). No obstante, la mayoría de los infantes con dislexia no son detectados y en consecuencia no son tratados. Por lo tanto, la dislexia puede causar varios problemas sociales y psicológicos para el individuo, la familia y la sociedad. Es necesario realizar un diagnóstico (detección) a

tiempo para aumentar el rendimiento académico en infantes, mejorando relaciones con su grupo social y familiar. De forma tradicional, la dislexia no cuenta con un método definitivo para ser detectada, sin embargo, las tasas de errores ortográficos y errores de lectura han sido la forma más común para detectar a las personas con dislexia, utilizando las populares evaluaciones de papel y lápiz en diferentes países(Cuetos et al., 2002, 2007).

En la actualidad, el aprendizaje computacional es una herramienta importante para la detección de la dislexia. En adición, destaca por mejorar los métodos convencionales de detección de la dislexia, e impulsar la exploración de nuevas perspectivas en la identificación de individuos disléxicos (Drigas,2019). Gracias a las investigaciones recientes, en los últimos años se han creado métodos para el diagnóstico y tratamiento de la dislexia utilizando aprendizaje computacional, aportando una herramienta de apoyo para la detección de la dislexia, como primer filtro para identificar a los grupos de infantes con riesgo de este trastorno. Es importante mencionar que estas herramientas son un apoyo y no sustituyen los expertos en la detección de dislexia.

En (Lakretz et al., 2015), se propone un método para la detección de dislexia con aprendizaje computacional, mediante el modelo *Latent Dirichlet Allocation* (LDA) y Naïve Bayes, los resultados obtenidos respaldaron que LDA captura mejor los patrones de errores de lectura teniendo como métrica el Área Bajo la Curva (AUC) de 80.1 % $\pm$ 0.05. Sin embargo, no mencionan si hubo un preprocesamiento de datos al realizar la experimentación.

En (Rello, 2015) se implementan Máquina de Soporte Vectorial (SVM), el cual, fue entrenado con un conjunto de datos obtenidos mediante las lecturas de cada texto, se registraron mediante seguimiento ocular, durante la experimentación utilizaron validación cruzada y la precisión lograda mediante este método fue de 80.1 %.

Por otra parte, (Cui et al., 2016) propone un enfoque para la detección de la dislexia,

el cual se basa en imágenes de resonancia magnética a infantes, con la premisa de que al analizar la materia blanca (WM) mediante máquina de vectores de soporte lineal (LSVM) se puede diagnosticar la dislexia. El método propuesto llevó a cabo una serie de preprocesamientos mediante software médico para el análisis de neuroimágenes y logro en precisión de 83.61 % durante la etapa de entrenamiento.

Posteriormente, (Rauschenberger, 2022) realiza el diagnóstico en niños mediante selección de características utilizando el algoritmo Extra Trees y Random Forests como un modelo de clasificación, construyeron un conjunto de datos mediante un juego web, donde se median tiempos de respuesta al resolver diferentes ejercicios fonológicos y visuales. Está compuesto por 313 datos, de los cuales 116 pertenecían a niños diagnosticados con dislexia. Además, la información capturada pertenecía a dos idiomas alemán y español. En los resultados se menciona que el método tiene una precisión de 74 % para alemán y 69 % para español; un F1 75 % para alemán y 75 % para español.

En las investigaciones mencionadas anteriormente, se puede notar la aplicación de diferentes métodos para la detección de la dislexia basados en aprendizaje computacional (ML, por sus siglas en inglés de *Machine Learning*). Independiente de la variedad de información para la construcción de un conjunto de datos, se implementan distintos métodos de aprendizaje computacional. Los cuales aún no muestran una exploración completa, por esta razón, en esta investigación se aborda el estudio de un método que permita mejorar la detección de dislexia en niños de preescolar, considerando la implementación y ajuste de los hiperparámetros de distintos algoritmos de aprendizaje computacional.

## 1.2. Planteamiento del Problema

Los perfiles de personas con dislexia son muy variadas, existen una serie de manifestaciones específicas que pueden llevar a una detección de dislexia. Las pruebas de detección

se basan en la observación de errores que un niño comete en sus actividades escolares. Estas pruebas se hacen mediante la observación del habla, el pensamiento, el deletreo de las palabras y la escritura, es decir, las observaciones tienen en cuenta las áreas de lenguaje oral, las capacidades cognitivas, conductuales, las dificultades del aprendizaje, la lectura y escritura para poder evaluar a los niños adecuadamente (Pérez,2014).

Una evaluación de diagnóstico de dislexia es la única manera de identificar formalmente la dislexia. Las evaluaciones de diagnóstico de dislexia son realizadas por profesores o psicólogos especialistas con experiencia. Generalmente, no hay financiación directa para las evaluaciones de diagnóstico en las escuelas, universidades o en cualquier otro lugar; normalmente, las evaluaciones deben ser pagadas por la persona individual, los padres o empleadores (BDA,2023).

En la actualidad el aprendizaje computacional ha logrado acercar la detección de dislexia utilizando métodos de clasificación, sin embargo, desde el punto de vista computacional aún no se tiene un método que asegure detectar la dislexia completamente. Además, recolectar información de niños de preescolar implica utilizar herramientas que no siempre son accesibles en las escuelas como lo son: estudios mediante resonancia magnética o dispositivos *"eye trackers"* los cuales se utilizan para seguir el movimiento ocular en la lectura de un texto. Además, que algunos métodos para recolectar pueden ser incómodos al tomar la información de los niños de preescolar.

Dado estos inconvenientes, el recolectar información puede generar efectos negativos en los niños al no tratarlos de manera adecuada, es por esto que, optar por una manera más natural es una medida necesaria para un niño de preescolar y plantearlo ante ellos mediante un videojuego, el cual lo incentivara a participar. De esta manera, recolectar información es más fiable y evitará efectos negativos en los infantes.

Tomando en cuenta lo planeado anteriormente, en este trabajo se utiliza el enfoque de (Rauschenberger, 2022), debido a la facilidad que representa la lectura de datos, sin

embargo, en los resultados obtenidos el método tiene una precisión menor a un 70 %. Por lo que se puede considerar la posibilidad de mejorar el desempeño con la finalidad de implementar una herramienta computacional. En esta investigación se plantea estudiar distintos algoritmos de aprendizaje computacional, los cuales bajo la selección de características y un entrenamiento óptimo pueden ser una opción para este enfoque y asegure la confiabilidad en el diagnóstico de dislexia. Agregado a esto, el preprocesamiento del conjunto de datos tiene una gran relevancia en las predicciones de estas herramientas de clasificación de la dislexia, debido a que la información recolectada puede tener datos, los cuales pueden ser irrelevantes en el aprendizaje de un algoritmo computacional, por lo cual, estaría complicando la clasificación de la dislexia.

Finalmente, se plantea la pregunta a resolver: ¿es posible mejorar la detección de dislexia mediante una selección óptima de características y algoritmos de aprendizaje computacional?, utilizando el enfoque planteado en (Rauschenberger,2022), el cual obtiene el conjunto de datos mediante un videojuego web.

## 1.3. Justificación

A nivel mundial, la dislexia afecta a 1 de cada 10 personas (IDA,2016). Es decir, se estima que alrededor de 700 millones de niños y adultos tienen dislexia. Esto representa que en la mayoría de los casos no son detectados o reciben pocos o ningún servicio. Para algunos individuos que nunca han sido diagnosticados, la dislexia es una discapacidad oculta que puede dar lugar a subempleo, dificultades para desenvolverse en entornos académicos, dificultades en el trabajo y una menor confianza en sí mismos. Incluso las personas diagnosticadas pueden tener dificultades con la lectura o la escritura en algunos aspectos de su vida. Los adultos jóvenes tienen efectos a largo plazo que incluyen el fracaso escolar, la depresión y un mayor riesgo de suicidio, delincuencia y reincidencia. En México, los datos estadísticos son limitados, sin embargo, se calcula que siete por

ciento de los niños presentan alteraciones del lenguaje que pueden llegarse a considerar como dislexia (Galán, 2018).

Actualmente, existen cuatro etapas definidas en la detección de la dislexia en función de edades de desarrollo. La primera es la detección en segundo año de educación preescolar; la segunda es la que se centra en los alumnos que cursan primero y segundo año de primaria; la tercera es la detección de alumnos en el tercer año de educación primaria y la cuarta es la de quienes cursan educación secundaria, bachillerato o estudios universitarios (Pérez, 2014). Esta investigación beneficia a la creación de herramientas de apoyo para la detección de niños que se encuentran en la primera etapa de dislexia. La dislexia afecta a niños, adultos, padres, familia, profesores y representa un daño, ya que afecta directamente en el interés del niño por el estudio, el cual se ve disminuido, en especial cuando el entorno es poco estimulante. Repercute, sobre las calificaciones, las cuales también se ven disminuidas. La tarea de diagnosticar la dislexia en niños tiene una gran relevancia debido a que al identificar a los grupos en riesgo, las escuelas y padres podrán comenzar a llevar acciones para el tratamiento de este trastorno.

Por otra parte, este trabajo aporta conocimientos sobre la implementación de distintos algoritmos para la identificación de la dislexia. Adicionalmente, permite conocer la validez del conjunto de datos propuesto por Rauschenberger, el cual fue recolectado mediante una aplicación web; así como el desempeño de distintos algoritmos de aprendizaje computacional en comparación a los explorados en su trabajo. En adición, generar un nuevo enfoque para la detección de la dislexia mediante el uso de métricas adecuadas al problema. Y demostrar si los algoritmos de aprendizaje computacional que hasta ahora han sido utilizados pueden mejorarse.

## 1.4. Hipótesis

"Utilizando técnicas de selección de características a través del conjunto de datos obtenido en juego web y el ajuste de los hiperparámetros en algoritmos de aprendizaje computacional, es posible mejorar la detección de dislexia en niños de preescolar."

## 1.5. Objetivos

A continuación se presentan los objetivos generales y específicos.

### 1.5.1. Objetivo general

Mejorar la detección de dislexia en niños de preescolar mediante un conjunto de datos obtenido mediante un juego web y algoritmos de aprendizaje computacional

### 1.5.2. Objetivo específicos

- Revisar el estado del arte sobre métodos de detección de dislexia mediante aprendizaje computacional.
- Implementar un método para el entrenamiento de los algoritmos de aprendizaje computacional.
- Evaluar los modelos obtenidos de distintos algoritmos de aprendizaje computacional entrenados con el método propuesto.
- Comparar los resultados obtenidos en la detección de dislexia en niños de preescolar.

## 1.6. Metas

1. Elaboración de un reporte sobre los métodos de procesamiento de conjunto de datos.

2. Elaboración de un reporte sobre los métodos de aprendizaje computacional.

3. Implementación de un método de algoritmos de aprendizaje computacional para la detección de dislexia en niños de preescolar.

4. Elaboración de un reporte comparativo de desempeño del método propuesto contra otros enfoques.

5. Elaboración del documento de tesis.

## 1.7. Estructura de la tesis

En el Capítulo 2 se muestra información sobre el aprendizaje computacional, así como las etapas para la elaboración de un proyecto de aprendizaje computacional. Posteriormente, en el Capítulo 3 se presenta el método propuesto para mejorar la detección de la dislexia utilizando un marco de trabajo para la generación de modelos para la detección de dislexia a partir de algoritmos de aprendizaje computacional. En los siguientes capítulos se muestran los resultados y las conclusiones de este trabajo de tesis.

# Capítulo 2

# Marco teórico

## 2.1. Aprendizaje computacional

El Aprendizaje Computacional es un subcampo de la inteligencia artificial, se define como un proceso que utiliza una colección de datos para extraer patrones y construir modelos utilizados con el análisis predictivo.

### 2.1.1. Tipos de aprendizaje

Existen cuatro tipos de aprendizaje computacional, los cuales son: aprendizaje supervisado, semisupervisado, no supervisado y por refuerzo (Burkov, 2019).

Las técnicas de aprendizaje computacional supervisado aprende mediante un modelo de la relación entre un conjunto de características descriptivas y una clase, con esto se obtiene un algoritmo capaz de hacer predicciones para nuevas instancias (Burkov, 2019). El **aprendizaje supervisado** requiere de un conjunto de datos etiquetados $(x_i, y_i)^N{}_{i=1}$. Donde cada elemento $x_i$ de $N$ se denomina un vector de características, el cual tiene una dimensión de $j = 1, ...D$ el cual describe la colección de ejemplos del fenómeno.

En cambio, en el **aprendizaje no supervisado**, el conjunto de datos es una colección de ejemplos sin etiquetar $(x_i)^N{}_{i=1}$. Donde $x$ es un vector de características, con el propósito de crear un modelo que tome el vector de características $x$ como entrada y lo transforme en otro vector o un valor que pueda utilizarse para resolver un problema práctico (Burkov 2019).

Por otra parte, **aprendizaje semisupervisado**, utiliza un conjunto de datos, los cuales contienen una colección de ejemplos etiquetados $(x_i, y_i)^N{}_{i=1}$ y no etiquetados $(x_i)^N{}_{i=1}$, donde el conjunto de datos contiene una mayor cantidad de datos no etiquetados. El objetivo del aprendizaje semisupervisado es aprovechar el número de ejemplos sin etiquetar para mejorar el rendimiento del modelo sin información adicional (Burkov, 2019).

Por último, el **aprendizaje por refuerzo**, está inspirada en la psicología conductista, en este tipo de aprendizaje la máquina "vive" en un entorno y tiene la capacidad de percibir el estado de ese entorno. El objetivo de un algoritmo de aprendizaje por refuerzo es aprender una política. Una política es una función (similar al modelo en el aprendizaje supervisado) que toma el vector de características de un estado como entrada y produce una acción óptima para ejecutar en ese estado. El aprendizaje por refuerzo resuelve un tipo concreto de problema en el que en los que la toma de decisiones es secuencial y objetivo a largo plazo. Un ejemplo de este tipo de aprendizaje son los juegos, la robótica, la gestión de recursos o la logística(Burkov, 2019).

## 2.2. Etapas de entrenamiento

Durante el desarrollo de un proyecto de ML se realiza un proceso dividido en distintas fases para obtener un modelo como resultado. En este proceso se compone de una serie de actividades repetitivas como lo son: recolección de datos, preprocesamiento de datos, selección de modelos, entrenamiento y evaluación del modelo. En la Figura 2.1 se representa una visión general de este ciclo de diseño de un modelo de ML.

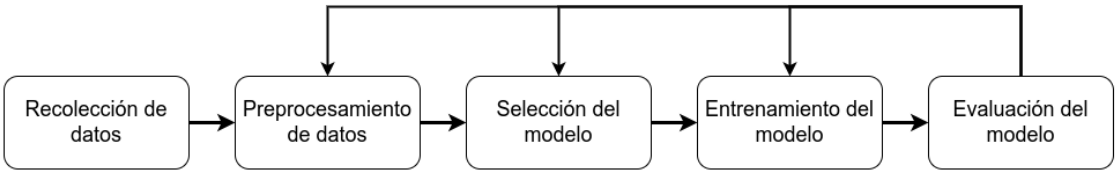


Figura 2.1: Ciclo de un proyecto de ML

Antes de realizar la etapa de entrenamiento es necesario recolectar y etiquetar los datos, para algoritmos de aprendizaje supervisado. En muchas ocasiones, representa un alto costo económico y de tiempo, debido al tiempo de recolección o la necesidad de equipo especializado. Posteriormente, el conjunto de datos obtenido es preprocesado. Para eliminar información irrelevante y preparar los datos para ser utilizados durante el entrenamiento. La selección del modelo, depende de la cantidad de datos disponibles. Además del tipo de predicciones que se realizará (regresión, clasificación), es necesario considerar que no existe un algoritmo que funcione para todos los problemas, es decir, depende de las necesidades que se tengan que cubrir y la cantidad de datos disponibles. Después de la selección del modelo, se realiza el entrenamiento con el conjunto de datos preprocesado y las características seleccionadas, así como un ajuste de parámetros del modelo y por último los resultados son evaluados respecto a métricas de interés como precisión y exhaustividad.

### 2.2.1. Preprocesamiento de datos

El éxito de un proyecto de ML depende de muchos factores. El factor mas importante es la calidad de los datos. Esto es crucial en el proyecto, debido a que la representación de la información influirá en el funcionamiento del algoritmo, si el conjunto de datos contiene información irrelevante, redundante, o datos atípicos o poco fiables; entonces los resultados en la fase de entrenamiento se verán afectados con un "*overfitting*" o "*underfitting*" (en español se puede traducir como sobre ajuste e infra juste, respectivamente).

Es por esto que, la información debe tener una fase de preprocesamiento de los datos, en la cual se incluye un proceso de limpieza de los datos, normalización, transformación y selección de características (Kotsiantis et al,2006). A continuación se presentan algunos algoritmos para el preprocesamiento de los datos.

### Limpieza de datos

El objetivo de la limpieza del conjunto de datos, es eliminar valores redundantes, nulos o atípicos, los datos atípicos se definen como: "un punto muy alejado de la media de la variable aleatoria correspondiente" (Theodoridis,2006), es decir, que la información representa errores de captura. Además, los datos con frecuencia contienen información con valores faltantes o nulos, esto representa un impacto negativo en el modelo, debido a que el modelo puede aprender patrones incorrectos y tener un modelo con tendencia a sesgos. Por último, eliminar información redundante o valores atípicos se puede mejorar la eficiencia del modelo.

### Escalamiento

El escalamiento es necesario cuando las características son numéricas, por ejemplo los recuentos, que pueden aumentar sin tener un límite. Algunos modelos como regresión logística o máquinas de soporte vectorial (SVM) son sensibles a los datos de entrada, y se ven afectados por la escala de los datos de entrada (Müller, 2017). En cambio, algoritmos como los árboles de decisión no son afectados por datos sin escalar (Müller, 2017). El escalamiento o normalización de características, se realiza normalmente de forma individual para cada característica (Zheng y Casari, 2018). A continuación se presentan varios tipos de operaciones de escalado comunes, cada una da como resultado una distribución de los valores de las características.

#### Escala mín-max

El escalado min-max se define de la siguiente manera: sea $x$ un valor de característica individual (es decir, un valor de la característica en algún punto de datos), y $min(x)$ , $max(x)$ los valores mínimo y máximo de esta característica en todo el conjunto de datos, respectivamente. El escalado mín-máx comprime (o estira) todos los valores de las características para que estén dentro del intervalo [0, 1] (Zheng y Casari, 2018). La fórmula para el escalado min-max es:

$$\tilde{x} = \frac{x - min(x)}{max(x) - min(x)} \tag{2.1}$$

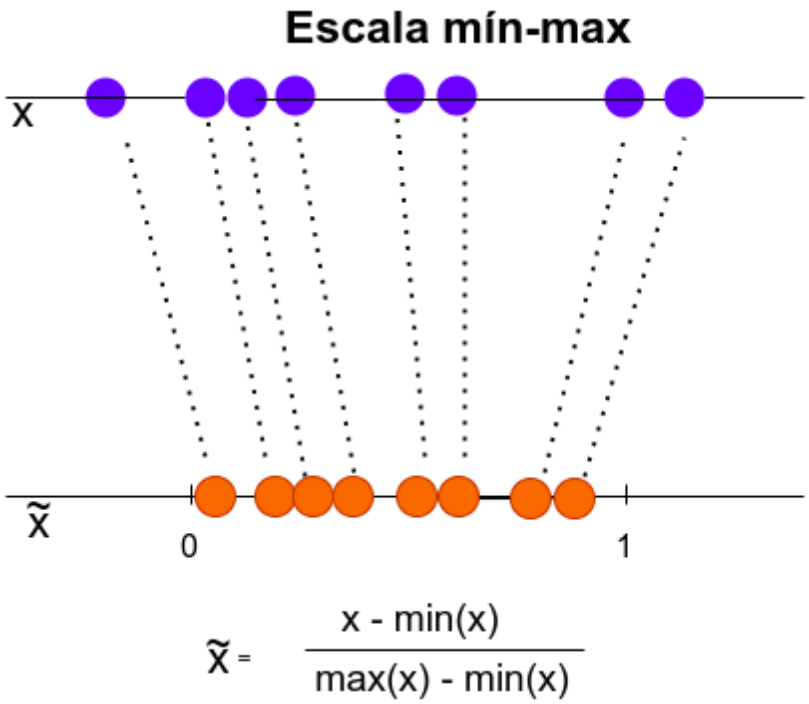


Figura 2.2: Escala Min-Max

**Estandarización (escalado de la varianza)**

La estandarización de características se define como:

$$\tilde{x} = \frac{x - \bar{x}}{\sqrt{\sigma^2}} \tag{2.2}$$

En la fórmula se observa que para obtener la normalización se resta la media sobre todos los puntos de datos y se divide por la raíz cuadrada de la varianza. Razón por la que también este tipo de escalado puede denominarse escalado de varianza. El objetivo de este tipo de escalado es tener un valor de media 0 y una varianza de 1. De esta forma que la característica original y la característica comparten una distribución gaussiana (Zheng y Casari, 2018). En la figura 2.3 se muestra la estandarización o escalado de varianza.

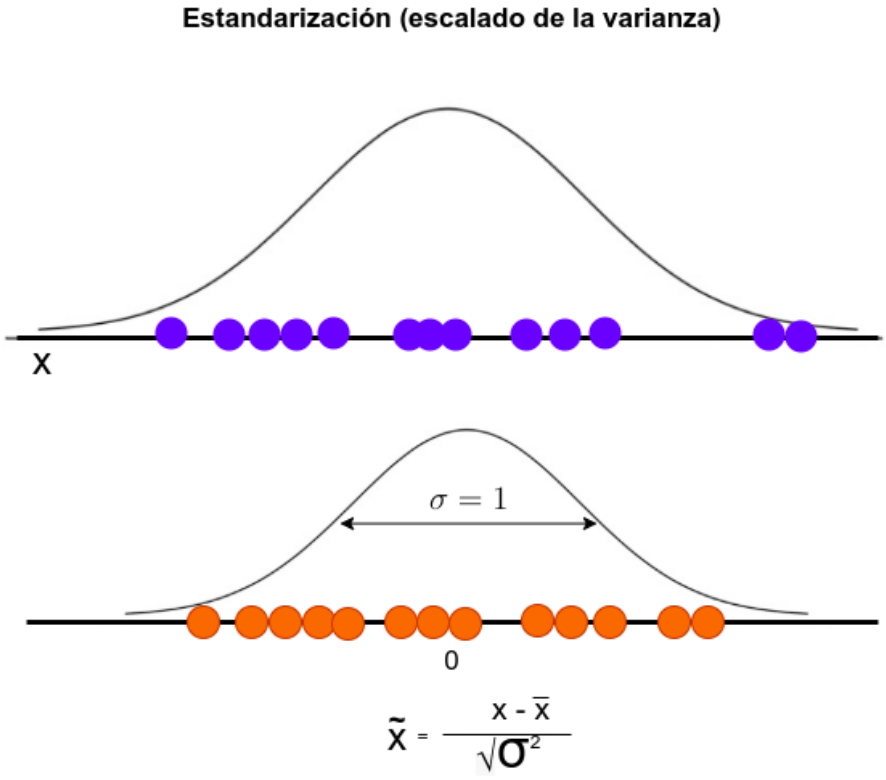


Figura 2.3: Escalado de la varianza

**Escala L2-normalización**

En esta técnica de escalamiento se divide el valor original de la característica por la norma L2, también conocida como norma euclidiana. Donde la norma euclidiana se define como:

$$\tilde{x} = \frac{x}{||x||_2} \tag{2.3}$$

La norma L2 mide la longitud del vector en el espacio de coordenadas. La definición puede derivarse del conocido teorema de Pitágoras que nos da la longitud de la hipotenusa de un triángulo rectángulo dadas las longitudes de los lados:

$$||x||_2 = \sqrt{{x_1}^2 + {x_2}^2 + \ldots + {x_m}^2} \tag{2.4}$$

La norma L2 suma los cuadrados de los valores de las características en los puntos de datos y, a continuación, obtiene la raíz cuadrada de la suma. A la normalización L2 también se llama L2 escala. (En términos generales, escalar significa multiplicar por una constante, mientras que la normalización podría implicar una serie de operaciones)(Zheng y Casari, 2018).

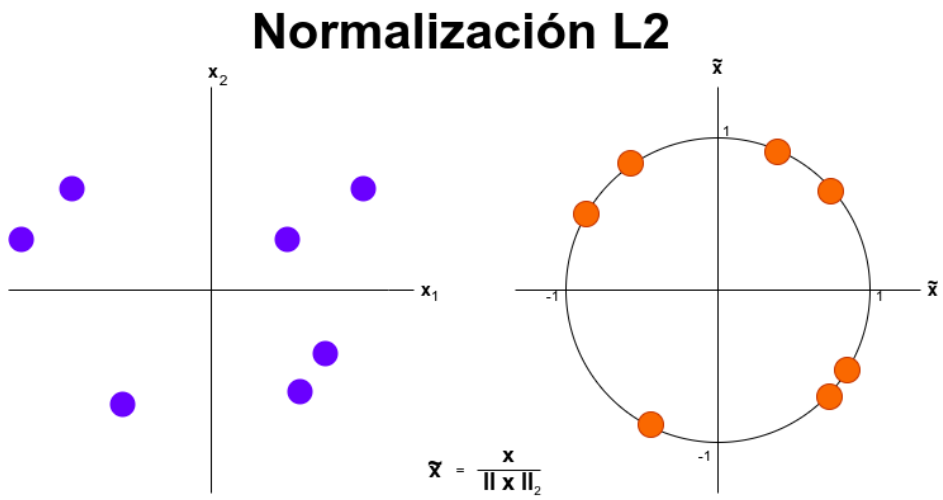


Figura 2.4: Escala L2-normalización

### Selección de características

La selección de características es un proceso clave en el desarrollo de un proyecto de aprendizaje computacional, ya que eliminan información que no es útil, al reducir la complejidad del modelo resultante. Con esto se tiene un beneficio al reducir el margen de error durante el aprendizaje del modelo (Zheng y Casari, 2018).

Los métodos de selección de características se dividen en tres (Zheng y Casari, 2018).

#### Métodos de filtro

Las técnicas de filtrado preprocesan las características para eliminar las que no sean útiles para el modelo. Para esto se puede realizar el cálculo de correlación o la información mutua entre cada característica y filtrar las características que estén por debajo de un umbral. Además, las técnicas de filtrado son menos costosas que los métodos de envoltura, sin embargo, los métodos de filtro no toman en cuenta el modelo que se emplea. Por lo tanto, es posible que no puedan seleccionar las características adecuadas para el modelo. En este caso se recomienda hacer un pre filtrado, para no eliminar inadvertidamente las características que sean de relevancia para el modelo (Zheng y Casari, 2018).

#### Métodos de envoltura

Los métodos de envoltura son costosos computacionalmente, sin embargo, permiten probar con subconjuntos de características, lo que significa que será una selección de características más cuidadosa y se evitará eliminar información importante para el aprendizaje del modelo. Los métodos de envoltura tratan a los modelos como una caja negra que proporciona mejores resultados de un subconjunto propuesto. (Zheng y Casari, 2018).

#### Métodos de integración

Los métodos de integración realizan la selección de características como parte del proceso de formación del modelo, es decir, el modelo selecciona las características más importantes, los modelos que realizan selección de características son los árboles de decisión y variantes de este modelo como lo son bosque aleatorio y árboles extremadamente

aleatorios. Otro ejemplo, es la regularización L1, la cual se añade al entrenamiento del modelo con el objetivo de disminuir el uso de características, a esto también se le conoce como escasez en el modelo. Este tipo de métodos no son tan potentes como los métodos de envoltura, sin embargo, su costo computacional es menor que los métodos de filtro. (Zheng y Casari, 2018).

### 2.2.2. Sobre muestreo(Oversampling)

Normalmente en la práctica existen casos donde un conjunto de datos está desbalanceado, es decir, tiene más datos de una clase de manera significativa, esto puede generar problemas en el rendimiento del modelo y afecta a varios clasificadores como lo son: árboles de decisión, redes neuronales multicapa, SVM y clasificadores por refuerzo. El desequilibrio entre clases no puede ser un problema si el aprendizaje es fácil; sin embargo, habrá casos en los que un desequilibrio representa un problema que afecta al aprendizaje del modelo. Para manejar este tipo de problemas se puede realizar un aumento de datos mediante múltiples copias de los ejemplos de la clase(esto tiene como nombre sobre muestreo). También puede intentar crear ejemplos sintéticos muestreando aleatoriamente los valores de las características de varios ejemplos de la clase minoritaria y combinándolos para obtener un nuevo ejemplo de esa clase. Existen dos algoritmos populares que sobre muestrean la clase minoritaria creando ejemplos sintéticos: la técnica de sobre muestreo sintético de minorías (SMOTE) y el método de muestreo sintético adaptativo (ADASYN). La técnica de sobre muestreo de SMOTE consiste en que cada muestra de la clase minoritaria, selecciona al azar uno o varios de sus vecinos más cercanos en el espacio de características. Estos vecinos se seleccionan utilizando medidas de similitud como la distancia euclidiana. Posteriormente, genera nuevas muestras sintéticas interpolando características entre la muestra original y sus vecinos seleccionados. Esto se hace multiplicando la diferencia entre las características de la muestra original y sus vecinos por un número

aleatorio entre 0 y 1, y sumándolo a las características de la muestra original (Burkov, 2019).

### 2.2.3. Ajuste de hiperparámetros

Los algoritmos de aprendizaje computacional cuentan con varios ajustes que podemos utilizar para controlar el comportamiento del algoritmo. Estos ajustes son llamados hiperparámetros o parámetros, los cuales están diseñados para mejorar el rendimiento de los algoritmos. En ocasiones se controlan hiperparámetros que no influyen en el aprendizaje del modelo porque son difíciles de optimizar y en consecuencia, el algoritmo no aprende durante el entrenamiento, esto se debe a que el hiperparámetro ajustado no es apropiado, eligiendo la máxima capacidad posible del modelo, lo que da lugar a un sobre ajuste(Goodfellow et al., 2016).

#### Búsqueda en malla (Grid Search)

Una tarea importante es encontrar los valores de los parámetros del algoritmo seleccionado que proporcionaran un resultado óptimo para mejorar el rendimiento del algoritmo de aprendizaje computacional. Para este fin se utiliza la búsqueda en malla, la cual es una técnica de ajuste de algoritmos, que prueba todas las combinaciones posibles de los parámetros que interesa ajustar en el modelo. Es un tipo de búsqueda exhaustiva, en la que se evalúan y comparan todas las combinaciones posibles para obtener un modelo de mejor rendimiento (Müller, 2017).

### 2.2.4. Validación cruzada

La validación cruzada es un método estadístico para evaluar el rendimiento de la generalización del modelo entrenado. Este método consiste en separar los datos de entrada en subconjuntos y evaluar el modelo entrenado con otro subconjunto complementario de

los datos. Con este tipo de validación se obtiene un modelo más estable y exhaustivo, en comparación cuando los datos son separados de forma simple (entrenamiento y prueba), ya que puede estar en riesgo de sobre ajuste (Müller, 2017).

En la validación cruzada se dividen los datos en varios subconjuntos, llamados "folds" y se entrena el modelo con una parte de los datos y se realiza la prueba con la otra parte que fue excluida del subconjunto de datos durante el entrenamiento. Este proceso se repite en varias ocasiones dependiendo del número de divisiones o "folds" realizados. En la figura 2.5 se muestra un ejemplo dónde se definen 5 divisiones.

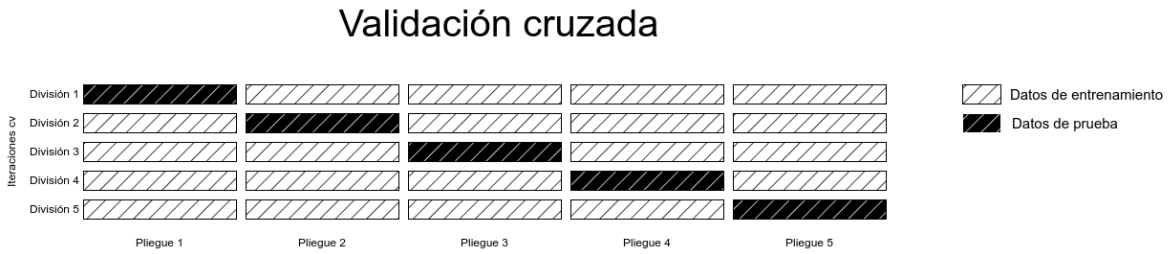


Figura 2.5: División de datos en la validación cruzada

## 2.3. Algoritmos de aprendizaje computacional supervisados

A continuación se describen algunos algoritmos de aprendizaje computacional supervisados, se explica la forma en la que aprenden y la manera en que realizan predicciones. Además, se muestra la complejidad de cada algoritmo. Por último, se resaltan los puntos fuertes y desventajas de cada modelo de aprendizaje supervisado.

### 2.3.1. Técnicas de aprendizaje estadístico

#### Clasificador bayesiano ingenuo (Naïve Bayes)

Los algoritmos basados en Naive Bayes son algoritmos de clasificación muy rápidos y sencillos, suelen ser adecuados para conjuntos de datos con una gran cantidad de características, la razón de su eficiencia es la manera de aprendizaje donde se observa cada característica de manera individual y se obtienen estadísticas simples por cada clase en cada característica. Este algoritmo se basa en el teorema de Bayes, el cual describe la relación de probabilidades condicionales de cantidades estadísticas, y el algoritmo se enfoca en encontrar la probabilidad de una etiqueta dadas las características observadas, esto lo podemos formular $P(L|caracteristicas)$. El teorema de Bayes expresa esto en términos de cantidades que se puede calcular de manera más rápida.

$$P(L|\text{características}) = \frac{P(L|\text{características})P(L)}{P(\text{características})} \tag{2.5}$$

Para la clasificación binaria, se toman decisiones para separar la información en sus respectivas clases mediante el cálculo del cociente de las probabilidades posteriores de cada clase.

$$\frac{P(L1|\text{características})}{P(L1|\text{características})} = \frac{P(L1|\text{características})P(L1)}{P(L2|\text{características})P(L2)} \tag{2.6}$$

En donde L1 y L2 representan las clases y el modelo nos permite calcular cada etiqueta o clase, este tipo de modelo se le denomina modelo generativo porque especifica el proceso en que se generan los datos. En esta parte del algoritmo se comienzan suposiciones "ingenuas" sobre el modelo generativo de cada etiqueta. Actualmente, existen distintos tipos de clasificadores de Bayes ingenuos, los cuales se basan en suposiciones ingenuas sobre los datos. Por ejemplo, el clasificador de Bayes ingenuo gaussiano supone que los datos están distribuidos en una distribución gaussiana simple, bajo esta suposición se puede realizar un entrenamiento calculando la media y la distribución estándar de los

datos etiquetados para realizar predicciones (Vanderplas, 2017). Los clasificadores bayes ingenuos hacen suposiciones estrictas sobre los datos, normalmente su funcionamiento con modelos complejos no tienen un buen funcionamiento, sin embargo, tienen ventajas a su favor:

- Son rápidos durante el entrenamiento y prueba.
- Son fáciles de interpretar en la mayoría de veces.
- Tienen pocos parámetros para ajustar.

Estas ventajas lo hacen ideal para una clasificación inicial de referencia y entender la complejidad que tienen los datos, además, funcionan bien en casos como: las clases tienen una separación clara y cuando los datos son de alta dimensionalidad. Por esta razón, los clasificadores simples, como Bayes ingenuo, tienden a funcionar igual o mejor que los clasificadores complejos a medida que aumenta la dimensionalidad de los datos (Vanderplas, 2017).

**Regresión Logística**

El modelo de regresión logística es un algoritmo de aprendizaje que se enfoca en la clasificación; sin embargo, recibe el nombre de regresión logística debido a que tiene una formulación matemática similar a la regresión linear y es capaz de realizar tanto clasificación binaria como clasificación múltiple (Burkov, 2019).

En la regresión logística, se trata de modelar $y_i$ como una función lineal de $x_i$, sin embargo, para la función binaria $y_i$ es un poco complicado de modelar. La combinación lineal de características como $wx_i+b$ es una función que va de $(-\infty, \infty)$ y $b$ es una función que va de $(-\infty, \infty)$, mientras que $y_i$ solo tiene dos valores posibles (0,1), para esto se define una etiqueta negativa (0) y una etiqueta positiva (1), con esto se busca encontrar una función continua simple cuyo codominio se encuentre en (0,1), es decir, si el valor

devuelto por el modelo para la entrada $x$ es muy próximo a 0, entonces le corresponde a la etiqueta negativa; en caso contrario le corresponde la etiqueta positiva (Burkov, 2019).

La **función logística estándar** (también conocida como **función sigmoide**) es capaz de clasificar con esta propiedad:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.7}$$

donde $e$ es la base del logaritmo natural (también conocido como *número de Euler*.

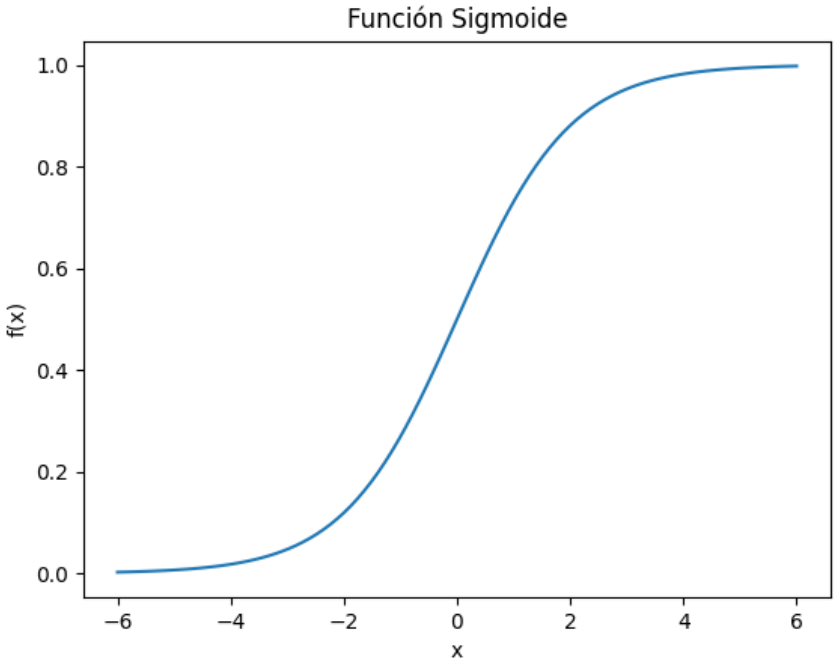


Figura 2.6: Función logística estándar

Como se puede observar en la figura 2.6, la función sigmoide se ajusta para problemas de clasificación binaria; ahora se puede optimizar los valores de $x$ y $b$, con el objetivo de interpretar la salida de $f(x)$ como la probabilidad de que $y_i$ sea positiva. Por lo tanto, podemos ver la función sigmoide de la siguiente manera:

$$f_{w,b}(x) = \frac{1}{1 + e^{-(wx+b)}} \tag{2.8}$$

El modelo de regresión logística es fácil de implementar y se utiliza para clasificación binaria y clasificación múltiple, proporciona resultados probabilísticos, lo que hace que se puedan ajustar los umbrales de decisión, con esto, podemos ajustar para conjuntos de datos desbalanceados.

### 2.3.2. Máquina de Soporte de Vectores

Las máquinas de soporte vectores (SVM, por sus siglas en inglés) permiten la regresión y la clasificación. En el caso de SVM la clasificación se denomina discriminativa debido a que en lugar de modelar cada clase, se centra en encontrar un linea o curva(en dos dimensiones) o un "manifold"(en múltiples dimensiones) que divide clases entre sí (Vanderplas,2017).

Los modelos de SVM son potentes y funcionan bien con una gran variedad de problemas de clasificación. Debido a que los SVM permiten límites de decisión compleja, sin importar que el conjunto de datos solo tenga pocas características, sin embargo, es necesario considerar que no se adaptan muy bien al número de muestras, es decir, afecta los tiempos de ejecución y en consecuencia entre más muestras tengan el algoritmo utilizara más tiempo de ejecución y más consumo de memoria. Agregado a esto, los modelos basados en SVM son sensibles al escalado de los datos y al ajuste de hiperparámetros (Müller,2017).

El aprendizaje del algoritmo de SVM da importancia a cada uno de los puntos de los datos durante el entrenamiento para representar el límite de decisión entre las dos clases, es por esto que se denominan vectores de soporte y dan nombre a la máquina de vectores de soporte (Müller,2017).

Para predecir un nuevo punto, se mide la distancia a cada uno de los vectores de soporte. Se toma una decisión de clasificación basada en las distancias a los vectores de soporte y en la importancia de los vectores de soporte que aprendió durante el entrenamiento.

El cálculo de la distancia entre los puntos de datos se mide mediante el kernel gaussiano:

$$k_{rbf}(x_1, x_2) = exp(\gamma||x_1 - x_2||^2) \tag{2.9}$$

Donde $x_1$ y $x_2$ son los puntos de datos, $||x_1 - x_2||$ denota la distancia euclidiana y $\gamma$ (gamma) es un parámetro que controla la anchura del kernel gaussiano.

Un valor bajo del valor gamma significa que el límite de decisión variará lentamente, lo que da lugar a un modelo de baja complejidad, mientras que un valor alto de gamma da lugar a un modelo más complejo (Vanderplas,2017).

### 2.3.3. Técnicas basadas en lógica

#### Bosque aleatorio (Random Forest)

Un bosque aleatorio se basa en los árboles de decisión. Por lo tanto, es necesario entender los árboles de decisión, los cuales son formas muy intuitivas de clasificar o etiquetar objetos, simplemente este algoritmo se plantea una serie de preguntas diseñadas para poder realizar la clasificación.

Un árbol de decisión bien construido, para la clasificación binaria, será mas eficiente al reducir a la mitad el número de opciones por cada pregunta realizada, esto aplica incluso para la clasificación múltiple. El objetivo de un árbol de decisión es decidir que preguntas realizar en cada paso. Sin embargo, los árboles de decisión pueden llegar a un sobre ajuste con una profundidad grande, es decir, el modelo se ajusta a los detalles de los datos particulares y no a las propiedades generales de los datos (Vanderplas,2017).

Este enfoque de clasificación, dio pie a una combinación de múltiples árboles de decisión para reducir el sobre ajuste, este enfoque es llamado ensamble, el cual utiliza un conjunto de clasificadores de forma paralela, con esto se toma un promedio de los resulta-

dos para encontrar una clasificación mejor. Un conjunto de árboles de decisión aleatorios se denomina bosque aleatorio o "Random forest".(Vanderplas,2017).

El modelo de bosque aleatorio es muy rápido en la etapa de entrenamiento y predicción, debido a que este algoritmo tiene una simplicidad en los árboles de decisión. Además, ambas tareas pueden paralizarse, porque los árboles de decisión son entidades independientes. Por último, permiten una clasificación probabilística y no son sensibles a un escalado de datos, lo cual los hace un algoritmo muy utilizado por la comunidad científica. Sin embargo, una desventaja de los bosques aleatorios es que los resultados no son fácilmente interpretables. (Vanderplas,2017).

### Árboles Extremadamente Aleatorios (Extra Trees)

El modelo de árboles extremadamente aleatorios es un tipo de modelo considerado de ensamble, el cual se construye mediante árboles de decisión, su diferencia respecto a otros métodos basados en ensambles, es que este algoritmo divide los nodos eligiendo los puntos de corte de forma totalmente aleatoria. Además, construye los árboles de decisión mediante una combinación de selección de características y un corte aleatorio de los nodos. Con esto se construyen árboles totalmente aleatorios cuyas estructuras son independientes de los valores de la clase. el modelo de árboles extremadamente aleatorios resuelve problemas, los cuales, contienen un gran número de características (numéricas normalmente) y dos clases (numérica o categórica).

El algoritmo de árboles extremadamente aleatorios construye un conjunto de árboles de decisión o regresión, con esto se basa en la aleatorización en los puntos de corte en la poda de los nodos, así como toma en cuenta todos los resultados de los demás árboles para reducir la varianza durante el entrenamiento (Geurts et al.,2006).

En (Geurts et al.,2006) se realizó un análisis de sesgo/varianza en el algoritmo de árboles extremadamente aleatorios y se demostró que existe una relación entre la varianza

y el sesgo, con esto podemos ajustar la aleatorización a un nivel óptimo para disminuir la varianza y en consecuencia el sesgo aumenta ligeramente. En cambio, si la aleatorización incrementa, la varianza disminuye ligeramente mientras que el sesgo aumenta a menudo de forma significativa. Por último, al aumentar el sesgo influye en que el algoritmo detecte las características de baja relevancia y reduce el tamaño efectivo de la muestra cuando hay muchos atributos de este tipo.

**Refuerzo de Gradiente (Gradient Boosting)**

El método de refuerzo de gradiente se basa en un árbol de regresión de gradiente reforzado, el cual acopla árboles de decisión para construir un modelo más potente. Este método tiene aplicaciones para la regresión y la clasificación. A diferencia del enfoque de bosque aleatorio, el algoritmo de refuerzo de gradiente se enfoca en la construcción de árboles aleatorios y en la elaboración de cada árbol de decisión, trata de corregir los errores del árbol anterior. Por defecto no hay aleatorización en los árboles, sin embargo, esto lo compensan con poca profundidad en los árboles (menores a 5 árboles), además, esto hace que consuma pocos recursos en memoria y las predicciones se vuelvan más rápidas.

Este método tiene principalmente el objetivo de combinar modelos sencillos(este enfoque se conoce como aprendizajes débiles), como árboles poco profundos, donde cada árbol de decisión ofrece predicciones buenas sobre una parte de los datos, por lo que añaden en cada iteración más árboles con el propósito de mejorar el rendimiento del algoritmo iterativamente. El método de refuerzo por gradiente es utilizado mucho en las industrias, además, tienen una configuración sensible de los parámetros en comparación con los bosques aleatorios, sin embargo, pueden proporcionar un mejor rendimiento si los parámetros se ajustan correctamente. (Müller,2017).

#### Ensamble de modelos

Un enfoque de aprendizaje es el ensamble de modelos, los cuales en lugar de tener un modelo con mucha precisión, se basa en entrenar un gran número de modelos de baja precisión y después combinar las predicciones de los modelos para obtener un modelo de alta precisión. Los modelos de baja precisión suelen ser débiles, es decir, son modelos simples; sin embargo, son rápidos de entrenar y rápidos en una predicción. Normalmente, los modelos más sencillos en un ensamble de modelos son los árboles de decisión, con poca profundidad, sin embargo, al tomar la combinación de un conjunto de árboles de decisión, su precisión termina siendo muy alta (Burkov, 2019).

### 2.3.4. Técnicas basadas en perceptrón

#### Perceptrón multicapa

El modelo de perceptrón multicapa pertenece a los algoritmos de aprendizaje profundo, este tipo de algoritmos resuelve el problema central del aprendizaje de representaciones introduciendo representaciones que se expresan en términos más simples. La red profunda feedforward o perceptrón multicapa (MLP) está constituida por una función matemática que asigna un conjunto de valores de entrada a valores de salida. La función matemática está conformada por funciones más simples. Con el objetivo de pensar en cada función matemática diferente como una nueva representación de los datos de entrada. La representación correcta de los datos ofrece una perspectiva del aprendizaje profundo (Goodfellow, 2016).

El algoritmo de MLP está constituido por una serie de capas, comenzando por la capa de entrada, la cual recibe los datos, posteriormente las capas ocultas se encargan de extraer las características más abstractas dependiendo de la cantidad de capas ocultas(estas se denomina “ocultas” porque sus valores no aparecen en los datos), con esto el modelo debe determinar que información es importante para la clasificación de las clases. Por

último, la capa de salida genera la clasificación del modelo basándose en la información dada por las capas anteriores (Goodfellow, 2016).

### 2.3.5. Evaluación de resultados

La clasificación binaria es de las aplicaciones más comunes y simples que existen, sin embargo, evaluar los modelos dependerá de la necesidad de resolver el problema.

#### Matriz de confusión

La matriz de confusión es de las formas más completas de representar los resultados de clasificación binaria, esta matriz está construida por dos columnas y dos filas en la cual las columnas corresponden a las clases predichas y filas a las clases verdaderas.

| | | **Predicción** | |
|---|---|---|---|
| | | **Positivos** | **Negativos** |
| **Verdadero** | **Positivos** | Verdaderos Positivos (VP) | Falsos Negativos (FN) |
| | **Negativos** | Falsos Positivos (FP) | Verdaderos Negativos (VN) |

Figura 2.7: Matriz de confusión

Las entradas de la diagonal principal (VP y VN) de la matriz de confusión corresponden a clasificaciones correctas, mientras que las otras entradas (FN y FP) nos indican los errores de clasificación que se tienen.

Se llaman verdaderos positivos (VP) a las muestras correctamente clasificadas pertenecientes a la clase positiva y verdaderos negativos (VN) a la clasificación correcta de la clase negativa. Además, s refiere a falsos negativos (FN) a los resultados que debieron ser clasificados como positivos en la predicción y los falsos positivos (FP) corresponden a los resultados clasificados de manera errónea como positivos (Müller,2017).

**Precisión**

La precisión es el número de predicciones correctas (VP y VN) dividido por el número de todas las muestras (VP, VN, FP,FN).

$$\text{precisión} = \frac{VP + VN}{VP + VN + FP + FN} \tag{2.10}$$

**Exhaustividad**

La exhaustividad o recall realiza una medición del número de muestras positivas, que son clasificadas como predicciones positivas. Esta métrica de rendimiento sirve para identificar las muestras positivas y evitar falsos positivos.

$$exhaustividad = \frac{VP}{VP + FN} \tag{2.11}$$

**F1-score**

Se tienen dos medidas importantes, las cuales son precisión y exhaustividad, sin embargo, la métrica F o F1-score es una media armónica que proporciona la relación entre ambas métricas, e incluso puede ser una medida mejor que la precisión en conjuntos de datos de clasificación binaria desbalanceados.

$$F_1 = 2 \cdot \frac{precision \cdot exhaustividad}{precision + exhaustividad} \tag{2.12}$$

## 2.4. Trabajos relacionados

Dado el problema de la detección de la dislexia en niños de preescolar, en la literatura se pueden encontrar distintas investigaciones sobre la detección de la dislexia mediante aprendizaje computacional, utilizando distintos modelos para el diagnóstico. Debido a

esto, se mencionan algunos trabajos relacionados con este proyecto de tesis clasificados mediante métodos basados en instrumentos especializados y métodos basados en software.

### 2.4.1. Métodos basados en instrumentos especializados

El método de (Rello,2015), proponen un conjunto de datos derivado de un experimento de seguimiento ocular con 97 personas con visión normal; 48 de ellos diagnosticados con dislexia y con un rango de edad de 11 a 50 años. Posteriormente, implementó el algoritmo de Máquina de Soporte de Vectores (SVM), para la clasificación de personas con y sin dislexia. Agregado a esto, se tomaron en cuenta tres características principales, las cuales son tiempo de lectura, media de fijaciones y edad del participante, así como validación cruzada de 10 veces al dividir los datos en 10 subconjuntos de datos y con una distribución de los datos aleatoria, con este modelo lograron un 80,18 % de precisión.

Posteriormente, en (Cui,2016) se propuso la detección de la dislexia mediante Máquinas de vectores de soporte lineal (LSVM) y un conjunto de datos recolectando imágenes de resonancia magnética en niños de 11 años en promedio, esto se planteó bajo la hipótesis de que la dislexia se desarrolla por múltiples causas y exhibe múltiples manifestaciones, lo que implica un efecto multidimensional distribuido en el cerebro humano(Cui,2016). Centrándose en buscar diferencias en niños con y sin dislexia, analizando la materia blanca mediante aprendizaje computacional. Durante el proceso de procesamiento de datos fueron seleccionados un total de 50 regiones del cerebro y analizados mediante una técnica de Mapa de Parcelación de Materia Blanca (WMPM), para obtener el conjunto de datos, posteriormente realizaron clasificación de características discriminatorias y eliminaron las características no informativas que se emplearon ampliamente para mejorar el rendimiento de la clasificación, como resultado obtuvieron un clasificador LSVM que realiza la detección a niños disléxicos de los niños no disléxicos con la precisión, sensibilidad, especificidad, de 83,61 %, 75,00 %, 90,91 % respectivamente.

### 2.4.2. Métodos basados en software

En el trabajo de (Lakretz,2015) se emplean modelos bayesianos, los cuales son modelos estadísticos, que se utilizan para modelar matemáticamente supuestos relacionados con la forma en que se generan los datos(los errores de lectura). Además, construyeron un conjunto de datos de 313 personas, de las cuales 97 tenían dislexia, y un rango de edad entre 7 a 62 años. Cabe destacar que no se menciona el uso de preprocesamiento y tampoco selección de características. Con la información recolectada emplearon los modelos Clasificador Bayesiano Ingenuo (naïve bayes) y Asignación Latente de Dirichlet (LDA por sus siglas en inglés *Latent Dirichlet Allocation*), el primer modelo lo emplearon debido a que han mostrado mucho éxito en las tareas de clasificación, por lo tanto, se tendrían ventajas de poder detectar subtipos de dislexia. El segundo modelo LDA permite inferir variables no observadas a partir de datos observados. Los resultados de esta investigación fueron que el método de LDA clasificaba mejor a las personas con dislexia con una métrica de área bajo la curva(AUC) 80.1 %.

Un modelo propuesto por (Rauschenberger, 2022) se propone un diagnóstico en niños utilizando un juego web llamado MusVis el cual está compuesto de una parte fonética y una parte visual en la que los niños completaran tareas de reconocimiento de sonidos y patrones, con la cual se obtendrán los tiempos de respuesta al completar las tareas, así como el número de aciertos y errores al completar cada ejercicio. El conjunto de datos, está formado por registros de 313 niños, de los cuales 116 son clasificados con la dislexia. En el entrenamiento fue realizado con Extra Trees y Random Forest se realiza validación cruzada y selección de características, sin embargo, no se menciona el uso de algún preprocesamiento en los datos. Por último, para el entrenamiento de los algoritmos utilizaron los parámetros predeterminados sugeridos en la biblioteca Scikit-learn. El enfoque que le dan es el de optimizar los recursos para detectar y tratar la dislexia; además, es independiente al idioma que el niño hable. Con esta propuesta para la detección de la

dislexia, tienen el objetivo de seguir recopilando información y poder seguir trabajando con la detección de este trastorno de aprendizaje.

### 2.4.3. Resumen

El aprendizaje computacional es un proceso automatizado, el cual se centra en la construcción de modelos para el análisis predictivo de datos. De los cuatro tipos de aprendizaje propuestos por (Burkov, 2019), el aprendizaje supervisado utiliza datos etiquetados durante su aprendizaje, en cambio, el aprendizaje no supervisado y semisupervisado utiliza datos sin etiquetar o una cantidad mínima de datos etiquetados y por último el aprendizaje por refuerzo se enfoca en la toma de decisiones en secuencia. Para la realización en una investigación sobre un proyecto de ML se debe aplicar una serie de pasos para poder llegar a un resultado, como primer paso se debe obtener los registros o datos para pasar una etapa de preprocesamiento para eliminar información irrelevante, así como una selección de características para reducir la complejidad del modelo resultante y evitar un overfitting, dependiendo de los datos y del objetivo que se tenga en el proyecto se realiza la selección del modelo, como siguiente paso, se entrena el modelo con un ajuste de hiperparámetros y una validación cruzada evaluar la generalización del modelo entrenado, por último, se evalúa el modelo para obtener el rendimiento mediante métricas de interés como recall para evaluar los casos clasificados correctamente o la precisión para evaluar el número de aciertos clasificados, cabe destacar que durante este proceso se pueden realizar ajustes en los pasos anteriores.

En la investigación bibliográfica podemos notar que actualmente existen pocas investigaciones sobre el reconocimiento de la dislexia, agregado a esto, algunas de las investigaciones para la detección de la dislexia conlleva un costo económico elevado para la recolección de datos como lo son los estudios mediante resonancia magnética y los “eye tracker”, con esto, tenemos que el diagnóstico en escuelas puede generar costos que no se

podrían cubrir en escuelas que no cuentan con los recursos para acceder a estas tecnologías. Sin embargo, hay enfoques que tratan de detectar la dislexia, optimizando recursos y empleando herramientas a las cuales pueden acceder las escuelas por un bajo costo, este es el caso de un juego web para recolectar información y reconocer grupos en riesgo de dislexia.

# Capítulo 3

# Método propuesto

Es importante tomar en cuenta el problema que se resolverá, el cual es mejorar la detección mediante una selección óptima de características y algoritmos de aprendizaje computacional, por lo tanto, el planteamiento del problema se resume a lo siguiente: ¿es posible mejorar la detección de dislexia mediante una selección óptima de características y algoritmos de aprendizaje computacional?

## 3.1. Descripción de alto nivel del método propuesto

En la presente investigación se propone un marco de trabajo para la creación de distintos modelos para la detección de dislexia a partir de los datos utilizados por Rauschenberger. En este marco de trabajo se consideran diferentes etapas, las cuales se muestran en la Figura 3.1.

La primera etapa consta de la obtención del conjunto de datos publicado por (Rauschenberger, 2022), el cual contiene información de niños con dislexia (grupo experimental) y sin dislexia (grupo de control); a partir de este conjunto de datos se obtendrán distintos tipos de modelos para la detección de dislexia, al seguir los pasos del método propuesto. Posteriormente, se realiza el procesamiento del conjunto de datos, al eliminar informa-

ción irrelevante para la predicción de dislexia. Y por último, en esta etapa, se aplica una técnica de balanceo de datos. En la segunda etapa, se contempla una selección de características para mejorar el rendimiento del modelo. Así como seleccionar los modelos de aprendizaje computacional, y la ejecución de tres experimentos, los cuales serán modelo tipo I) Entrenar los modelos con las configuraciones por defecto de la biblioteca implementada y el conjunto de datos balanceado, modelo tipo II) Entrenar los modelos mediante selección de características y el conjunto de datos en balanceado y el modelo tipo III) Entrenar con ensamble de modelos y la selección de características. La última etapa comprende el entrenamiento con búsqueda en malla, la evaluación de los modelos y comparación con lo publicado en el estado del arte.

## 3.2. Marco de referencia de trabajo

A continuación se presenta el marco de referencia de trabajo para la implementación del método propuesto y se detalla cada parte que lo conforma.

1. **Obtención del conjunto de datos sobre niños con y sin dislexia.**

   Se tomará el conjunto de datos utilizado en el método de (Rauschenberger, 2022), el cual contiene un total de 313 datos, de los cuales 116 pertenecen a niños diagnosticados con dislexia y el resto pertenece a nuestro grupo de control.

2. **Preprocesamiento del conjunto de datos.**

   En este paso se realizó una limpieza de los datos, comenzando con la eliminación de datos vacíos, incompletos o que no presentan información relevante para nuestro objetivo. Como segundo paso se llevara a cabo un balanceo de datos con la técnica de sobre muestreo (oversampling), debido a que solo se cuentan con 116 registros de infantes con dislexia, esto genera un desbalance en los datos respecto a nuestro grupo de control (197 registros de infantes sin dislexia), es por esto que la técnica de sobre

muestreo es importante para balancear los datos de nuestra clase de interés (Con dislexia), por último, se aplicaran técnicas de escalamiento al conjunto de datos, como lo son escalados de varianza (estandarización), min-max y L2-normalización de los datos cuantitativos con el propósito de trabajar con modelos que aumentan su rendimiento con datos escalados.

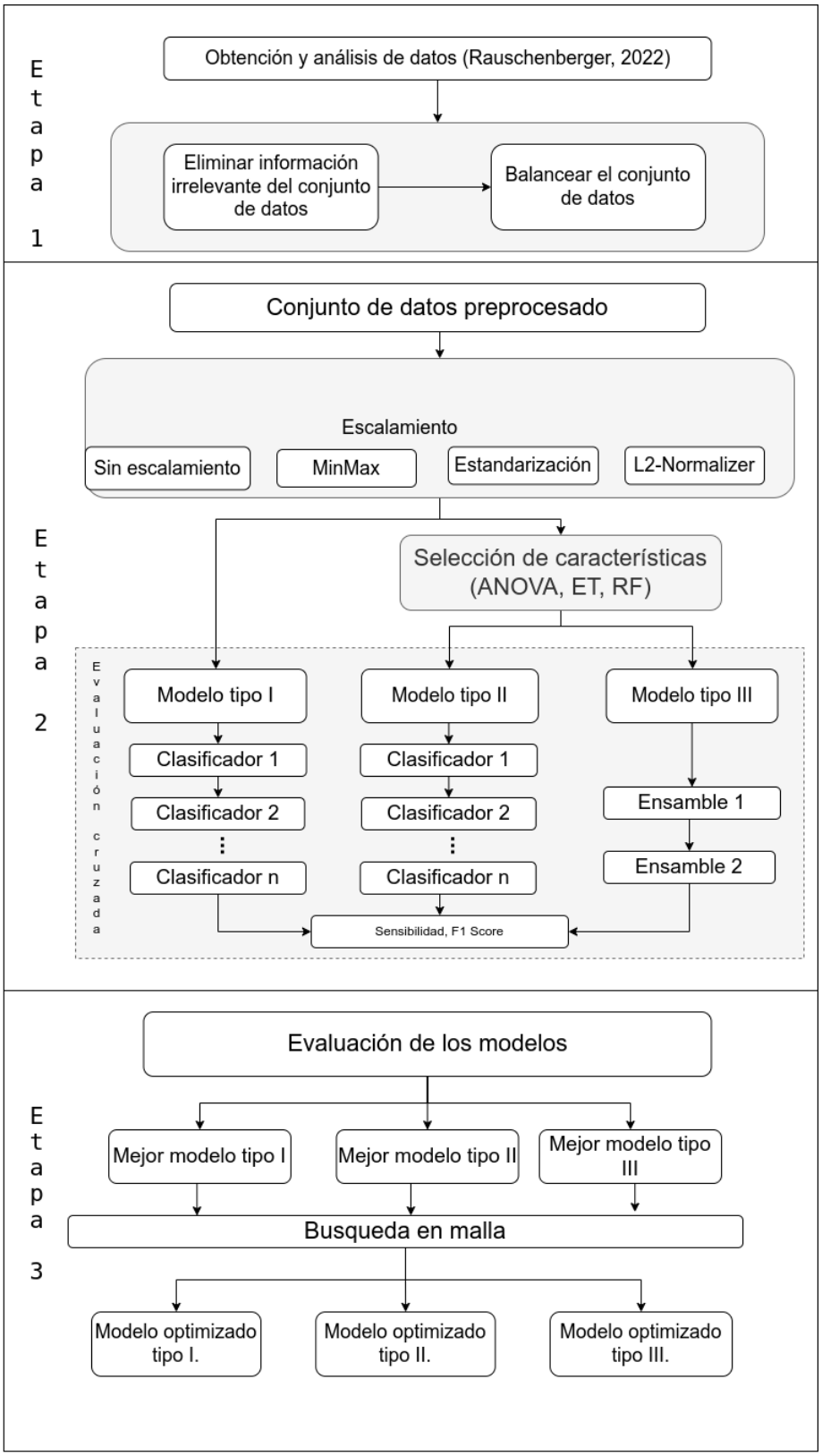


Figura 3.1: Etapas del marco de trabajo propuesto para la detección de dislexia

3. **Selección de características del conjunto de datos.**

   Para la selección de características se utilizará un método de filtro, el cual es ANOVA, para realizar una selección de características que tengan correlación entre las mismas y filtrar las características significativas para los modelos a entrenar. Así mismo, se propone la selección mediante métodos de integración, árboles extremadamente aleatorios y Bosque Aleatorio.

4. **Seleccionar modelo de aprendizaje computacional.**

   Para el entrenamiento y selección de modelo se proponen realizar distintos entrenamientos para los modelos de tipo I y tipo II con siete modelos, los cuales son: Bayes Ingenuo (NB), Regresión Logística (LR), Máquina de Soporte de Vectores (SVM), Árboles Extremadamente Aleatorios (ET), Bosque Aleatorio (RF), Perceptrón Multicapa (MLP) y Potenciación del Gradiente (GB), adicionalmente a esto se sugieren realizar entrenamientos con un modelo tipo III el cual está conformado por un ensamble de modelos, el primer ensamble estará conformado por los siete modelos mencionados anteriormente y un segundo modelo considerando cuatro de los mejores modelos.

5. **Entrenamiento del modelo**

   En esta parte del método se realiza la experimentación con el conjunto de datos balanceado y tres técnicas de escalamiento(Estandarización, Min-Max y L2-Normalización) para los tres tipos de modelos, además, se realiza selección de características para los modelos tipo II y tipo III, los cuales se conforman por las técnicas de selección de características (ANOVA, árboles extremadamente aleatorios y Bosque Aleatorio). Adicionalmente, a esto, en los entrenamientos se utilizará validación cruzada para conocer el rendimiento del modelo. Por último, se tomarán en cuenta las métricas de sensibilidad y F1 score para evaluar los modelos resultan-

tes.

6. **Evaluación de resultados obtenidos y comparación con los publicados en el estado del arte.**

   Por último, en esta parte del método se realizará una comparación entre los modelos realizados en el paso 5 para realizar una búsqueda en malla con los mejores modelos tipo I, tipo II y tipo III. Finalmente, comparar los resultados con los publicados en la investigación de (Rauschenberger, 2022).

## 3.3. Resumen

En este capítulo es importante mencionar que el método propuesto está diseñado para comprobar la hipótesis planteada en esta investigación, es por esto que se presenta un método conformado por tres etapas con subetapas con un propósito en específico en cada una de ellas. En la primera etapa está enfocada en la limpieza y balance de los datos, en la segunda etapa se llevarán a cabo experimentos con distintos clasificadores con el objetivo de encontrar los mejores modelos por cada tipo, tomando en cuenta distintos escalamientos, selección de características, así como validación cruzada para poder observar la generalización que tienen los clasificadores. En la última etapa se tomaba el mejor clasificador por cada tipo y se realizará una búsqueda en malla para optimizar los modelos, así como comparar los resultados obtenidos con los resultados de la investigación de (Rauschenberger, 2022).

# Capítulo 4

# Resultados

El problema de la detección de dislexia mediante modelos de aprendizaje automático bajo un enfoque supervisado requiere del uso de datos etiquetados, en el trabajo de (Rauschenberger, 2022) se utilizaron datos obtenidos de un juego web independiente del lenguaje para detectar si un niño tiene dislexia o no. En esta tesis se planteó la hipótesis de que el uso de técnicas de selección de características del conjunto de datos y la configuración correcta de los hiperparámetros en algoritmos de aprendizaje computacional, mejorará la detección de dislexia en niños de preescolar.

En este capítulo, se presentan los resultados del método propuesto, comenzando con un análisis del conjunto de datos. Posteriormente, se muestran los resultados de tres experimentos donde se obtuvieron siete modelos mediante el entrenamiento con un conjunto de datos balanceado mediante una técnica de sobre muestreo denominada SMOTE. Los resultados de los experimento corresponden a los algoritmos de aprendizaje computacional, los cuales son: Bayes Ingenuo (NB), Regresión Logística (LR), Máquina de Soporte de Vectores (SVM), Árboles Extremadamente Aleatorios (ET), Bosque Aleatorio (RF), Perceptrón Multicapa (MLP) y Potenciación del Gradiente (GB).

Durante la fase de experimentación se utilizó una computadora con un sistema operativo Debian GNU/Linux 11 (bullseye) x86_64, kernel: 5.10.0-28-amd6, CPU: Intel i7-8550U (8) a 4.000 GHz y Memoria RAM de 16 Gb. Además, el desarrollo de los experimentos fue en el lenguaje Python versión 3.11.0, las librerías de scikit-learn con versión: 1.3.0 y imbalanced-learn versión: 0.12.0.

## 4.1. Conjunto de datos

El conjunto de datos utilizado en la experimentación se seleccionó de un repositorio público de University of Applied Science Emden/Leer (Rauschenberger, 2022). El conjunto contiene información del juego web para detección de dislexia "MusVis". El juego recompila información en general sobre tiempos de respuestas y números de errores sobre el desempeño de un conjunto de pruebas visuales y auditivas de niños que tenían un diagnóstico previo de dislexia. Una característica esencial del juego fue que no depende de que el niño tenga habilidades de lectura o dominio de un idioma en particular. El conjunto de datos fue etiquetado a partir de diagnóstico previamente realizado por un especialista a cada infante participante. El conjunto de datos está compuesto por 116 infantes etiquetados con dislexia son 116 y 197 niños sin dislexia, sumando un total de 313 registros.

### 4.1.1. Análisis descriptivo de los atributos de la base de datos MusVis

En este paso, se realizó un análisis de cada uno de los atributos del conjunto de datos MusVis, de acuerdo a (Rauschenberger, 2022). En la Tabla 4.1 se muestra información adicional relacionada con los participantes.

Tabla 4.1: *Información de los niños participantes en MusVis*

| **Características de los participantes** | **Descripción** |
|---|---|
| Edad | Los infantes tienen entre 7 y 12 años de edad. |
| Infantes masculinos | 153 |
| Infantes femeninos | 160 |
| Lenguaje | Español, alemán e inglés. |

De acuerdo con las edades de los niños participantes, en la Tabla 4.2 se muestra la distribución de las edades separadas por diagnóstico.

Tabla 4.2: Edades de los niños en el conjunto de datos MusVis

| **Edad** | **Sin dislexia** | **Con dislexia** |
|---|---|---|
| 7 | 24 | 11 |
| 8 | 38 | 9 |
| 9 | 13 | 23 |
| 10 | 46 | 38 |
| 11 | 39 | 22 |
| 12 | 37 | 13 |

Durante la interacción con el juego, por cada niño que participó se obtuvieron 201 características, que incluyen características auditivas y visuales, de cuerdo al diseño de los retos con los que interactúa el niño (ver Tabla 4.3).

Tabla 4.3: Características visuales y auditivas del conjunto de datos MusVis

| Rango | **Característica Auditiva** | Rango | **Característica Visual** |
|---|---|---|---|
| 10-17 | Hora de hacer clic | 106-113 | Hora de hacer clic |
| 18-25 | Clics totales | 114-121 | Clics totales |
| 26-33 | Duración por ronda | 122-129 | Respuestas correctas |
| 34-41 | Duración interacción | 130-137 | Respuestas incorrectas |
| 42-49 | Tiempo medio de clic | 138-145 | Precisión |
| 50-57 | Lógica | 146-153 | Eficiencia |
| 58-65 | 2° clic | 154-161 | 2° clic |
| 66-73 | 3° clic | 162-169 | 3° clic |
| 74-81 | 4° clic | 170-177 | 4° clic |
| 82-89 | 5° clic | 178-185 | 5° clic |
| 90-97 | 6° clic | 186-193 | 6° clic |
| 98-105 | Instrucciones | 194-201 | Hora de último clic |

Adicional a estas características, se incluye información demográfica, las cuales se describen en la Tabla 4.4, cabe mencionar que en cuestiones de lenguaje, los participantes hablaban hasta dos lenguajes, como por ejemplo español y catalán o inglés-alemán

Tabla 4.4: Características demográficas de los participantes

| **Características** | **Descripción** |
|---|---|
| Edad | El rango de edad de los infantes está en el rango de 7 a 12 años. |
| Género | Es una característica binaria, con valor *femenino* o *masculino* |
| Lenguaje | Los idiomas que los infantes hablaban *español*, *catalán*, *alemán* o *inglés* |
| Lenguaje nativo | Para saber si el idioma utilizado para las instrucciones es el primer idioma de los participantes, siendo *Sí*, *No* o *Tal vez*. |
| Instrumento | Muestra si el participante toca un instrumento musical, siendo *No*, *menos de 6 meses* o *más de 6 meses*. |
| Memoria | Indica hasta qué punto el participante conoce el juego de Memoria visual, siendo *Participante no dio ninguna respuesta*, *Participante no conoce el juego*, *Jugó una vez*, *Jugó algunas veces* o *Jugó mucho*. |
| Calificación de la parte auditiva | Indica la respuesta autoinformada con una escala Likert de 6 niveles Field y Hole, (2003) a la afirmación: "la parte auditiva fue fácil para los participantes". Los valores son *Respuesta desconocida*, *Totalmente en desacuerdo*,*En desacuerdo*, *Indeciso*, *De acuerdo* o *Totalmente de acuerdo*. |
| Calificación de la parte visual | Indica la respuesta autodeclarada de la afirmación: "la parte visual fue fácil para los participantes". (misma escala Likert de la característica 7) |
| Dispositivo | Es el dispositivo que utilizaron los participantes y es una característica binaria, con el valor para *computadora* o *tableta* |

## 4.2. Preprocesamiento

Con la finalidad de realizar los experimentos, a partir del conjunto de datos MusVis, en este primer paso se realiza la limpieza, separación, transformación y balanceo de datos.

### 4.2.1. Limpieza de los datos

Durante el preprocesamiento, se realiza la eliminación de algunas columnas, principalmente las variables que identifican a los infantes o tipo de navegador usado durante el juego, dado que se consideran irrelevante por la naturaleza de la información. Durante un primer análisis de los datos se identificaron características que no aportan información o relación con la variable dependiente. En la Tabla 4.5, se listan las características a descartar y que están asociadas con la base de datos de la funcionalidad requerida de la aplicación, por lo tanto, quedan fuera del tema de investigación.

Tabla 4.5: Características eliminadas durante la limpieza.

| **Característica eliminada** | **Descripción** |
|---|---|
| id | Valor numérico que identifica el registro del usuario (niño) que participó en el juego. |
| db_browser | Valor categórico que representa el navegador web utilizado por el usuario. |
| mus_1_hashcode | Código alfanumérico de identificación. |

### 4.2.2. Separación de datos en entrenamiento y prueba

El conjunto de datos MusVis ha sido separado en un 80 % para la fase de entrenamiento y el 20 % restante para la prueba. En la Tabla 4.6 se muestra la distribución de la clase dislexia (clase 1) y la clase sin dislexia (clase 0) para ambos subconjuntos.

Tabla 4.6: Separación de datos en entrenamiento y prueba

| Clase | MusVis-Entrenamiento | MusVis-Prueba |
|---|---|---|
| Dislexia | 93 | 23 |
| No-dislexia | 157 | 40 |

## 4.3. Experimento 1: Modelo tipo I con el conjunto de datos MusVis-Entrenamiento balanceado

En este experimento, la base de datos MusVis fue balanceada mediante la técnica de oversamplig SMOTE. La base de datos MusVis-Entrenamiento desbalanceada está compuesta por 157 casos sin dislexia y 93 con dislexia, dando un total de 250 datos, sin embargo, al realizar la técnica de SMOTE, la distribución de los datos fue de 50-50 % para cada clase.

### 4.3.1. MusVis-Entrenamiento balanceado

En la Tabla 4.7 se muestran los resultados de la validación cruzada k-fold con un k=5 para los modelos: Bayes Ingenuo (NB), Regresión Logística (LR), Máquina de Soporte de Vectores (SVM), Árboles Extremadamente Aleatorios (ET), Bosque Aleatorio (RF), Perceptrón Multicapa (MLP) y Potenciación del Gradiente (GB). Si se considera una evaluación general del rendimiento de los modelos mediante la métrica de F1, el modelo RL alcanza un 56.04 % en F1, con una exactitud balanceada de 57 %, seguida de GB, y RF con un F1 de 54.3 % y con una exactitud balanceada de 54.7 %, 51.6 % y 52.4 % respectivamente.

Tabla 4.7: Validación cruzada kfold con un k = 5 en el conjunto de datos MusVis balanceado.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | S | F1 | B Acc | DE. (S) | S | F1 | B Acc | DE. (S) |
| RL | 100.0 % | 100.0 % | 100.0 % | 0 | 57.0 % | 56.04 % | 57.0 % | 0.117 |
| GB | 100.0 % | 100.0 % | 100.0 % | 0 | 54.7 % | 54.3 % | 54.7 % | 0.079 % |
| RF | 100.0 % | 100.0 % | 100.0 % | 0 | 52.4 % | 51.6 % | 52.4 % | 0.079 |
| MLP | 97.9 % | 97.9 % | 97.9 % | 0.027 | 51.2 % | 51.1 % | 27.0 % | 0.066 |
| ET | 100.0 % | 100.0 % | 100.0 % | 0 | 49.8 % | 46.9 % | 49.8 % | 0.056 |
| NB | 77.0 % | 76.8 % | 77.0 % | 0.021 % | 47. % 3 | 46.4 % | 47.3 % | 0.084 |
| SVM | 59.5 % | 55.9 % | 59.5 % | 0.052 % | 47.2 % | 44.5 % | 47.02 % | 0.061 |

Nota: Resultados de Sensibilidad (S), Medida (F1), Exactitud balanceada (B Acc) y Desviación Estándar (DE (S)) de la validación cruzada con un k = 5, en los modelos: Regresión Logística (LR), Potenciación del Gradiente (GB), Bosque Aleatorio (RF), Perceptrón Multicapa (MLP), Árboles Extremadamente Aleatorios (ET), Bayes Ingenuo (NB) y Máquina de Soporte de Vectores (SVM) en el conjunto de datos MusVis balanceado.

Es importante recordar que la clase de interés es la clase de dislexia (clase 1). Por lo que, la métrica de sensibilidad implica medir la recuperación de mayor cantidad de verdaderos positivos (niños con dislexia) sin importar la cantidad de falsos verdaderos (niños que no tienen dislexia, pero fueron clasificados con dislexia). En una situación hipotética de implementación de una herramienta de computación inteligente, es preferible diagnosticar niño sin dislexia como disléxico y que en un siguiente diagnóstico (realizado por el experto) se descarte, en contraste a un escenario fatal, donde se diagnostique a un niño disléxico sin dislexia y, por lo tanto, no sea tratado por un profesional. Por lo tanto, si se analizan los resultados por clase y con la métrica de sensibilidad en la validación cruzada (ver Tabla 4.8) el modelo RL supera a todos los modelos entrenados con una sensibilidad de 48.5 % en la clase de dislexia.

Tabla 4.8: Validación cruzada con un k = 5 en el conjunto de datos MusVis balanceado por clase.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | Sin dislexia | | Con dislexia | | Sin dislexia | | Con dislexia | |
| | S | F1 | S | F1 | S | F1 | S | F1 |
| RL | 100 % | 100 % | 100 % | 100 % | 61.1 % | 64.2 % | 48.5 % | 48.5 % |
| GB | 100 % | 100 % | 100 % | 100 % | 69.4 % | 67.7 % | 40.0 % | 0.41 % |
| MLP | 98.2 % | 979 % | 97.6 % | 97.8 % | 63.5 % | 63.3 % | 38.8 % | 38.8 % |
| NB | 79.4 % | 77.3 % | 74.6 % | 76.3 % | 62.2 % | 61.2 % | 32.3 % | 31.6 % |
| SVM | 71.5 % | 61.4 % | 47.5 % | 50.5 % | 63.5 % | 59.9 % | 31.0 % | 29.0 % |
| RF | 100 % | 100 % | 100 % | 100 % | 74.5 % | 69.0 % | 30.3 % | 34.2 % |
| ET | 100 % | 100 % | 100 % | 100 % | 82.2 % | 71.0 % | 17.4 % | 22.9 % |

Nota: Resultados de Sensibilidad (S) y F1 de la validación cruzada con un k = 5 en los modelos: Bayes Ingenuo (NB), Regresión Logística (LR), Máquina de Soporte de Vectores (SVM), Árboles Extremadamente Aleatorios (ET), Bosque Aleatorio (RF), Perceptrón Multicapa (MLP) y Potenciación del Gradiente (GB) en el conjunto de datos MusVis balanceado por clase.

## 4.4. Experimento 2: Modelo tipo II con el conjunto de datos MusVis-Entrenamiento balanceado y con selección de características

En este experimento, previo al proceso de entrenamiento de los modelos, primero se realizó una selección de características. Las técnicas utilizadas fueron: Árboles Extremadamente Aleatorios, ANOVA y Bosque Aleatorio. A continuación se muestran los resultados obtenidos con Árboles Extremadamente Aleatorios y con ANOVA (los resultados de Bosque Aleatorio se omiten, dado que se obtuvo un rendimiento inferior a los otros dos métodos, ver Tabla 5.4.2 en el anexo, para los resultados de Bosque Aleatorio).

### 4.4.1. Entrenamiento MusVis balanceado y selección de características mediante Árboles Extremadamente Aleatorios

En el trabajo (Rauchenberger, 2022) se utiliza el método de selección de características Árboles Extremadamente Aleatorios, por lo que en este experimento se utiliza esta técnica para mejorar los resultados obtenidos en el experimento de la sección 4.3.1.

En las Tablas 4.9, 4.10, 4.11,4.12, se muestran los resultados de la validación cruzada k-fold con un k=5 para los modelos: Bayes Ingenuo (NB), Regresión Logística (LR), Máquina de Soporte de Vectores (SVM), Árboles Extremadamente Aleatorios (ET), Bosque Aleatorio (RF), Perceptrón Multicapa (MLP) y Potenciación del Gradiente (GB). Considerando las tres técnicas de escalamiento y la selección de características de Árboles Extremadamente Aleatorios. Los modelos GB-MinMax, GB-Estandar y GB con un F1 obtienen 71.4 %, 69.1 % y 66.1 %, respectivamente en la prueba de k-fold, superando el mejor resultado obtenido en el experimento uno, donde RL obtuvo un 56.04 % de F1 en el conjunto de prueba (Tabla 4.7).

Es necesario, evaluar los modelos por la capacidad de recuperación de los casos de niños con dislexia. Por lo que, en las tablas 4.13, 4.14,4.15,4.16, se muestran los resultados de evaluación ordenados por la sensibilidad de la clase con dislexia. Los resultados de evaluación de los modelos muestra que en comparación con el resultado obtenido en el experimento uno por clase (modelo RL con un 48.5 % de sensibilidad de en la clase dislexia, ver Tabla 4.8) se obtuvieron ocho modelos que mejoran la recuperación en la clase dislexia, el modelo GB-MinMax obtuvo el valor más alto con 61.1 %.

Tabla 4.9: Resultados de F1 en MusVis balanceada sin escalamiento.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | S | F1 | B Acc | DE. (S) | S | F1 | B Acc | DE. (S) |
| GB | 100 % | 100 % | 100 % | 0 | 66.0 % | 66.1 % | 66.0 % | 0.052 |
| RL | 100 % | 100 % | 100 % | 0 | 59.8 % | 59.4 % | 59.8 % | 0.076 |
| ET | 100 % | 100 % | 100 % | 0 | 56.0 % | 55.6 % | 56.0 % | 0.086 |
| MLP | 89.5 % | 89.4 % | 89.5 % | 0.118 | 53.2 % | 52.6 % | 53.2 % | 0.067 |
| NB | 77.5 % | 77.3 % | 77.5 | 0.025 | 51.9 % | 51.6 % | 51.9 % | 0.045 |
| RF | 100 % | 100 % | 100 % | 0 | 51.5 % | 50.9 % | 51.5 % | 0.063 |
| SVM | 62.7 % | 62.0 % | 62.7 % | 0.022 | 46.5 % | 44.0 % | 46.5 % | 0.055 |

Nota: Resultados de Sensibilidad (S), Medida (F1), Exactitud balanceada (B Acc) y Desviación Estándar (DE (S)), utilizando validación cruzada con un k = 5 en los modelos: Potenciación del Gradiente (GB), Regresión Logística (LR), Árboles Extremadamente Aleatorios (ET), Perceptrón Multicapa (MLP), Bayes Ingenuo (NB), Bosque Aleatorio (RF) y en el conjunto de datos MusVis balanceado, utilizando selección de características mediante Árboles Extremadamente Aleatorios y sin escalamiento.

Tabla 4.10: Resultados de F1 en MusVis balanceada con normalización.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | S | F1 | B Acc | DE. (S) | S | F1 | B Acc | DE. (S) |
| GB-Normalizer | 100 % | 100 % | 100 % | 0 | 60.7 % | 60.5 % | 60.7 % | 0.074 |
| MLP-Normalizer | 99.7 % | 99.7 % | 99.7 % | 0.005 | 55.8 % | 55.6 % | 55.8 % | 0.063 |
| RF-Normalizer | 100 % | 100 % | 100 % | 0 | 54.6 % | 54.4 % | 54.6 % | 0.091 |
| ET-Normalizer | 100 % | 100 % | 100 % | 0 | 54.1 % | 53.2 % | 54.1 % | 0.051 |
| NB-Normalizer | 71.1 % | 71.5 % | 71.5 % | 0.011 | 54.2 % | 52.8 % | 54.2 % | 0.099 |
| RL-Normalizer | 61.8 % | 61.7 % | 61.8 % | 0.027 | 49.7 % | 48.4 % | 49.7 % | 0.062 |
| SVM-Normalizer | 65.7 % | 65.0 % | 65.7 % | 0.046 | 48.7 % | 45.6 % | 48.7 % | 0.042 |

Nota: Resultados de Sensibilidad (S), Medida (F1), Exactitud balanceada (B Acc) y Desviación Estándar (DE (S)), utilizando validación cruzada con un k = 5 en los modelos: Potenciación del Gradiente (GB), Perceptrón Multicapa (MLP), Bosque Aleatorio (RF), Árboles Extremadamente Aleatorios (ET), Bayes Ingenuo (NB), Regresión Logística (LR) y Máquina de Soporte de Vectores (SVM) en el conjunto de datos MusVis balanceado, utilizando selección de características mediante Árboles Extremadamente Aleatorios y con escalamiento L2-Normalizer.

Tabla 4.11: Resultados de F1 en MusVis balanceada con escalamiento Min-Max.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | S | F1 | B Acc | DE. (S) | S | F1 | B Acc | DE. (S) |
| GB-MinMax | 100 % | 100 % | 100 % | 0 | 71.3 % | 71.4 % | 71.3 % | 0.064 |
| RF-MinMax | 100 % | 100 % | 100 % | 0 | 56.8 % | 56.7 % | 56.8 % | 0.059 |
| ET-MinMax | 100 % | 100 % | 100 % | 0 | 56.2 % | 55.4 % | 56.2 % | 0.041 |
| RL-MinMax | 86 % | 86 % | 86 % | 0.036 | 55.6 % | 55.4 % | 55.6 % | 0.116 |
| MLP-MinMax | 100 % | 100 % | 100 % | 0 | 54.9 % | 54.7 % | 54.9 % | 0.037 |
| SVM-MinMax | 98.0 % | 98.0 % | 98.0 % | 0.006 | 55.2 | 53.3 % | 55.2 % | 0.027 |
| NB-MinMax | 80.2 % | 80.1 % | 80.2 % | 0.017 | 51.8 % | 50.6 % | 51.8 % | 0.025 |

Nota: Resultados de Sensibilidad (S), Medida (F1), Exactitud balanceada (B Acc) y Desviación Estándar (DE (S)), utilizando validación cruzada con un k = 5 en los modelos: Potenciación del Gradiente (GB), Bosque Aleatorio (RF), Árboles Extremadamente Aleatorios (ET), Regresión Logística (LR), Perceptrón Multicapa (MLP), Máquina de Soporte de Vectores (SVM) y Bayes Ingenuo (NB) en el conjunto de datos MusVis balanceado, utilizando selección de características mediante Árboles Extremadamente Aleatorios y escalamiento Min-Max.

Tabla 4.12: Resultados de F1 en MusVis balanceada con escalamiento estándar.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | S | F1 | B Acc | DE. (S) | S | F1 | B Acc | DE. (S) |
| GB-Estandar | 100 % | 100 % | 100 % | 0 | 69.0 % | 69.1 % | 69.0 % | 0.068 |
| RL-Estandar | 99.9 % | 99.9 % | 99.9 % | 0.001 | 61.4 % | 61.3 % | 61.4 % | 0.135 |
| MLP-Estandar | 100 % | 100 % | 100 % | 0 | 60.2 % | 60.3 % | 60.2 % | 0.132 |
| ET-Estandar | 100 % | 100 % | 100 % | 0 | 59.9 % | 59.3 % | 59.9 % | 0.055 |
| RF-Estandar | 100 % | 100 % | 100 % | 0 | 53.3 % | 53.0 % | 53.3 % | 0.081 |
| NB-Estandar | 76.7 % | 76.5 % | 76.7 % | 0.023 | 54.1 % | 53.0 % | 54.1 % | 0.057 |
| SVM-Estandar | 97.7 % | 97.7 % | 97.7 % | 0.009 | 53.5 % | 51.2 % | 53.5 % | 0.029 |

Nota: Resultados de Sensibilidad (S), Medida (F1), Exactitud balanceada (B Acc) y Desviación Estándar (DE (S)), utilizando validación cruzada con un k = 5 en los modelos: Potenciación del Gradiente (GB), Regresión Logística (LR), Perceptrón Multicapa (MLP), Árboles Extremadamente Aleatorios (ET), Bosque Aleatorio (RF), Bayes Ingenuo (NB) y Máquina de Soporte de Vectores (SVM) en el conjunto de datos MusVis balanceado, utilizando selección de características mediante Árboles Extremadamente Aleatorios y escalamiento estándar.

Tabla 4.13: Resultados de sensibilidad por clase en MusVis sin escalamiento.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | Sin dislexia | | Con dislexia | | Sin dislexia | | Con dislexia | |
| | S | F1 | S | F1 | S | F1 | S | F1 |
| GB | 100 % | 100 % | 100 % | 100 % | 78.3 % | 76.0 % | 53.7 % | 56.3 % |
| RL | 100 % | 100 % | 100 % | 100 % | 66.8 % | 68.4 % | 52.7 % | 50.3 % |
| NB | 74.2 % | 76.5 % | 80. %8 | 78.2 % | 58.6 % | 61.2 % | 45.2 % | 42.0 % |
| MLP | 89.1 % | 89.4 % | 89.9 % | 89.5 % | 62.3 % | 63.2 % | 44.1 % | 42.1 % |
| SVM | 68.1 % | 64.3 % | 57.3 % | 59.7 % | 57.3 % | 57.4 % | 35.6 % | 30.6 % |
| ET | 100 % | 100 % | 100 % | 100 % | 80.9 % | 72.8 % | 31.2 % | 38.5 % |
| RF | 100 % | 100 % | 100 % | 100 % | 75.1 % | 68.8 % | 28.0 % | 33.0 % |

Nota: Resultados de Sensibilidad (S), F1 utilizando validación cruzada con un k = 5 en los modelos: Potenciación del Gradiente (GB), Regresión Logística (LR), Bayes Ingenuo (NB), Perceptrón Multicapa (MLP), Máquina de Soporte de Vectores (SVM), Árboles Extremadamente Aleatorios (ET) y Bosque Aleatorio (RF) en el conjunto de datos MusVis balanceado por clase, utilizando selección de características mediante Árboles Extremadamente Aleatorios y sin escalamiento.

Tabla 4.14: Resultados de sensibilidad por clase en MusVis con escalador normalizer.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | Sin dislexia | | Con dislexia | | Sin dislexia | | Con dislexia | |
| | S | F1 | S | F1 | S | F1 | S | F1 |
| NB-Normalizer | 59.5 % | 67.5 % | 83.5 % | 74.6 % | 51.7 % | 57.9 % | 56.7 % | 47.7 % |
| SVM-Normalizer | 52.7 % | 60.4 % | 78.8 % | 69.6 % | 41.5 % | 48.1 % | 56.0 % | 43.1 % |
| GB-Normalizer | 100 % | 100 % | 100 % | 100 % | 68.8 % | 69.4 % | 52.6 % | 51.6 % |
| RL-Normalizer | 62.5 % | 62.1 % | 61.1 % | 61.4 % | 52.9 % | 57.0 % | 46.4 % | 39.8 % |
| MLP-Normalizer | 99.0 % | 99.7 % | 100 % | 99.7 % | 67.5 % | 66.8 % | 44.1 % | 44.4 % |
| RF-Normalizer | 100 % | 100 % | 100 % | 100 % | 73.8 % | 69.2 % | 35.4 % | 39.7 % |
| ET-Normalizer | 100 % | 100 % | 100 % | 100 % | 80.2 % | 71.9 % | 28.1 % | 34.6 % |

Nota: Resultados de Sensibilidad (S), F1 utilizando validación cruzada con un k = 5 en los modelos: Bayes Ingenuo (NB), Máquina de Soporte de Vectores (SVM), Potenciación del Gradiente (GB), Regresión Logística (LR), Perceptrón Multicapa (MLP), Bosque Aleatorio (RF) y Árboles Extremadamente Aleatorios (ET) en el conjunto de datos MusVis balanceado por clase, utilizando selección de características mediante Árboles Extremadamente Aleatorios y escalamiento normalizer.

Tabla 4.15: Resultados de sensibilidad por clase en MusVis con escalador Min-Max.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | Sin dislexia | | Con dislexia | | Sin dislexia | | Con dislexia | |
| | S | F1 | S | F1 | S | F1 | S | F1 |
| GB-MinMax | 100 % | 100 % | 100 % | 100 % | 81.5 % | 79.5 % | 61.1 % | 63.3 % |
| RL-MinMax | 87.7 % | 86.2 % | 84.4 % | 85.8 % | 68.2 % | 67.5 % | 43.1 % | 43.4 % |
| MLP-MinMax | 100 % | 100 % | 100 % | 100 % | 68.9 % | 67.2 % | 40.9 % | 42.1 % |
| NB-MinMax | 81.6 % | 80.4 % | 78.8 % | 79.8 % | 63.7 % | 62.9 % | 39.8 % | 38.2 % |
| RF-MinMax | 100 % | 100 % | 100 % | 100 % | 78.3 % | 72.1 % | 35.3 % | 41.3 % |
| ET-MinMax | 100 % | 100 % | 100 % | 100 % | 83.4 % | 73.9 % | 29.1 % | 37.0 % |
| SVM-MinMax | 99.5 % | 98.0 % | 96.4 % | 97.9 % | 87.9 % | 75.1 % | 22.6 % | 31.4 % |

Nota: Resultados de Sensibilidad (S), F1 utilizando validación cruzada con un k = 5 en los modelos: Potenciación del Gradiente (GB), Regresión Logística (LR), Perceptrón Multicapa (MLP), Bayes Ingenuo (NB), Bosque Aleatorio (RF), Árboles Extremadamente Aleatorios (ET) y Máquina de Soporte de Vectores (SVM) en el conjunto de datos MusVis balanceado por clase, utilizando selección de características mediante Árboles Extremadamente Aleatorios y escalamiento MinMax.

Tabla 4.16: Resultados de sensibilidad por clase en MusVis con escalador Estandar.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | Sin dislexia | | Con dislexia | | Sin dislexia | | Con dislexia | |
| | S | F1 | S | F1 | S | F1 | S | F1 |
| GB-Estandar | 100 % | 100 % | 100 % | 100 % | 78.9 % | 77.6 % | 59.1 % | 60.5 % |
| RL-Estandar | 100 % | 99.9 % | 99.8 % | 99.9 % | 70.1 % | 70.7 % | 52.6 % | 51.9 % |
| MLP-Estandar | 100 % | 100 % | 100 % | 100 % | 73.3 % | 71.6 % | 47.2 % | 49.0 % |
| NB-Estandar | 76.7 % | 76.55 | 76.7 % | 76.5 % | 65.1 % | 64.1 % | 43.2 % | 41.9 % |
| ET-Estandar | 100 % | 100 % | 100 % | 100 % | 85.3 % | 76.1 % | 34.5 % | 42.6 % |
| RF-Estandar | 100 % | 100 % | 100 % | 100 % | 73.2 % | 68.4 % | 33.3 % | 37.6 % |
| SVM-Estandar | 99.3 % | 97.8 % | 96.1 % | 97.7 % | 85.3 % | 73.6 % | 21.6 % | 28.9 % |

Nota: Resultados de Sensibilidad (S), F1 utilizando validación cruzada con un k = 5 en los modelos: Potenciación del Gradiente (GB), Regresión Logística (LR), Perceptrón Multicapa (MLP), Bayes Ingenuo (NB), Árboles Extremadamente Aleatorios (ET), Bosque Aleatorio (RF) y Máquina de Soporte de Vectores (SVM) en el conjunto de datos MusVis balanceado por clase, utilizando selección de características mediante Árboles Extremadamente Aleatorios y escalamiento estandar.

### 4.4.2. Entrenamiento MusVis balanceado y selección de características ANOVA

En esta sección se analizan los resultados obtenidos mediante la técnica de selección de características denominada como análisis de varianza (ANOVA), esta técnica selección es de tipo filtro, el cual calcula si existe una relación estadísticamente significativa entre cada característica y la clase. Esta técnica de selección tiene la propiedad de ser univariantes, lo cual significa que solo consideran cada característica individualmente, es decir, que una característica se descarta si solo es informativa cuando se combina con otra. Además, pueden ser muy rápidas de calcular y no requieren la construcción de un modelo, y funciona de manera independiente del modelo.

En los resultados, se presenta la aplicación de ANOVA junto con tres técnicas de escalamiento: Min-Max, Estandar y L2-Normalizer. Posteriormente, se entrenaron los modelos: Bayes Ingenuo (NB), Regresión Logística (LR), Máquina de Soporte de Vectores (SVM), Árboles Extremadamente Aleatorios (ET), Bosque Aleatorio (RF), Perceptrón Multicapa (MLP) y Potenciación del Gradiente (GB) utilizando la base de datos MusVis balanceada con SMOTE.

En las tablas 4.17, 4.18, 4.19, 4.20 se muestran los resultados de la validación cruzada k-fold con un k=5 para los modelos: Bayes Ingenuo (NB), Regresión Logística (LR), Máquina de Soporte de Vectores (SVM), Árboles Extremadamente Aleatorios (ET), Bosque Aleatorio (RF), Perceptrón Multicapa (MLP) y Potenciación del Gradiente (GB) considerando las tres técnicas de escalamiento. Si se considera el desempeño general de los modelos, el modelo RL-estandar con un F1 de 56.6 % supera al mejor modelo obtenido en el experimento uno, que es RL con F1 de 56.04 % (ver Tabla 4.7).

Tabla 4.17: Resultados de F1 en MusVis balanceada sin escalamiento.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | S | F1 | B Acc | DE. (S) | S | F1 | B Acc | DE. (S) |
| RL | 91.1 % | 91.1 % | 91.1 % | 0.034 | 56.5 % | 56.0 % | 56.5 % | 0.054 |
| GB | 100 % | 100 % | 100 % | 0 | 54.9 % | 54.6 % | 54.9 % | 0.11 |
| NB | 70.2 % | 69.6 % | 70.2 % | 0.049 | 53.9 % | 53.3 % | 53.9 % | 0.046 |
| MLP | 96.4 % | 96.3 % | 96.4 % | 0.043 | 54.0 % | 52.7 % | 54.0 % | 0.064 |
| ET | 100 % | 100 % | 100 % | 0 | 52.7 % | 51.0 % | 52.7 % | 0.056 |
| RF | 100 % | 100 % | 100 % | 0 | 50 %3 | 50 % | 50.3 % | 0.083 |
| SVM | 63.1 % | 60.7 % | 63.1 % | 0.059 | 48.5 % | 44.2 % | 48.5 % | 0.036 |

Nota: Resultados de Sensibilidad (S), Medida (F1), Exactitud balanceada (B Acc) y Desviación Estándar (DE (S)), utilizando validación cruzada con un k = 5 en los modelos: Regresión Logística (LR), Potenciación del Gradiente (GB), Bayes Ingenuo (NB), Perceptrón Multicapa (MLP), Árboles Extremadamente Aleatorios (ET), Bosque Aleatorio (RF) y Máquina de Soporte de Vectores (SVM) en el conjunto de datos MusVis balanceado, utilizando selección de características mediante ANOVA y sin escalamiento.

Tabla 4.18: Resultados de F1 en MusVis balanceada con normalización.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | S | F1 | B Acc | DE. (S) | S | F1 | B Acc | DE. (S) |
| GB-Normalizer | 100 % | 100 % | 100 % | 0 | 55.5 % | 55.3 % | 55.5 % | 0.09 |
| MLP-Normalizer | 92.5 % | 92.5 % | 92.5 % | 0.011 | 53.2 % | 52.9 % | 53.2 % | 0.092 |
| ET-Normalizer | 100 % | 100 % | 100 % | 0 | 51.6 % | 51.2 % | 51.6 % | 0.042 |
| NB-Normalizer | 75.8 % | 75.6 % | 75.8 % | 0.013 | 52.0 % | 50.2 % | 52.0 % | 0.05 |
| SVM-Normalizer | 66.7 % | 66.3 % | 66.7 % | 0.046 | 50.6 % | 50.2 % | 50.6 % | 0.118 |
| RF-Normalizer | 100 % | 100 % | 100 % | 0 | 47.9 % | 47.2 % | 47.9 % | 0.077 |
| RL-Normalizer | 60.9 % | 60.8 % | 60.9 % | 0.021 | 45.4 % | 45.0 % | 21.4 % | 0.08 |

Nota: Resultados de Sensibilidad (S), Medida (F1), Exactitud balanceada (B Acc) y Desviación Estándar (DE (S)), utilizando validación cruzada con un k = 5 en los modelos: Potenciación del Gradiente (GB), Perceptrón Multicapa (MLP), Árboles Extremadamente Aleatorios (ET), Bayes Ingenuo (NB), Máquina de Soporte de Vectores (SVM), Bosque Aleatorio (RF), y Regresión Logística (LR) en el conjunto de datos MusVis balanceado, utilizando selección de características mediante ANOVA y con escalamiento L2-Normalizer.

Tabla 4.19: Resultados de F1 en MusVis balanceada con escalamiento Min-Max.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | S | F1 | B Acc | DE. (S) | S | F1 | B Acc | DE. (S) |
| RF-MinMax | 100 % | 100 % | 100 % | 0 | 55.1 % | 54.2 % | 55.1 % | 0.081 |
| MLP-MinMax | 100 % | 100 % | 100 % | 0 | 53.3 % | 0.53 % | 53.3 % | 0.082 |
| RL-MinMax | 83.8 % | 83.8 % | 83.8 % | 0.014 | 51.4 % | 51.2 % | 51.4 % | 0.098 |
| GB-MinMax | 100 % | 100 % | 100 % | 0 | 51.3 % | 51.1 % | 51.3 % | 0.089 |
| NB-MinMax | 66.8 % | 64.0 % | 66.8 % | 0.042 | 55.8 % | 49.8 % | 55.8 % | 0.029 |
| ET-MinMax | 100 % | 100 % | 100 % | 0 | 48.8 % | 48.0 % | 48.8 % | 0.057 |
| SVM-MinMax | 72.2 % | 72.0 % | 72.2 % | 0.026 | 46.6 % | 46.0 % | 46.6 % | 0.048 |

Nota: Resultados de Sensibilidad (S), Medida (F1), Exactitud balanceada (B Acc) y Desviación Estándar (DE (S)), utilizando validación cruzada con un k = 5 en los modelos: Bosque Aleatorio (RF), Perceptrón Multicapa (MLP), Regresión Logística (LR), Potenciación del Gradiente (GB), Bayes Ingenuo (NB), Árboles Extremadamente Aleatorios (ET) y Máquina de Soporte de Vectores (SVM) en el conjunto de datos MusVis balanceado, utilizando selección de características mediante ANOVA y escalamiento Min-Max.

Tabla 4.20: Resultados de F1 en MusVis balanceada con escalamiento estándar.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | S | F1 | B Acc | DE. (S) | S | F1 | B Acc | DE. (S) |
| RL-Estandar | 100 % | 100 % | 100 % | 0 | 56.9 % | 56.5 % | 56.9 % | 0.083 |
| MLP-Estandar | 100 % | 100 % | 100 % | 0 | 56.2 % | 55.8 % | 56.2 % | 0.092 |
| GB-Estandar | 100 % | 100 % | 100 % | 0 | 53.5 % | 53.2 % | 53.5 % | 0.08 |
| RF-Estandar | 100 % | 100 % | 100 % | 0 | 51.5 % | 50.9 % | 51.5 % | 0.077 |
| SVM-Estandar | 95.5 % | 95.5 % | 95.5 % | 0.013 | 50.2 % | 49.7 % | 50.2 % | 0.058 |
| NB-Estandar | 66.7 % | 63.3 % | 66.7 % | 0.058 | 54.5 % | 48.8 % | 54.5 % | 0.072 |
| ET-Estandar | 100 % | 100 % | 100 % | 0 | 49.4 % | 48.1 % | 49.4 % | 0.075 |

Resultados de Sensibilidad (S), Medida (F1), Exactitud balanceada (B Acc) y Desviación Estándar (DE (S)), utilizando validación cruzada con un k = 5 en los modelos: Regresión Logística (LR), Perceptrón Multicapa (MLP), Potenciación del Gradiente (GB), Bosque Aleatorio (RF), Máquina de Soporte de Vectores (SVM), Bayes Ingenuo (NB) y Árboles Extremadamente Aleatorios (ET) en el conjunto de datos MusVis balanceado, utilizando selección de características mediante ANOVA y escalamiento estándar.

Nuevamente, al realizar un análisis solo por la clase con dislexia y considerando la sensibilidad como métrica de evaluación, la tabla de posiciones de desempeño se ven

afectada. Bajo este ranking de posiciones de los modelos, ahora NB-MinMax obtiene un 76.1 % (ver tabla 4.23) y NB-Estandar un 72.9 % (ver tabla 4.24)de sensibilidad, superando al modelo LR que obtuvo un 48.5 % en el experimento uno, (ver Tabla 4.8) y al modelo GB-MinMax que obtuvo el valor más alto con 61.1 % en el experimento dos con selección de características mediante Árboles Extremadamente Aleatorios(ver Tabla 4.15). Sin embargo, se incrementa el error ante la clase sin dislexia, ya que su sensibilidad baja hasta 35.5 % en comparación de los resultados anteriores.

Tabla 4.21: Resultados de sensibilidad por clase en MusVis sin escalamiento.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | Sin dislexia | | Con dislexia | | Sin dislexia | | Con dislexia | |
| | S | F1 | S | F1 | S | F1 | S | F1 |
| RL | 90.9 % | 91.1 % | 91.3 % | 91.1 % | 63.6 % | 65.5 % | 49.4 % | 46.5 % |
| NB | 70.7 % | 70.3 % | 69.8 % | 68.9 % | 61.7 % | 62.9 % | 46.1 % | 43.7 % |
| MLP | 95.0 % | 96.2 % | 97.7 % | 96.5 % | 62.5 % | 63.1 % | 45.4 % | 42.2 % |
| GB | 100 % | 100 % | 100 % | 100 % | 68.7 % | 67.6 % | 41.1 % | 41.6 % |
| SVM | 69.6 % | 63.1 % | 56.7 % | 58.4 % | 60.5 % | 57.1 % | 36.6 % | 31.4 % |
| RF | 100 % | 100 % | 100 % | 100 % | 69.3 % | 65.9 % | 31.4 % | 34.1 % |
| ET | 100 % | 100 % | 100 % | 100 % | 81.5 % | 71.8 % | 23.8 % | 30.2 % |

Nota: Resultados de Sensibilidad (S), F1 utilizando validación cruzada con un k = 5 en los modelos: Regresión Logística (LR), Bayes Ingenuo (NB), Perceptrón Multicapa (MLP), Potenciación del Gradiente (GB), Máquina de Soporte de Vectores (SVM), Bosque Aleatorio (RF) y Árboles Extremadamente Aleatorios (ET) en el conjunto de datos MusVis balanceado por clase, utilizando selección de características mediante ANOVA y sin escalamiento.

Tabla 4.22: Resultados de sensibilidad por clase en MusVis con escalador normalizer.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | Sin dislexia | | Con dislexia | | Sin dislexia | | Con dislexia | |
| | S | F1 | S | F1 | S | F1 | S | F1 |
| NB-Normalizer | 65.7 % | 73.1 % | 85.9 % | 78.0 % | 49.0 % | 55.3 % | 55.0 % | 45.1 % |
| MLP-Normalizer | 92.8 % | 92.5 % | 92.1 % | 92.4 % | 64.2 % | 64.5 % | 42.2 % | 41.3 % |
| GB-Normalizer | 100 % | 100 % | 100 % | 100 % | 70.0 % | 68.1 % | 40.9 % | 42.5 % |
| RL-Normalizer | 60.1 % | 60.5 % | 61.6 % | 61.1 % | 50.9 % | 54.3 % | 40.0 % | 07.5 % |
| SVM-Normalizer | 71.6 % | 67.8 % | 61.7 % | 64.9 % | 61.6 % | 61.2 % | 39.7 % | 39.2 % |
| ET-Normalizer | 100 % | 100 % | 100 % | 100 % | 73.2 % | 68.2 % | 30.1 % | 34.1 % |
| RF-Normalizer | 100 % | 100 % | 100 % | 100 % | 70.0 % | 65.4 % | 25.9 % | 29.0 % |

Nota: Resultados de Sensibilidad (S), F1 utilizando validación cruzada con un k = 5 en los modelos: Bayes Ingenuo (NB), Perceptrón Multicapa (MLP), Potenciación del Gradiente (GB), Regresión Logística (LR), Máquina de Soporte de Vectores (SVM), Árboles Extremadamente Aleatorios (ET) y Bosque Aleatorio (RF) en el conjunto de datos MusVis balanceado por clase, utilizando selección de características mediante ANOVA y escalamiento normalizer.

Tabla 4.23: Resultados de sensibilidad por clase en MusVis con escalador Min-Max.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | Sin dislexia | | Con dislexia | | Sin dislexia | | Con dislexia | |
| | S | F1 | S | F1 | S | F1 | S | F1 |
| NB-MinMax | 40.9 % | 54.3 % | 92.6 % | 73.7 % | 35.5 % | 46.3 % | 76.1 % | 53.4 % |
| SVM-MinMax | 69.4 % | 71.3 % | 75.1 % | 72.7 % | 52.2 % | 55.6 % | 40.9 % | 36.3 % |
| RL-MinMax | 83.9 % | 83.8 % | 83.7 % | 83.8 % | 63.0 % | 63.4 % | 39.8 % | 39.1 % |
| MLP-MinMax | 100 % | 100 % | 100 % | 100 % | 68.7 % | 66.9 % | 37.8 % | 39.2 % |
| GB-MinMax | 100 % | 100 % | 100 % | 100 % | 64.9 % | 64.3 % | 37.7 % | 37.8 % |
| RF-MinMax | 100 % | 100 % | 100 % | 100 % | 77.7 % | 71.4 % | 32.5 % | 37.0 % |
| ET-MinMax | 100 % | 100 % | 100 % | 100 % | 69.4 % | 65.4 % | 28.1 % | 30.6 % |

Nota: Resultados de Sensibilidad (S), F1 utilizando validación cruzada con un k = 5 en los modelos: Bayes Ingenuo (NB), Máquina de Soporte de Vectores (SVM), Regresión Logística (LR), Perceptrón Multicapa (MLP), Potenciación del Gradiente (GB), Bosque Aleatorio (RF) y Árboles Extremadamente Aleatorios (ET) en el conjunto de datos MusVis balanceado por clase, utilizando selección de características mediante ANOVA y escalamiento MinMax.

Tabla 4.24: Resultados de sensibilidad por clase en MusVis con escalador Estandar.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | Sin dislexia | | Con dislexia | | Sin dislexia | | Con dislexia | |
| | S | F1 | S | F1 | S | F1 | S | F1 |
| NB-Estandar | 0.397 | 52.7 % | 93.7 % | 73.9 % | 36.0 % | 45.8 % | 72.9 % | 51.9 % |
| RL-Estandar | 100 % | 100 % | 100 % | 100 % | 64.3 % | 65.9 % | 49.5 % | 47.1 % |
| MLP-Estandar | 100 % | 100 % | 100 % | 100 % | 68.1 % | 67.7 % | 44.3 % | 43.9 % |
| GB-Estandar | 100 % | 100 % | 100 % | 100 % | 68.1 % | 66.7 % | 38.9 % | 39.7 % |
| SVM-Estandar | 97.2 % | 95.6 % | 93.7 % | 95.4 % | 71.3 % | 66.7 % | 29.1 % | 32.8 % |
| RF-Estandar | 100 % | 100 % | 100 % | 100 % | 73.8 % | 68.3 % | 29.1 % | 33.6 % |
| ET-Estandar | 100 % | 100 % | 100 % | 100 % | 75.1 % | 68.0 % | 23.8 % | 28.2 % |

Nota: Resultados de Sensibilidad (S), F1 utilizando validación cruzada con un k = 5 en los modelos: Bayes Ingenuo (NB), Regresión Logística (LR), Perceptrón Multicapa (MLP), Potenciación del Gradiente (GB), Máquina de Soporte de Vectores (SVM), Bosque Aleatorio (RF) y Árboles Extremadamente Aleatorios (ET) en el conjunto de datos MusVis balanceado por clase, utilizando selección de características mediante ANOVA y escalamiento estandar.

## 4.5. Experimento 3: Modelo tipo III con el conjunto de datos MusVis-Entrenamiento balanceado y con ensamble de modelos

Los métodos de basados en ensambles son algoritmos que combinan más de un modelo de aprendizaje computacional, a continuación se presentan dos modelos de ensambles, en primer ensamble combina siete modelos de aprendizaje, Bayes Ingenuo (NB), Regresión Logística (LR), Máquina de Soporte de Vectores (SVM), Árboles Extremadamente Aleatorios (ET), Bosque Aleatorio (RF), Perceptrón Multicapa (MLP) y Potenciación del Gradiente (GB) y un segundo modelo que combina los cuatro mejores algoritmos respecto a los resultados de la experimentación 2, Regresión Logística (LR), Bosque Aleatorio (RF), Perceptrón Multicapa (MLP) y Potenciación del Gradiente (GB). Sé utilizando dos criterios de votación difícil (Hard) y suave(Soft), la diferencia entre ellas es que el

método Hard realiza una decisión tomando en cuenta una votación por mayoría de votos respecto a los algoritmos que conforman el ensamble y la votación soft se basa en la función argmax (las sumas de las probabilidades predichas) y la selección de características ANOVA debido a que presento mejores resultados en la experimentación dos respecto a la clase de interés (infantes con dislexia).

En las tablas 4.25,4.26,4.27 se muestran los resultados de la validación cruzada k-fold con un k=5 para los dos ensambles mencionados anteriormente, considerando dos tipos de votación hard y soft, así como tres técnicas de escalamiento. Si se considera el desempeño general de los modelos, ninguno supera al mejor modelo obtenido en el experimento uno, que es RL con 56.04 % en F1(ver Tabla 4.7). El mejor modelo obtenido es el ensamble 1 con votación soft con un F1 de 53.5 %.

Tabla 4.25: Resultados de ensambles con los datos balanceados, Validación cruzada (5 folders), escalamiento Normalizer y selección de características Anova.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | S | F1 | B Acc | DE (S) | S | F1 | B Acc | DE. (S) |
| Ensamble2H-Normalizer | 89.0 % | 88.9 % | 89.0 % | 0.01 | 53.1 % | 52.6 % | 53.1 % | 0.121 |
| Ensamble2S-Normalizer | 100 % | 100 % | 100 % | 0 | 51.8 % | 51.0 % | 51.8 % | 0.093 |
| Ensamble1H-Normalizer | 84.4 % | 84.3 % | 84.4 % | 0.048 | 49.7 % | 49.6 % | 49.7 % | 0.078 |
| Ensamble1S-Normalizer | 88.5 % | 88.5 % | 88.5 % | 0.033 | 48.7 % | 48.6 % | 48.7 % | 0.04 |

Nota: Resultados de Sensibilidad (S), Medida (F1), Exactitud balanceada (B Acc) y Desviación Estándar (DE (S)) en la validación cruzada con un k = 5 en los ensambles uno y dos en el conjunto de datos MusVis balanceado utilizando votación hard y soft con selección de características mediante ANOVA y escalamiento Normalizer.

Tabla 4.26: Resultados de ensambles con los datos balanceados, Validación cruzada (5 folders), escalamiento MinMax y selección de características Anova.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | S | F1 | B Acc | DE (S) | S | F1 | B Acc | DE. (S) |
| Ensamble1H-MinMax | 99.2 % | 99.2 % | 99.2 % | 0.005 | 50.4 % | 50.0 % | 50.4 % | 0.105 |
| Ensamble2S-MinMax | 100 % | 100 % | 100 % | 0 | 49.6 % | 48.9 % | 49.6 % | 0.08 |
| Ensamble1S-MinMax | 88.5 % | 88.5 % | 88.5 % | 0.033 | 48.7 % | 48.6 % | 48.7 % | 0.04 |
| Ensamble2H-MinMax | 96.4 % | 96.4 % | 96.4 % | 0.011 | 49.0 % | 48.1 % | 49.0 % | 0.073 |

Nota: Resultados de Sensibilidad (S), Medida (F1), Exactitud balanceada (B Acc) y Desviación Estándar (DE (S)) en la validación cruzada con un k = 5 en los ensambles uno y dos en el conjunto de datos MusVis balanceado utilizando votación hard y soft con selección de características mediante ANOVA y escalamiento Normalizer.

Tabla 4.27: Resultados de ensambles con los datos balanceados, Validación cruzada (5 folders), escalamiento Estandar y selección de características Anova.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | S | F1 | B Acc | DE (S) | S | F1 | B Acc | DE. (S) |
| Ensamble1S-Estandar | 100 % | 100 % | 100 % | 0 | 53.8 % | 53.5 % | 53.8 % | 0.092 |
| Ensamble1H-Estándar | 100 % | 100 % | 100 % | 0 | 50.5 % | 50.2 % | 50.5 % | 0.036 |
| Ensamble2H-Estandar | 100 % | 100 % | 100 % | 0 | 50.7 % | 49.4 % | 50.7 % | 0.078 |
| Ensamble2S-Estandar | 100 % | 100 % | 100 % | 0 | 49.4 % | 49.3 % | 49.4 % | 0.066 |

Nota: Resultados de Sensibilidad (S), Medida (F1), Exactitud balanceada (B Acc) y Desviación Estándar (DE (S)) en la validación cruzada con un k = 5 en los ensambles uno y dos en el conjunto de datos MusVis balanceado utilizando votación hard y soft con selección de características mediante ANOVA y escalamiento Normalizer.

Al realizar un análisis solo por la clase con dislexia y considerando la sensibilidad como métrica de evaluación, la tabla de posiciones de desempeño se ven afectada. Bajo este ranking de posiciones de los ensambles, ahora el ensamble 1 con votación hard y escalamiento normalizer obtiene un 37.8 % de sensibilidad (ver tabla 4.28), sin embargo, el mejor ensamble no mejora al modelo LR que obtuvo un 48.5 % en el experimento uno, (ver Tabla 4.8), tampoco al modelo GB-MinMax que obtuvo el valor más alto con 61.1 %

en el experimento dos con selección de características mediante Árboles Extremadamente Aleatorios (ver Tabla 4.15), así como el experimento 2 con selección de características ANOVA con el modelo NB-MinMax que obtuvo una sensibilidad de 76.1 % (ver Tabla 4.23).

Tabla 4.28: Resultados de ensambles con los datos balanceados, Validación cruzada (5 folders), escalamiento Normalizer y selección de características Anova por clase.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | Sin dislexia | | Con dislexia | | Sin dislexia | | Con dislexia | |
| Ensamble1H-Normalizer | 83.3 % | 84.1 % | 85.5 % | 84.6 % | 61.7 % | 62.1 % | 37.8 % | 37.2 % |
| Ensamble1S-Normalizer | 86.4 % | 88.3 % | 90.5 % | 88.7 % | 630. % | 62.3 % | 34.5 % | 34.9 % |
| Ensamble2S-Normalizer | 100 % | 100 % | 100 % | 100 % | 70.0 % | 67.0 % | 33.5 % | 35.1 % |
| Ensamble2H-Normalizer | 100 % | 90.1 % | 78.1 % | 87.7 % | 73.8 % | 69.0 % | 32.5 % | 36.3 % |

Nota: Resultados de Sensibilidad (S) y F1 por clase en la validación cruzada con un k = 5 en los ensambles uno y dos en el conjunto de datos MusVis balanceado utilizando votación hard y soft con selección de características mediante ANOVA y escalamiento L2-Normalizer.

Tabla 4.29: Resultados de ensambles con los datos balanceados, Validación cruzada (5 folders), escalamiento Min-Max y selección de características Anova por clase.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | Sin dislexia | | Con dislexia | | Sin dislexia | | Con dislexia | |
| Ensamble1H-MinMax | 99.8 % | 99.2 % | 98.7 % | 99.2 % | 66.2 % | 64.6 % | 34.6 % | 35.4 % |
| Ensamble1S-MinMax | 86.4 % | 88.3 % | 90.5 % | 88.7 % | 63.0 % | 62.3 % | 34.5 % | 34.9 % |
| Ensamble2S-MinMax | 100 % | 100 % | 100 % | 100 % | 65.6 % | 63.9 % | 33.6 % | 33.8 % |
| Ensamble2H-MinMax | 100 % | 96.5 % | 92.8 % | 96.2 % | 73.1 % | 67.1 % | 24.9 % | 29.2 % |

Nota: Resultados de Sensibilidad (S) y F1 por clase en la validación cruzada con un k = 5 en los ensambles uno y dos en el conjunto de datos MusVis balanceado utilizando votación hard y soft con selección de características mediante ANOVA y escalamiento MinMax.

Tabla 4.30: Resultados de ensambles con los datos balanceados, Validación cruzada (5 folders), escalamiento Estandar y selección de características Anova por clase.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | Sin dislexia | | Con dislexia | | Sin dislexia | | Con dislexia | |
| Ensamble1S-Estandar | 100 % | 100 % | 100 % | 100 % | 71.9 % | 68.5 % | 35.75 | 38.5 % |
| Ensamble2S-Estandar | 100 % | 100 % | 100 % | 100 % | 65.5 % | 63.9 % | 33.3 % | 34.6 % |
| Ensamble1H-Estandar | 100 % | 100 % | 100 % | 100 % | 68.7 % | 65.8 % | 32.3 % | 34.7 % |
| Ensamble2H-Estandar | 100 % | 100 % | 100 % | 100 % | 77.6 % | 69.7 % | 23.8 % | 29.1 % |

Nota: Resultados de Sensibilidad (S) y F1 por clase en la validación cruzada con un k = 5 en los ensambles uno y dos en el conjunto de datos MusVis balanceado utilizando votación hard y soft con selección de características mediante ANOVA y escalamiento Estandar.

## 4.6. Evaluación de resultados en el conjunto de datos MusVis-Prueba

En esta sección, se presentan los resultados de la evaluación de los modelos entrenados en los tres experimentos anteriores sobre el conjunto de datos MusVis-Prueba, con la finalidad de comparar el desempeño contra el trabajo realizado en (Rauschenberger, 2022).

### 4.6.1. Evaluación de resultados del modelo tipo I

En la Figura 4.1 se muestra la comparación del desempeño de los modelos obtenidos en el experimento 1 (sección 4.3.1) mediante la métrica de sensibilidad obtenida para ambas clases. Si se considera que el objetivo es mejorar la identificación de los casos con dislexia, en la Figura 4.1 se muestra como baseline el trabajo de (Rauschenberger, 2022), el cual es un modelo RL entrenado con un conjunto de datos desbalanceado y una separación de datos en proporción de 80:20 para los datos de entrenamiento y prueba, además no se utilizó selección de características durante su aprendizaje. En los resultados

de evaluación, solo los modelos RL, GB logran superar la capacidad de recuperación de los niños con dislexia con respecto al baseline. En el caso de GB se observa que el rendimiento en la sensibilidad de los niños sin dislexia baja, se puede interpretar como la clasificación incorrecta de varios niños sin dislexia han sido clasificados de forma incorrecta (como verdaderos negativos). Pero en el caso RL el desempeño es mucho menor para la clase con dislexia respecto al modelo GB.

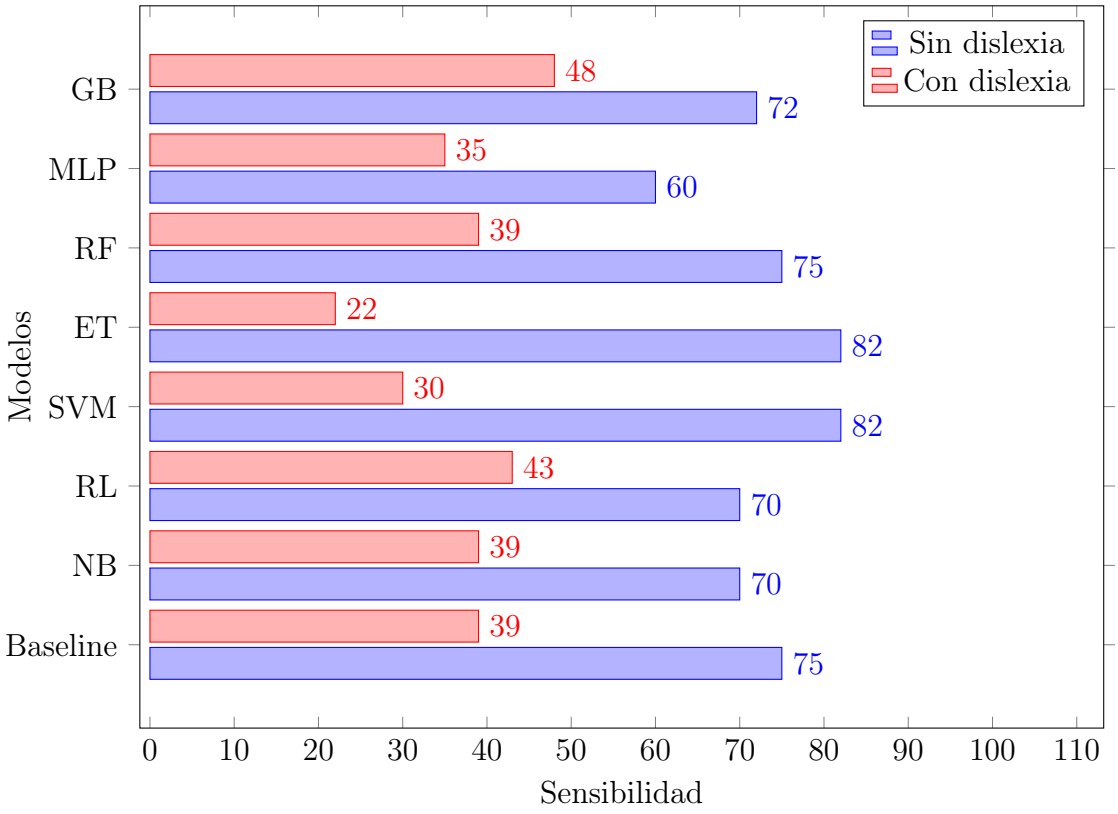


Figura 4.1: Resultados de la sensibilidad de los modelos obtenidos en el Experimento 1.

*Nota. Se muestran los resultados de sensibilidad para la clase sin dislexia (azul) y dislexia (rojo) mediante la métrica de sensibilidad para los modelos: Potenciación del Gradiente (GB), Perceptrón Multicapa (MLP), Bosque Aleatorio (RF), Árboles Extremadamente Aleatorios (ET), Máquina de Soporte de Vectores (SVM), Regresión Logística (LR) y Bayes Ingenuo (NB) en el conjunto de datos MusVis-Prueba sin balanceo de datos.*

### 4.6.2. Evaluación de resultados del modelo tipo II

Al evaluar el desempeño de los modelos obtenidos en la sección 4.4 de los experimentos 4.9, 4.10, 4.11, 4.12, se muestra el desempeño sobre el conjunto de datos MusVis-Prueba desbalanceado. En las Figuras 4.2, 4.3, 4.4, 4.5, se muestra mediante la métrica de sensibilidad que los modelos obtuvieron con una selección de características de árboles extremadamente aleatorios. Los modelos NB-Normalizer, RL-Estandar, RL-Normalizer, RL, SVM-Normalizer, SVM, RF-MinMax, GB-Estandar, GB-MinMax, GB-Normalizer y GB superan el baseline en la clase de interés(clase con dislexia).

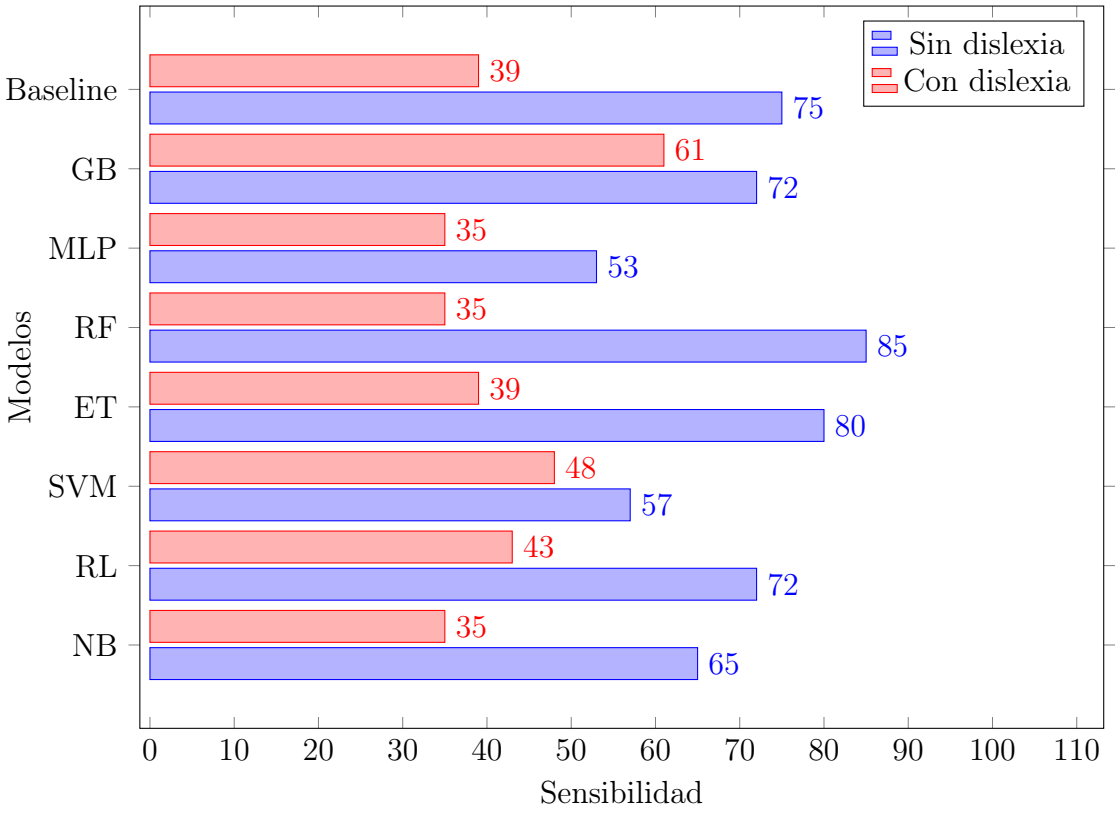


Figura 4.2: Sensibilidad del modelo 2.1 sin escalamiento

*Nota. Se muestran los resultados de sensibilidad para la clase sin dislexia (azul) y dislexia (rojo) mediante la métrica de sensibilidad para los modelos: Potenciación del Gradiente (GB), Perceptrón Multicapa (MLP), Bosque Aleatorio (RF), Árboles Extremadamente Aleatorios (ET), Máquina de Soporte de Vectores (SVM), Regresión Logística (LR) y Bayes Ingenuo (NB) en el conjunto de datos MusVis-Prueba sin balanceo de datos, selección de características árboles extremadamente aleatorios y sin escalamiento.*

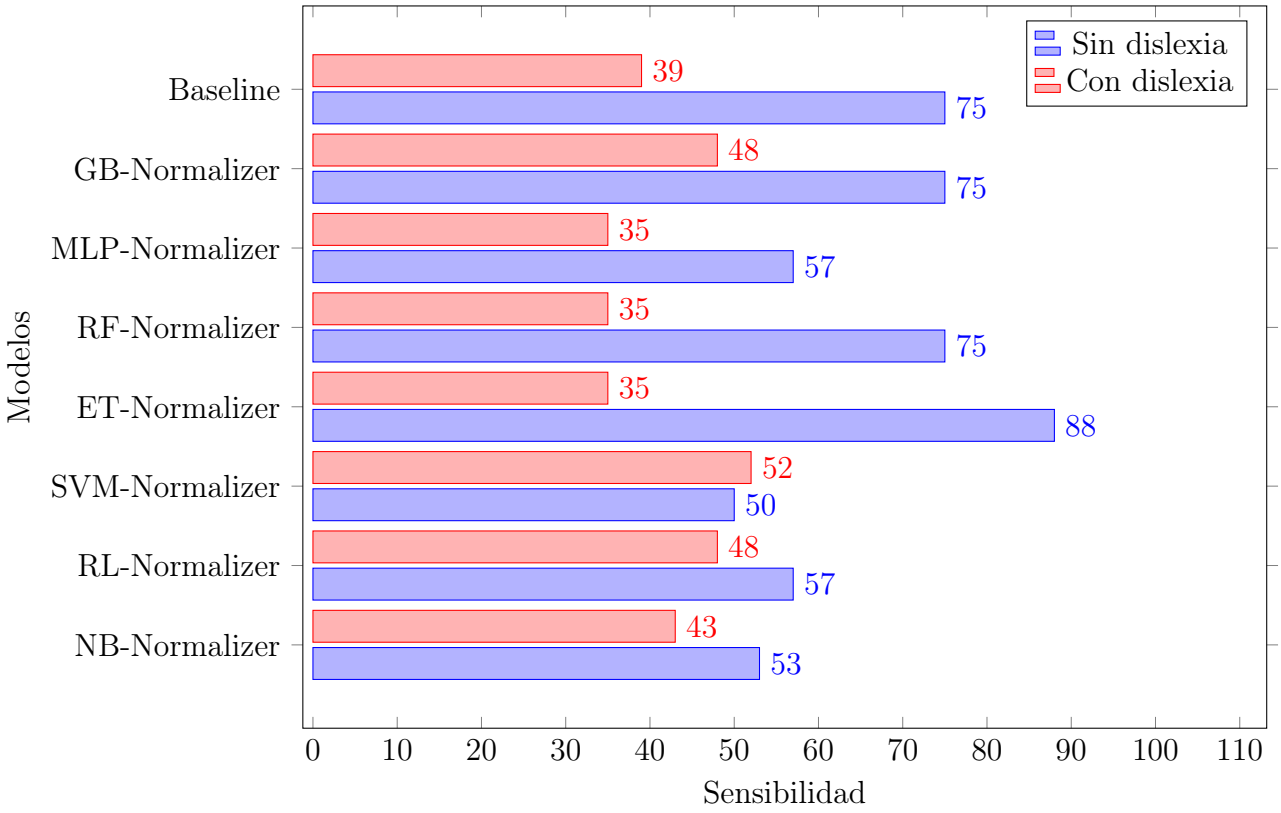


Figura 4.3: Sensibilidad del modelo 2.1 con escalamiento normalizer

*Nota. Se muestran los resultados de sensibilidad para la clase sin dislexia (azul) y dislexia (rojo) mediante la métrica de sensibilidad para los modelos: Potenciación del Gradiente (GB), Perceptrón Multicapa (MLP), Bosque Aleatorio (RF), Árboles Extremadamente Aleatorios (ET), Máquina de Soporte de Vectores (SVM), Regresión Logística (LR) y Bayes Ingenuo (NB) en el conjunto de datos MusVis-Prueba sin balanceo de datos, selección de características árboles extremadamente aleatorios y escalamiento L2-Normalizer.*

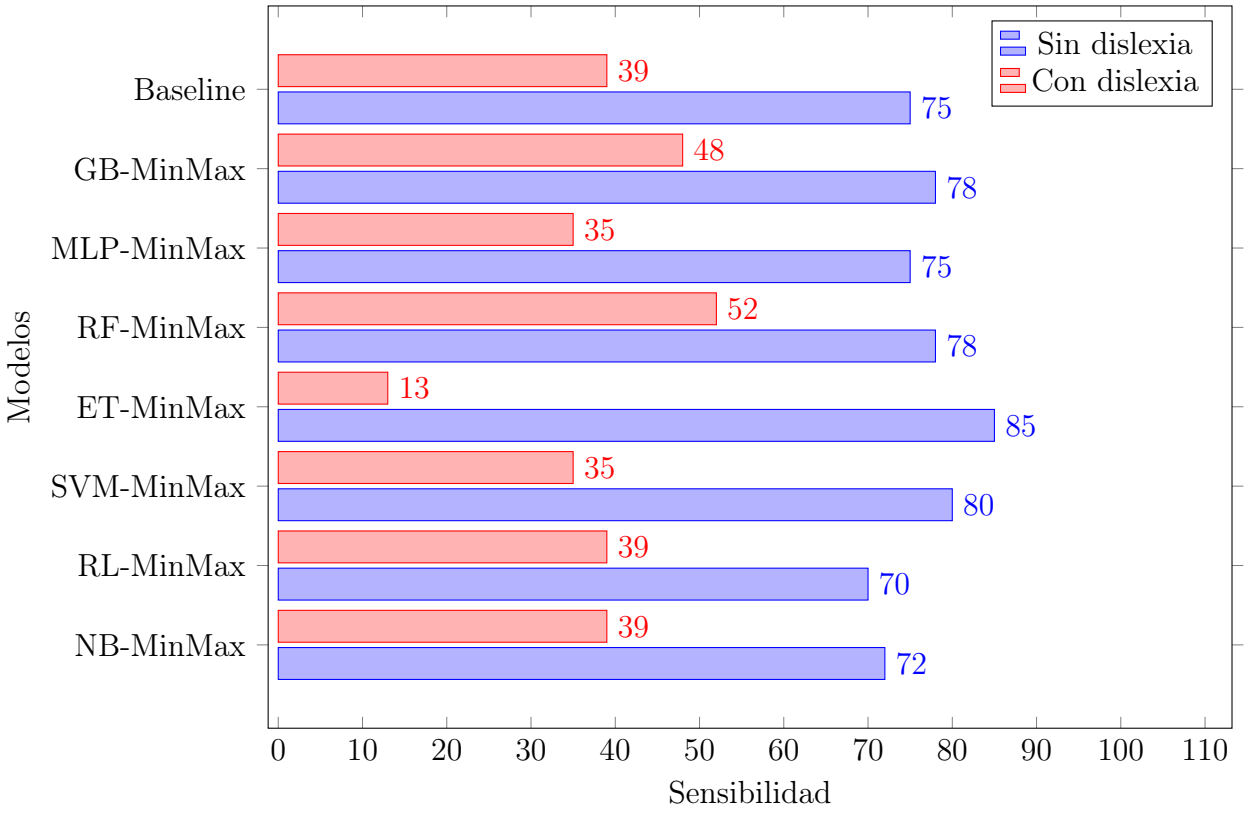


Figura 4.4: Sensibilidad del modelo 2.1 con escalamiento min-max

*Nota. Se muestran los resultados de sensibilidad para la clase sin dislexia (azul) y dislexia (rojo) mediante la métrica de sensibilidad para los modelos: Potenciación del Gradiente (GB), Perceptrón Multicapa (MLP), Bosque Aleatorio (RF), Árboles Extremadamente Aleatorios (ET), Máquina de Soporte de Vectores (SVM), Regresión Logística (LR) y Bayes Ingenuo (NB) en el conjunto de datos MusVis-Prueba sin balanceo de datos, selección de características árboles extremadamente aleatorios y escalamiento MinMax.*

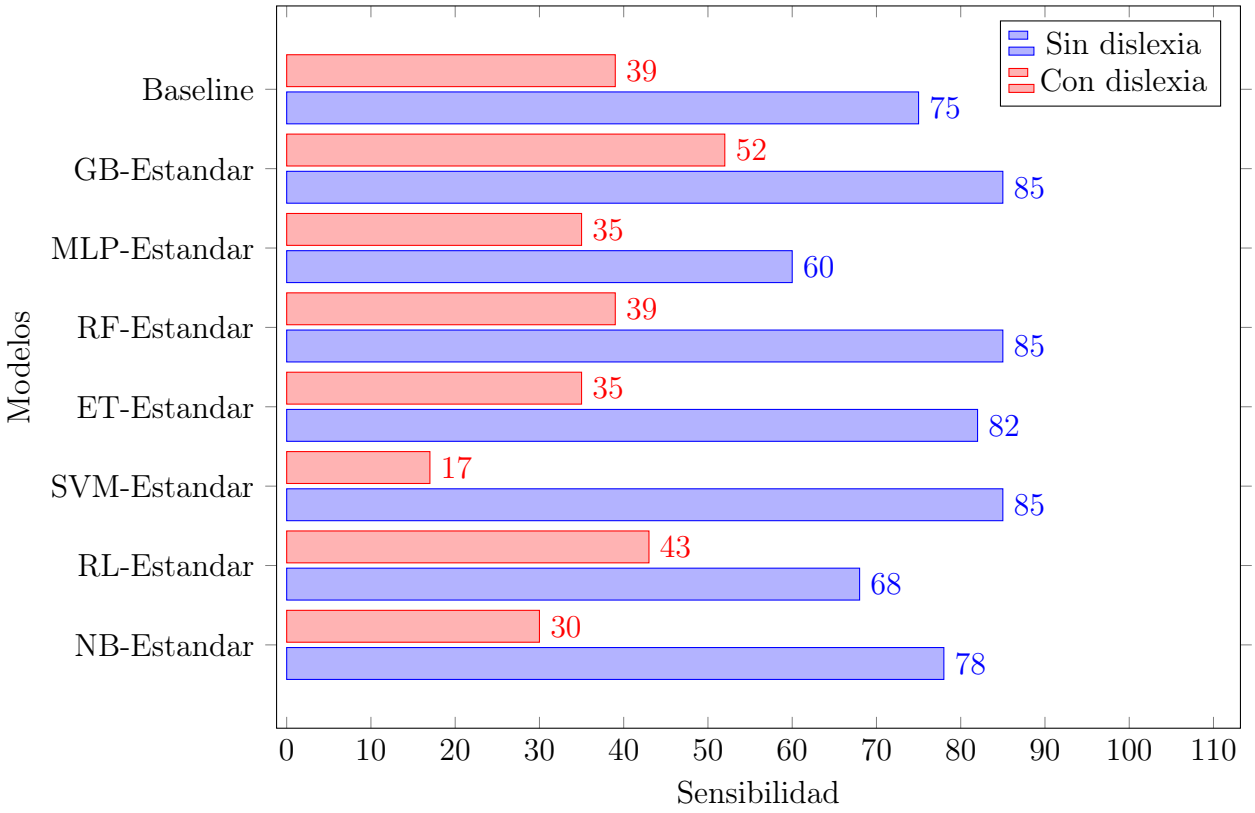


Figura 4.5: Sensibilidad del modelo 2.1 con escalamiento estandar

*Nota. Se muestran los resultados de sensibilidad para la clase sin dislexia (azul) y dislexia (rojo) mediante la métrica de sensibilidad para los modelos: Potenciación del Gradiente (GB), Perceptrón Multicapa (MLP), Bosque Aleatorio (RF), Árboles Extremadamente Aleatorios (ET), Máquina de Soporte de Vectores (SVM), Regresión Logística (LR) y Bayes Ingenuo (NB) en el conjunto de datos MusVis-Prueba sin balanceo de datos, selección de características árboles extremadamente aleatorios y escalamiento estandar.*

Al evaluar el desempeño de los modelos obtenidos en la sección 4.4 de los experimentos 4.21, 4.22, 4.23,4.20, se muestra el desempeño sobre el conjunto de datos MusVis-Prueba desbalanceado. En las figuras 4.6, 4.7,4.8,4.9 se muestra mediante la métrica de sensibilidad que los modelos obtuvieron con una selección de características de ANOVA. Los modelos NB-Estandar, NB-MinMax, NB-Normalizer, NB, RL-Estandar, RL-MinMax, RL-Normalizer, SVM-MinMax, SVM-Normalizer, RF-Estandar, RF, MLP, GB-Normalizer, GB superan el baseline en la clase de interés(clase con dislexia).

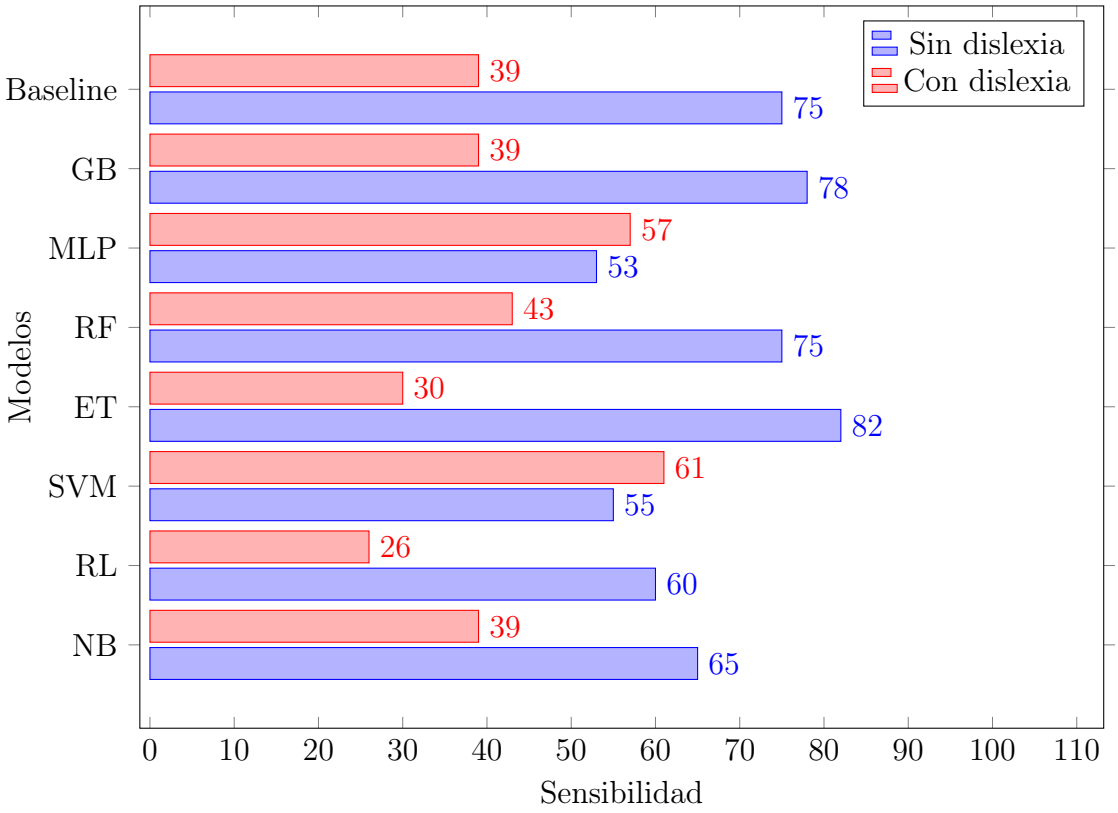


Figura 4.6: Sensibilidad del modelo 2.2 sin escalamiento

*Nota. Se muestran los resultados de sensibilidad para la clase sin dislexia (azul) y dislexia (rojo) mediante la métrica de sensibilidad para los modelos: Potenciación del Gradiente (GB), Perceptrón Multicapa (MLP), Bosque Aleatorio (RF), Árboles Extremadamente Aleatorios (ET), Máquina de Soporte de Vectores (SVM), Regresión Logística (LR) y Bayes Ingenuo (NB), en el conjunto de datos MusVis-Prueba sin balanceo de datos, selección de características ANOVA y sin escalamiento*

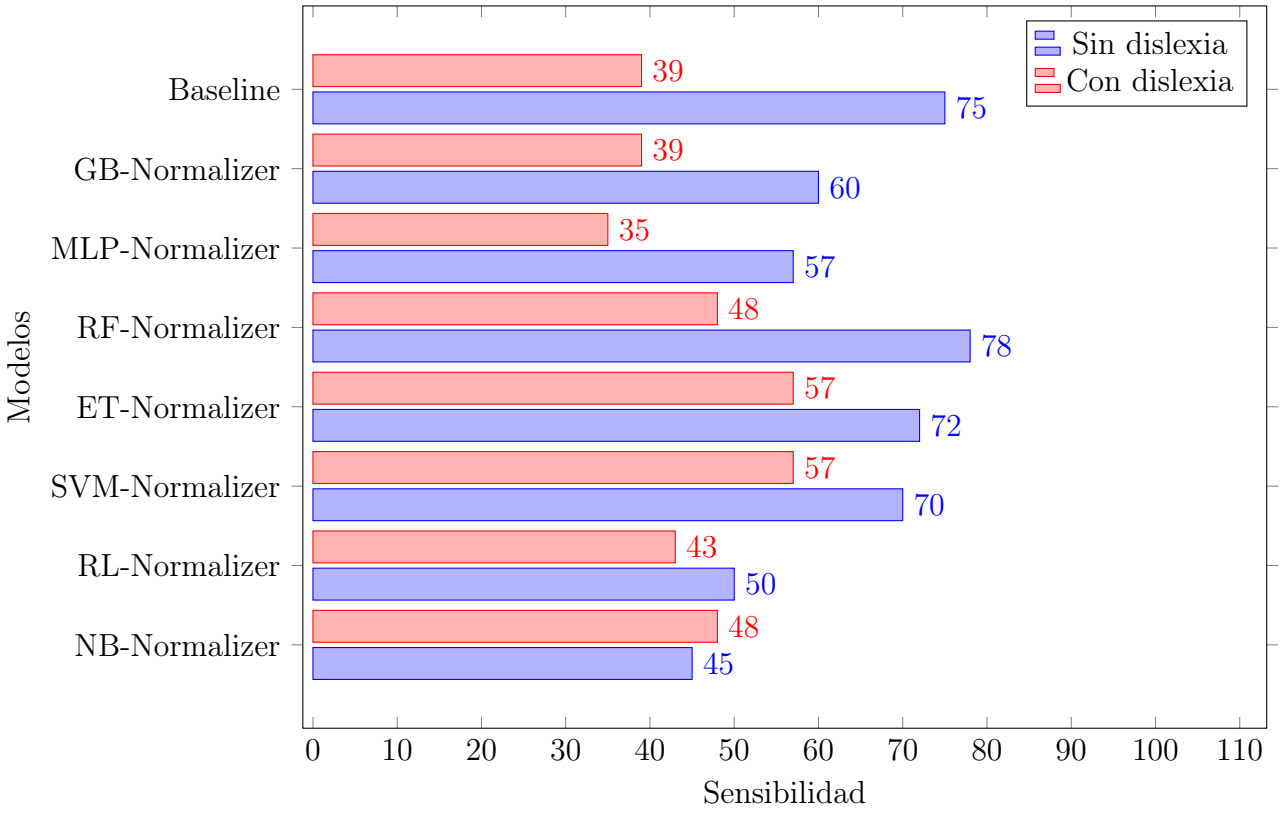


Figura 4.7: Sensibilidad del modelo 2.2 con escalamiento normalizer

*Nota. Se muestran los resultados de sensibilidad para la clase sin dislexia (azul) y dislexia (rojo) mediante la métrica de sensibilidad para los modelos: Potenciación del Gradiente (GB), Perceptrón Multicapa (MLP), Bosque Aleatorio (RF), Árboles Extremadamente Aleatorios (ET), Máquina de Soporte de Vectores (SVM), Regresión Logística (LR) y Bayes Ingenuo (NB), en el conjunto de datos MusVis-Prueba sin balanceo de datos, selección de características ANOVA y escalamiento L2-Normalizer*

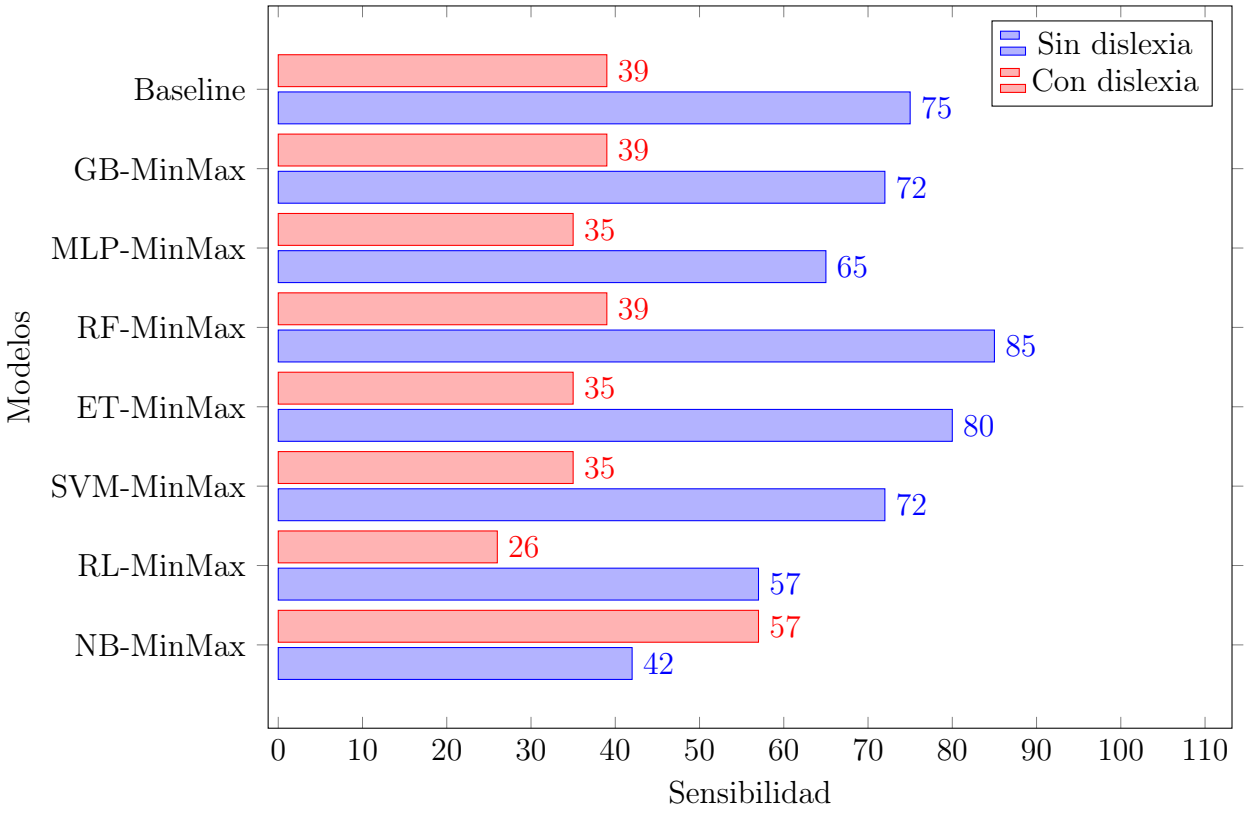


Figura 4.8: Sensibilidad del modelo 2.2 con escalamiento min-max

*Nota. Se muestran los resultados de sensibilidad para la clase sin dislexia (azul) y dislexia (rojo) mediante la métrica de sensibilidad para los modelos: Potenciación del Gradiente (GB), Perceptrón Multicapa (MLP), Bosque Aleatorio (RF), Árboles Extremadamente Aleatorios (ET), Máquina de Soporte de Vectores (SVM), Regresión Logística (LR) y Bayes Ingenuo (NB), en el conjunto de datos MusVis-Prueba sin balanceo de datos, selección de características ANOVA y escalamiento Min-Max.*

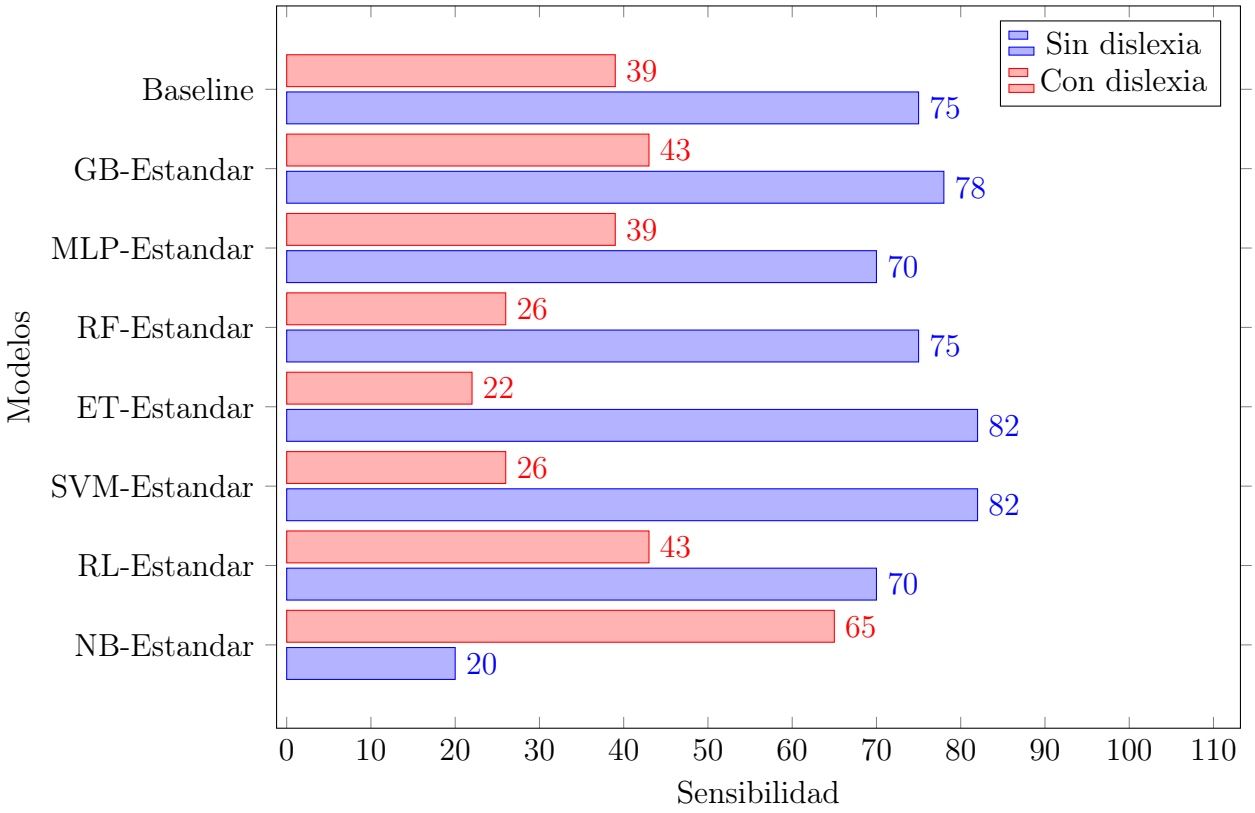


Figura 4.9: Sensibilidad del modelo 2.2 con escalamiento estandar

*Nota. Se muestran los resultados de sensibilidad para la clase sin dislexia (azul) y dislexia (rojo) mediante la métrica de sensibilidad para los modelos: Potenciación del Gradiente (GB), Perceptrón Multicapa (MLP), Bosque Aleatorio (RF), Árboles Extremadamente Aleatorios (ET), Máquina de Soporte de Vectores (SVM), Regresión Logística (LR) y Bayes Ingenuo (NB), en el conjunto de datos MusVis-Prueba sin balanceo de datos, selección de características ANOVA y escalamiento Estandar.*

### 4.6.3. Evaluación de resultados del modelo tipo III

Al evaluar el desempeño de los modelos obtenidos en la sección 4.4 de los experimentos 4.28, 4.29, 4.30, se muestra el desempeño sobre el conjunto de datos MusVis-Prueba desbalanceado. En las figuras 4.10, 4.11, 4.12 se muestra mediante la métrica de sensibilidad que los modelos obtuvieron con una selección de características de ANOVA y ensamble de modelos. Los modelos Ensamble-1-H-Normalizer y Ensamble-1-S-Normalizer superan

el baseline en la clase de interés(clase con dislexia).

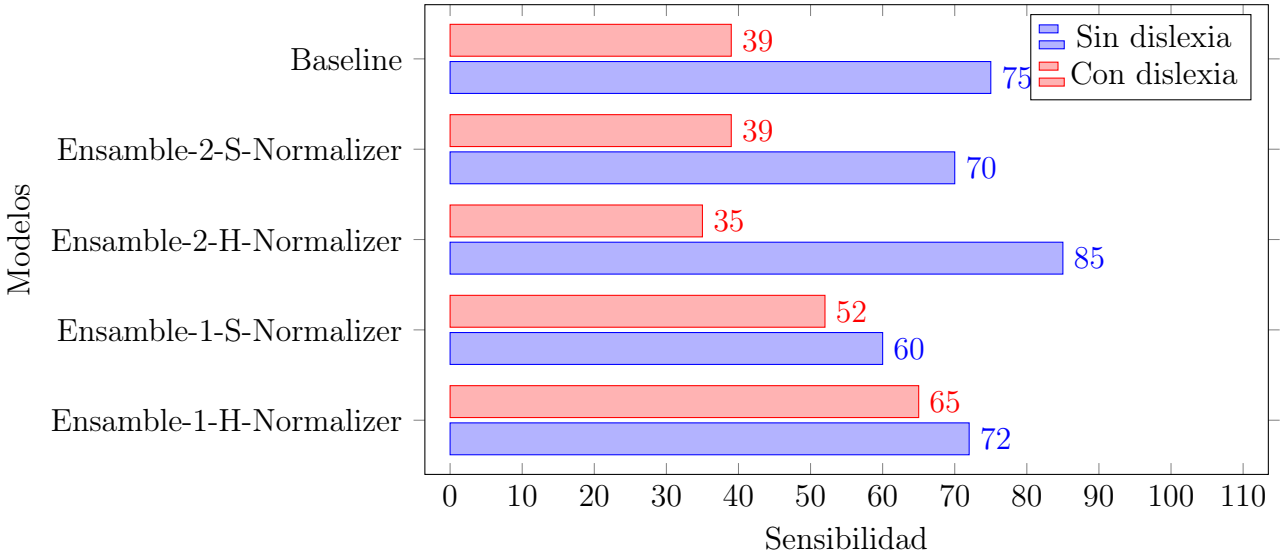


Figura 4.10: Sensibilidad del modelo 3 con escalamiento L2-Normalizer.

*Nota. Se muestran los resultados de sensibilidad para la clase sin dislexia (azul) y dislexia (rojo) mediante la métrica de sensibilidad para los ensambles de modelos en el conjunto de datos MusVis-Prueba sin balanceo de datos, selección de características ANOVA, así como escalamiento L2-Normalizer.*

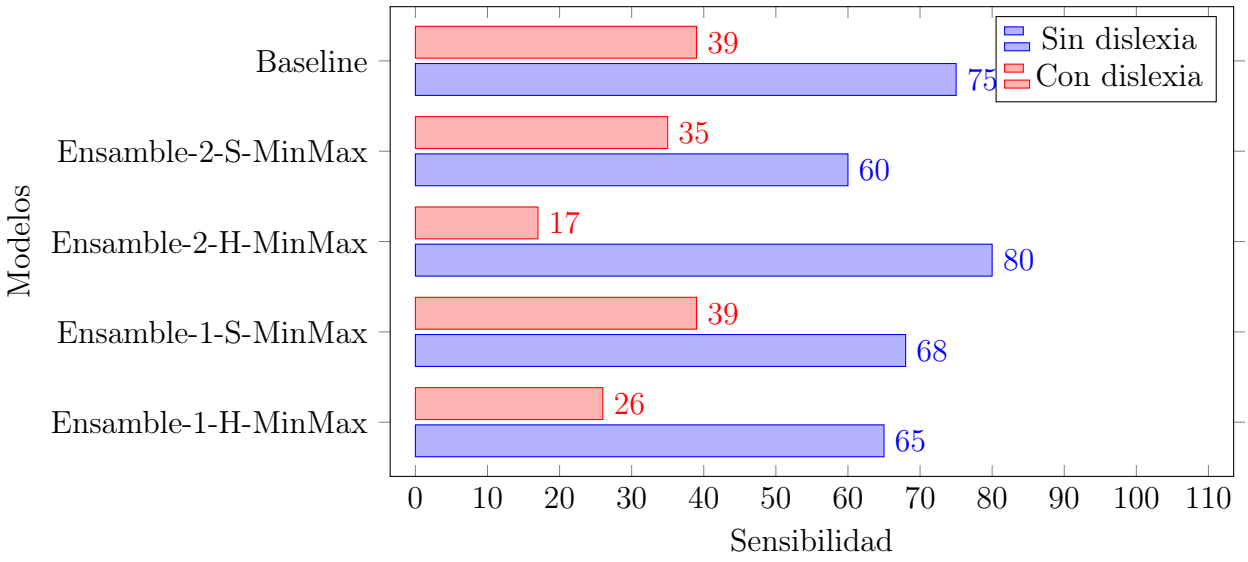


Figura 4.11: Sensibilidad del modelo 3 con escalamiento MinMax.

*Nota. Se muestran los resultados de sensibilidad para la clase sin dislexia (azul) y dislexia (rojo) mediante la métrica de sensibilidad para los ensambles de modelos en el*

*conjunto de datos MusVis-Prueba sin balanceo de datos, selección de características ANOVA, así como escalamiento MinMax.*

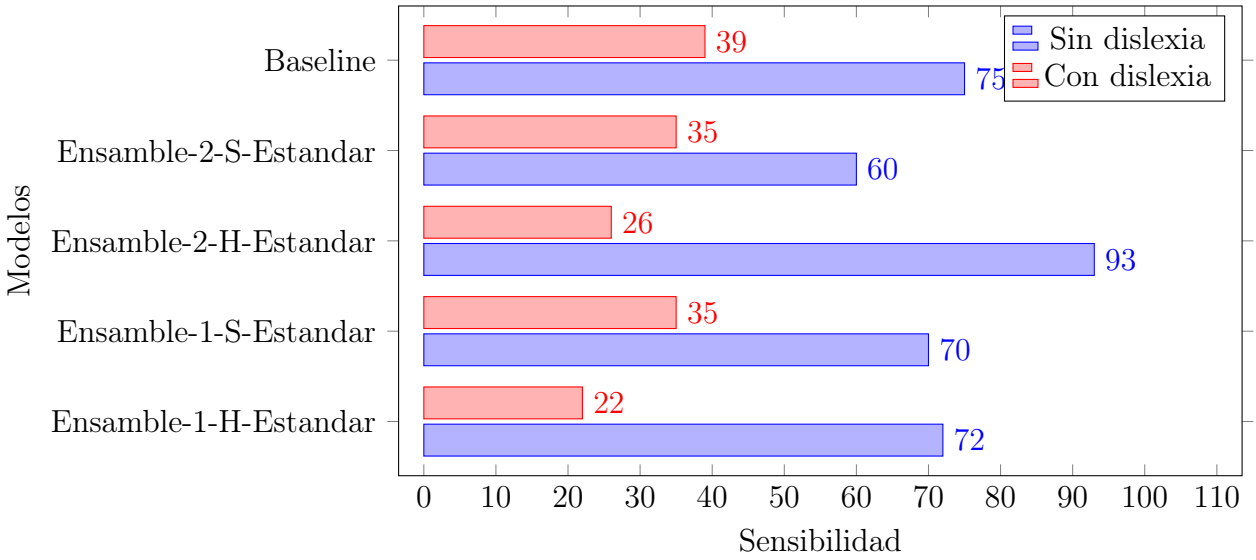


Figura 4.12: Sensibilidad del modelo 3 con escalamiento estandar.

*Nota. Se muestran los resultados de sensibilidad para la clase sin dislexia (azul) y dislexia (rojo) mediante la métrica de sensibilidad para los ensambles de modelos en el conjunto de datos MusVis-Prueba sin balanceo de datos, selección de características ANOVA, así como escalamiento estandar.*

## 4.7. Busqueda en malla

En esta sección, se presentan los resultados de la evaluación de los mojores modelos entrenados en los tres experimentos anteriores sobre el conjunto de datos MusVis-Prueba y una busqueda en malla , con la finalidad de comparar el desempeño contra el trabajo realizado en (Rauschenberger, 2022).

### 4.7.1. Modelo tipo I

En la figura 4.1 se muestra que el mejor modelo tomando en cuenta la clase con dislexia es GB, por lo cual, se realizó un ajuste de hiperparámetros mediante búsqueda en malla(grid search), a continuación se muestra en la figura 4.13, los resultados de búsqueda en malla en el clasificador GB.

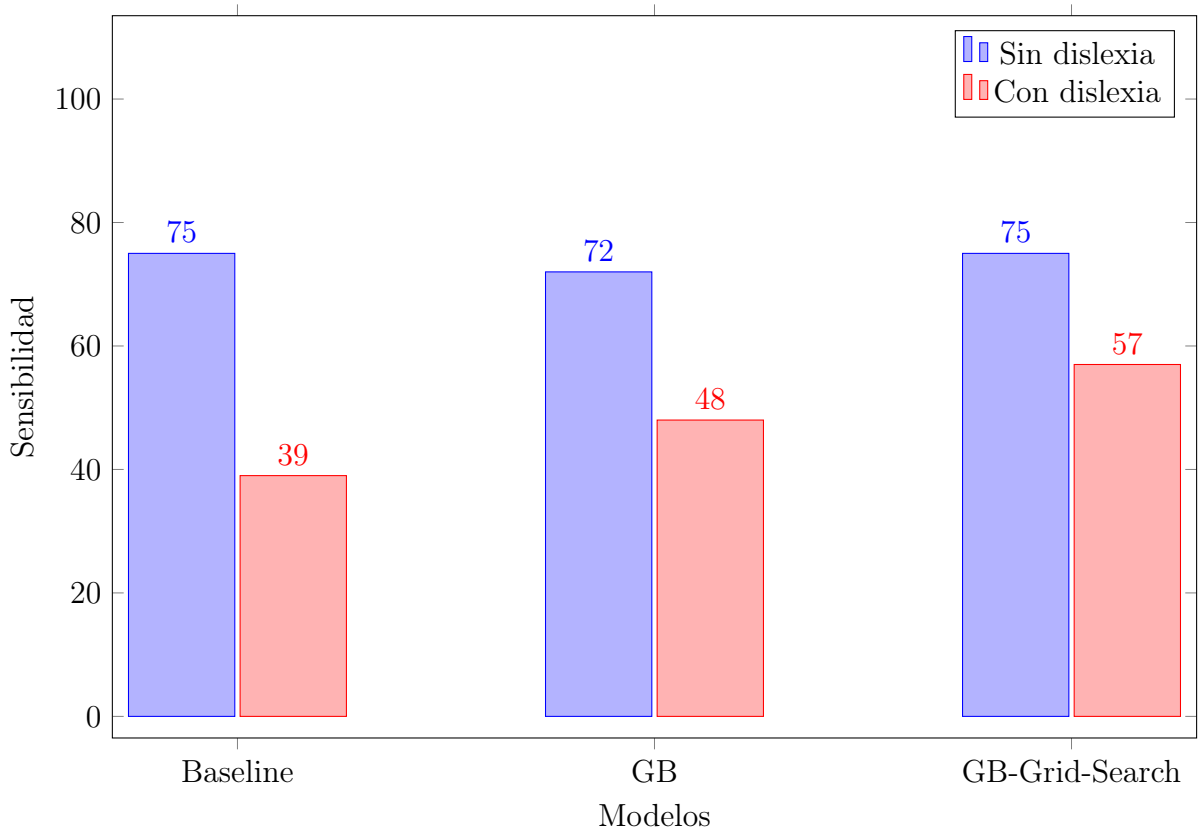


Figura 4.13: Resultados de la sensibilidad del los modelos obtenidos en el Experimento 1 con búsqueda en malla

*Nota. Se muestran los resultados de sensibilidad para la clase sin dislexia (azul) y dislexia (rojo) mediante la métrica de sensibilidad para los modelos: Potenciación del Gradiente (GB) sin y con búsqueda en malla en el conjunto de datos MusVis-Prueba sin balanceo de datos.*

### 4.7.2. Modelo tipo II

En la figura 4.2 se muestra que el mejor modelo tomando en cuenta la clase con dislexia es GB, por lo cual, se realizó un ajuste de hiperparámetros mediante búsqueda en malla(grid search), a continuación se muestra en la figura 4.14, los resultados de búsqueda en malla en el clasificador GB.

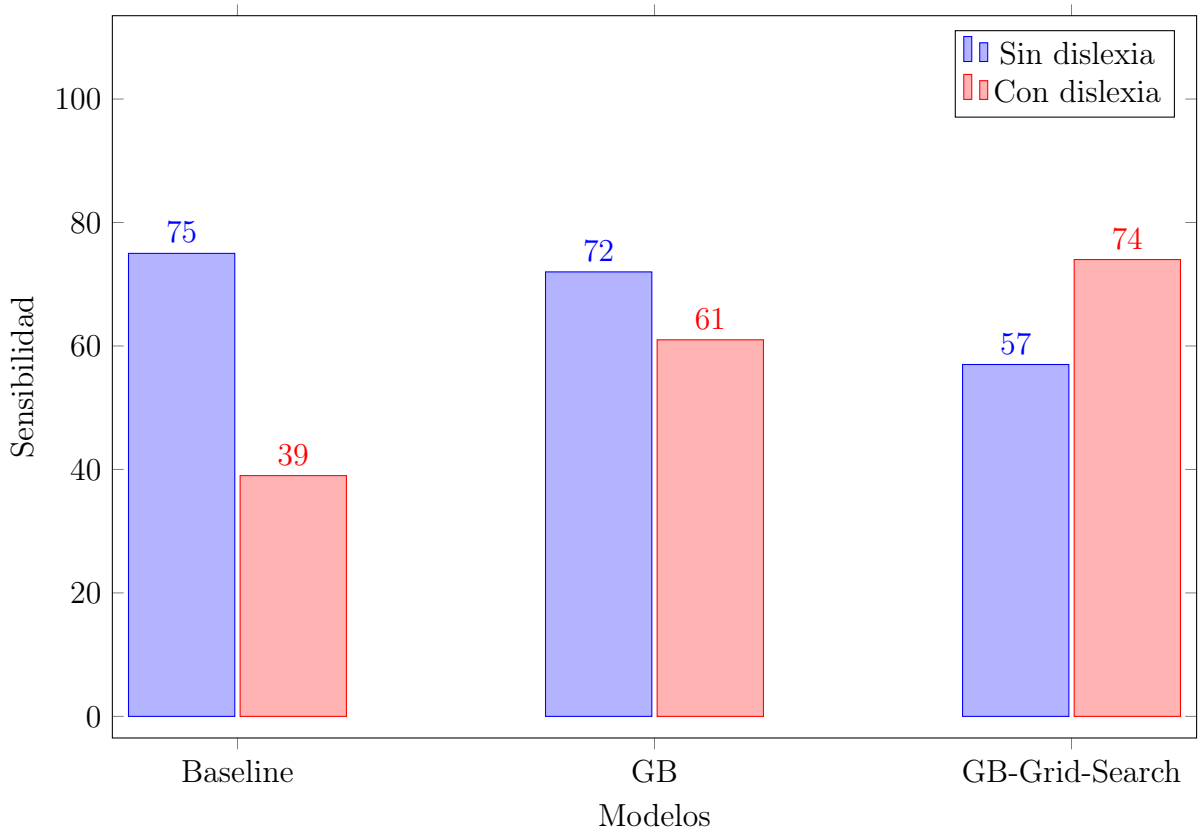


Figura 4.14: Resultados de la sensibilidad del los modelos obtenidos en el Experimento 2.1 con búsqueda en malla

*Nota. Se muestran los resultados de sensibilidad para la clase sin dislexia (azul) y dislexia (rojo) mediante la métrica de sensibilidad para los modelos: Potenciación del Gradiente (GB) sin y con búsqueda en malla en el conjunto de datos MusVis-Prueba sin balanceo de datos.*

En la figura 4.9 se muestra que el mejor modelo, tomando en cuenta la clase con dislexia es NB-Estandar, por lo cual, se realizó un ajuste de hiperparámetros mediante búsqueda en malla(grid search), a continuación se muestra en la figura 4.15, los resultados

de búsqueda en malla en el clasificador NB con escalamiento estandar.

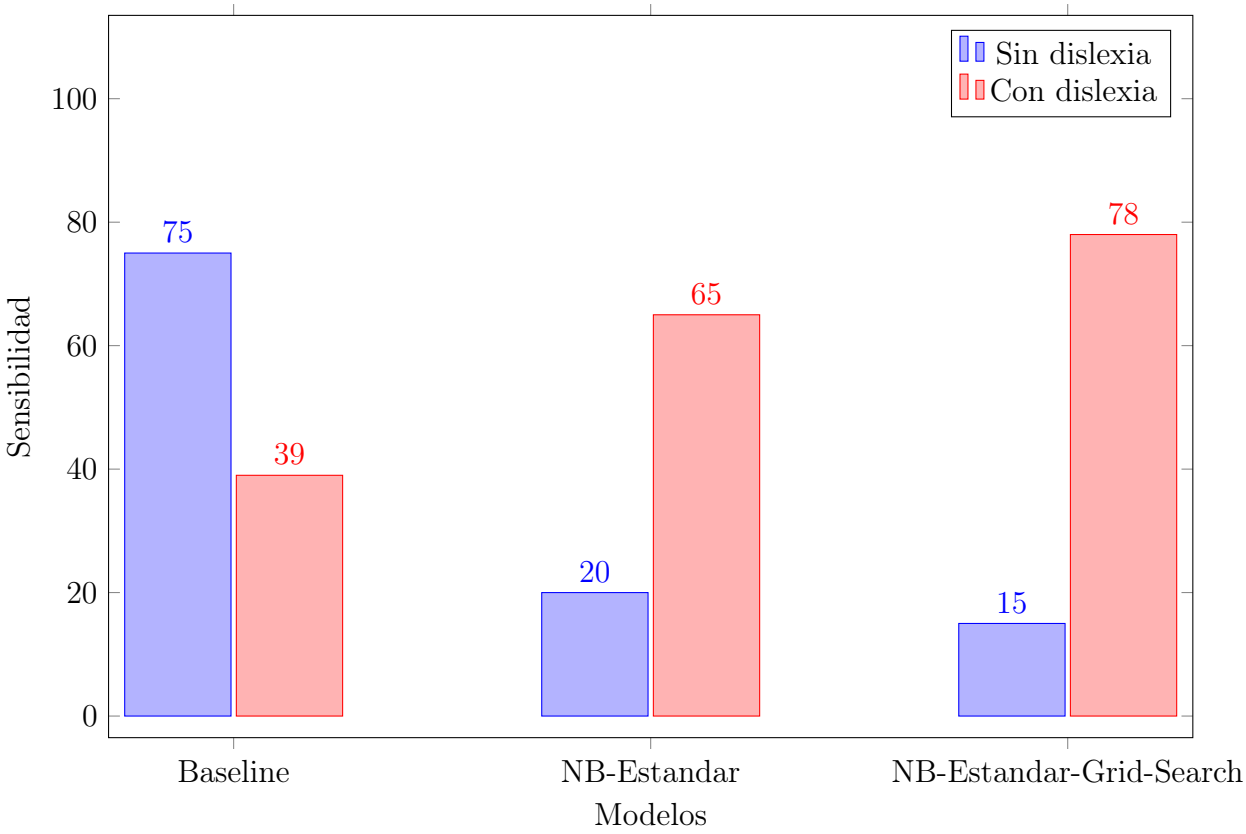


Figura 4.15: Resultados de la sensibilidad del los modelos obtenidos en el Experimento 2.2 con búsqueda en malla

*Nota. Se muestran los resultados de sensibilidad para la clase sin dislexia (azul) y dislexia (rojo) mediante la métrica de sensibilidad para los modelos: Naive Bayes (NB) con escalamiento estandar, sin búsqueda en malla y con búsqueda en malla en el conjunto de datos MusVis-Prueba sin balanceo de datos.*

### 4.7.3. Modelo tipo III

En la figura 4.10 se muestra que el mejor modelo tomando en cuenta la clase con dislexia es Ensamble-1-H-Normalizer, por lo cual, se realizó un ajuste de hiperparámetros mediante búsqueda en malla(grid search), a continuación se muestra en la figura 4.16, los resultados de búsqueda en malla en el ensamble 1 con votación hard, selección de características ANOVA y escalamiento L2-Normalizer.

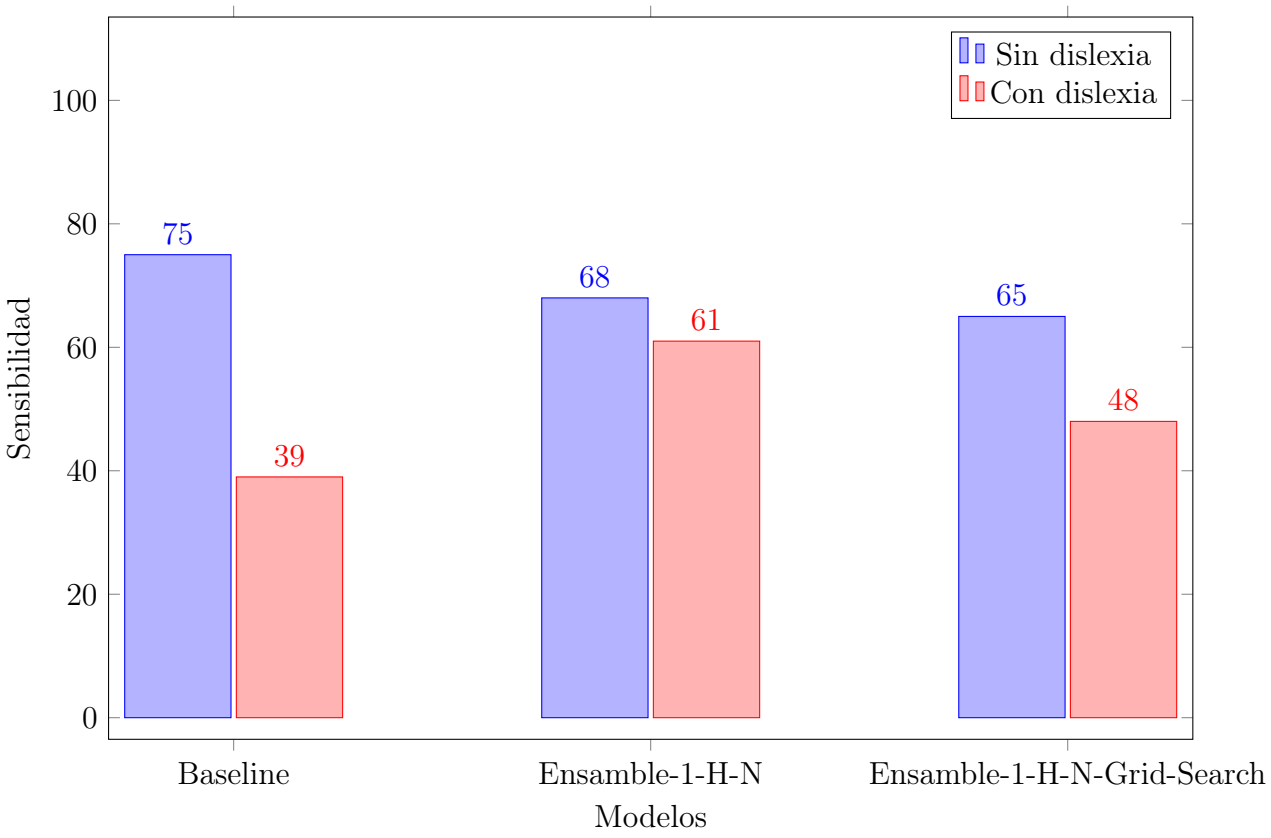


Figura 4.16: Resultados de la sensibilidad del los modelos obtenidos en el Experimento 3 con búsqueda en malla

*Nota. Se muestran los resultados de sensibilidad para la clase sin dislexia (azul) y dislexia (rojo) mediante la métrica de sensibilidad para el ensamble 1, compuesto por los modelos: Bayes Ingenuo (NB), Regresión Logística (LR), Máquina de Soporte de Vectores (SVM), Árboles Extremadamente Aleatorios (ET), Bosque Aleatorio (RF), Perceptrón Multicapa (MLP) y Potenciación del Gradiente (GB) con escalamiento L2-normalizer, sin búsqueda en malla y con búsqueda en malla en el conjunto de datos MusVis-Prueba sin balanceo de datos.*

# Capítulo 5

# Conclusiones

La dislexia es un trastorno del aprendizaje caracterizado por la dificultad de reconocimiento preciso o fluido de palabras. El diagnóstico ocurre generalmente en las primeras etapas de la vida, cuando los niños aprenden a leer. La falta de un diagnóstico genera problemas sociales y psicológicos para el individuo, por lo tanto, es necesario realizar la detección a tiempo. En los trabajos previos a esta investigación, se plantean distintos enfoques sobre el uso de algoritmos de aprendizaje computacional que varían de los datos utilizados para caracterizar a una persona diagnosticada con dislexia de aquella que no presenta este trastorno. Existe un enfoque novedoso, basado en datos obtenidos mediante la interacción en un juego Web, donde se miden el tiempo de respuesta y errores de los participantes, donde se estudia la correlación entre estos valores y el diagnóstico previo de dislexia del participante.

En este proyecto de tesis se muestra el uso de distintos modelos de aprendizaje computacional y técnicas de preprocesamiento de datos para mejorar la detección de la dislexia a partir de los datos que fueron obtenidos en un trabajo previo.

El método propuesto en esta tesis confirman que la detección de dislexia mediante

el videojuego web como “MusVis” que recolecta información de tiempos de respuesta y errores ante pruebas visuales y auditivas puede ser mejorado mediante la selección correcta de las características extraídas de los datos que componen un conjunto de datos de entrenamiento para un modelo de aprendizaje computacional, sin embargo, es importante considerar otros aspectos para mejorar el desempeño de los modelos obtenidos mediante algoritmos de aprendizaje computacional. En primer lugar, el escalamiento de los datos de acuerdo al tipo de algoritmo utilizado. Además, del desbalance natural que presentan los datos adquiridos durante la interacción de los niños con el juego, puesto que la diferencia de niños que presentan dislexia es mayor en comparación a los que no presentan dislexia, dado que las estadísticas indican que 1 de cada 10 niños pueden presentar dislexia, algo que puede ocasionar que afecta el desempleo durante la evaluación de los modelos de aprendizaje automático. En adición, uno de los elementos claves para mostrar mejora en el proceso detección de dislexia es el uso de la métrica correcta. Durante la evaluación de los resultados se consideró que la sensibilidad es ideal para mejorar la identificación de casos de niños con dislexia, dado que esta medida no se ve afectada por los falsos positivos, considerando, el caso hipotético del desarrollo de una herramienta computacional para el apoyo en detección de dislexia, es preferible indicar que un niño tienen dislexia y que en un diagnóstico más exhaustivo por un experto sea descartado, a un caso contrario, donde un niño sea diagnosticado sin dislexia y no se realice más pruebas por el experto en trastornos de dislexia.

Los resultados de esta tesis muestran que el objetivo general, planteado al inicio, fue alcanzado al mejorar la detección de dislexia en el conjunto de datos “MusVis” mediante la obtención de modelos de aprendizaje computacional. Para lograr este objetivo se revisó el estado del arte respecto a la detección de dislexia mediante técnicas de aprendizaje computación. Además de, un método propuesto para generar tres diferentes tipos

de modelos para abordar el problema de la detección de dislexia sobre el conjunto de datos en particular estudiado. De acuerdo, con los resultados se evaluó el desempeño de cada tipo de modelos obtenido mediante métricas de F1 y sensibilidad. Lo que permitió la comparación entre los modelos y la propuesta original realizada en el estado del arte sobre un conjunto de prueba.

De acuerdo con los resultados obtenidos en esta tesis, se puede concluir que no existe evidencia que indique que la hipótesis planteada en esta tesis sea rechazada, debido a que los resultados de los modelos de tipo I, mejoraron significativamente los resultados en comparación con la propuesta original del trabajo previo.

En los resultados de la evaluación de los modelos del tipo I, II y III existen modelos que logran obtener un 78 % de sensibilidad en recuperación de la clase de niños con dislexia, en comparación de un 39 % de sensibilidad obtenido por la propuesta original del trabajo previo del estado del arte. Al realizar una evaluación cuantitativa de los modelos de clasificación, sus características seleccionadas, el aumento de datos, así como la configuración de los hiperparámetros, se obtuvo que los cuatro mejores algoritmos de aprendizaje supervisado para la detección de la dislexia en niños de preescolar son GB del tipo modelo I con 75 % de sensibilidad en la clase sin dislexia y 57 % en la clase con dislexia, para los modelos del tipo II, el modelo GB obtuvo una sensibilidad de 57 % y 74 % para las clases sin dislexia y con dislexia respectivamente, el modelo 2.2 NB-Estandar con un 15 % y 78 % de sensibilidad para las clases sin dislexia y con dislexia respectivamente. Para el tipo de modelo III, el modelo Ensamble-1-H-Normalizer obtuvo un 65 % y 48 % de sensibilidad para las clases sin dislexia y con dislexia respectivamente.

## 5.1. Aportaciones

- En esta tesis se plantea un método para la detección de dislexia que puede ser visto como un marco de referencia de trabajo para generar tres tipos de modelos. De los cuales, el tipo II obtuvo los mejores resultados.
- Además, se plantea la evaluación basada en sensibilidad por tipo de clases (dislexia y sin dislexia) para evaluar los modelos resultantes, con un enfoque basado en mejorar la recuperación de la clase dislexia.
- En los resultados de esta investigación se muestran que el uso de técnicas de escalamiento, selección de características y balanceo de datos con técnicas de oversampling SMOTE puede mejorar el resultado de evaluación de los modelos obtenidos mediante la métrica de sensibilidad.

## 5.2. Trabajo futuro

Debido a los objetivos propuestos al inicio de esta tesis, se descartaron algunos trabajos o la indagación de algunos trabajos, pero que pueden ser planteados para un trabajo a futuro.

- Debido a que solo se realizaron experimentos con siete clasificadores y dos ensambles de los mismos, aún podrían mejorarse, se deja para un futuro proponer diferentes modelos.
- La creación de un juego Web para recuperar información y ampliar el tamaño del conjunto de datos “MusVis”.
- La comparación entre distintas técnicas de balanceo de datos, como pueden ser el undersampling y oversampling.

# Anexos

## 5.3. Selección de características

### 5.3.1. Separación de datos

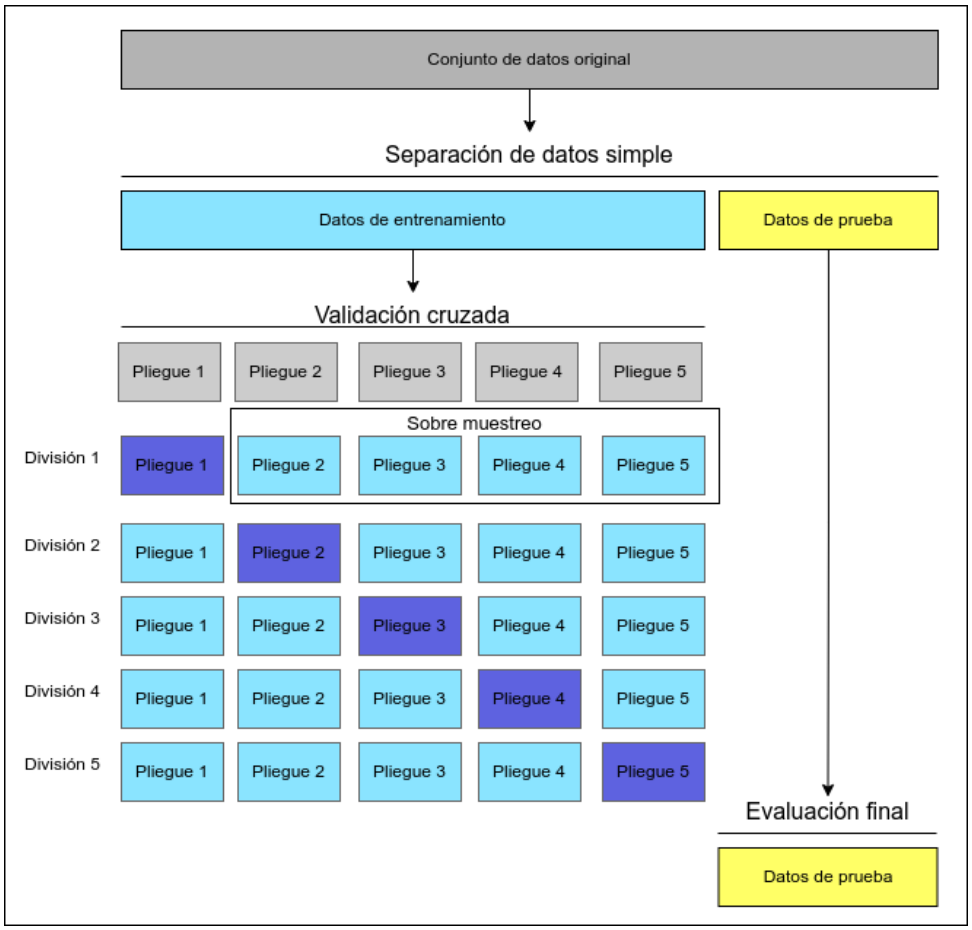


Figura 5.1: Separación de datos en entrenamiento y prueba

## 5.4. Selección de características

### 5.4.1. selección de características con ANOVA

Tabla 5.1: Porcentaje de percentiles seleccionados por Anova en cada modelo

| Modelo | Porcentaje de características utilizado por el modelo | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 % | 3 % | 6 % | 10 % | 15 % | 20 % | 30 % | 40 % | 60 % | 80 % | 100 % |
| NB-Estandar | | | | | | | | **x** | | | |
| NB-MinMax | | | | | | | **x** | | | | |
| NB-Normalizer | | | | | | | | | | **x** | |
| NB | | | | | | **x** | | | | | |
| RL-Estandar | | | | | | | | | | | **x** |
| RL-MinMax | | | | | | | | **x** | | | |
| RL-Normalizer | | | | | | | | | **x** | | |
| RL | | | | | | | | **x** | | | |
| SVM-Estandar | | | | | | | | **x** | | | |
| SVM-MinMax | | **x** | | | | | | | | | |
| SVM-Normalizer | | | | **x** | | | | | | | |
| SVM | | | | | **x** | | | | | | |
| ET-Estandar | | | | | | | **x** | | | | |
| ET-MinMax | | | | | | **x** | | | | | |
| ET-Normalizer | | | | | **x** | | | | | | |
| ET | | | | | | | | | | | **x** |
| RF-Estandar | | | | | | | **x** | | | | |
| RF-MinMax | | | | | | | | | | **x** | |
| RF-Normalizer | | | | | | | **x** | | | | |
| RF | | | | | | | **x** | | | | |
| MLP-Estandar | | | | | | | | | | | **x** |
| MLP-MinMax | | | | | | | | | **x** | | |
| MLP-Normalizer | | | | | | | | | | | **x** |
| MLP | | | | | | | | | | | **x** |
| GB-Estandar | | | | | | | | | | | **x** |
| GB-MinMax | | | | | | | | | | | **x** |
| GB-Normalizer | | | | | | | | | | | **x** |
| GB | | | | | | | | | | **x** | |

Lista de características seleccionadas por ANOVA para el entrenamiento de los algoritmos mostrados en la tabla 5.1

- 1 % de caracteristicas ANOVA
    1. mus_1_time_thirdclick_1_4
    2. vis_1_time_fourthclick_2_9
    3. vis_1_totalclicks_1_9
- 3 % de caracteristicas ANOVA
    1. mus_1_time_thirdclick_1_4
    2. vis_1_hits_1_9
    3. vis_1_out_1_9
    4. vis_1_out_4_4
    5. vis_1_time_fourthclick_2_9
    6. vis_1_totalclicks_1_9
    7. vis_1_totalclicks_4_4
- 6 % de caracteristicas ANOVA
    1. mus_1_averageclicktime_1_4
    2. mus_1_duration_4_4
    3. mus_1_firstclick_2_6
    4. mus_1_time_thirdclick_1_4
    5. mus_1_time_thirdclick_3_4
    6. mus_1_time_thirdclick_4_4
    7. vis_1_firstclick_2_9
    8. vis_1_hits_1_9
    9. vis_1_hits_4_4
    10. vis_1_out_1_9
    11. vis_1_out_4_4
    12. vis_1_time_fourthclick_2_9
    13. vis_1_totalclicks_1_9
    14. vis_1_totalclicks_4_4
- 10 % de caracteristicas ANOVA
    1. mus_1_averageclicktime_1_4
    2. mus_1_averageclicktime_4_6
    3. mus_1_duration_4_4
    4. mus_1_firstclick_2_6
    5. mus_1_precis_duration_4_4
    6. mus_1_time_fifthclick_2_4

7. mus_1_time_fourthclick_4_6
8. mus_1_time_secondclick_1_4
9. mus_1_time_sixclick_2_4
10. mus_1_time_thirdclick_1_4
11. mus_1_time_thirdclick_3_4
12. mus_1_time_thirdclick_4_4
13. vis_1_firstclick_2_9
14. vis_1_hits_1_9
15. vis_1_hits_4_4
16. vis_1_out_1_9
17. vis_1_out_4_4
18. vis_1_time_fourthclick_1_9
19. vis_1_time_fourthclick_2_9
20. vis_1_time_secondclick_2_4
21. vis_1_time_sixclick_3_9
22. vis_1_timelastclick_1_9
23. vis_1_totalclicks_1_9
24. vis_1_totalclicks_4_4

- 15 % de caracteristicas ANOVA

1. mus_1_averageclicktime_4_6
2. mus_1_duration_4_4
3. mus_1_duration_4_6
4. mus_1_firstclick_2_6
5. mus_1_precis_duration_4_4
6. mus_1_precis_duration_4_6
7. mus_1_time_fifthclick_2_4
8. mus_1_time_fourthclick_2_4
9. mus_1_time_fourthclick_4_6
10. mus_1_time_secondclick_1_4
11. mus_1_time_sixclick_2_4
12. mus_1_time_sixclick_4_6
13. mus_1_time_thirdclick_1_4
14. mus_1_time_thirdclick_3_4
15. mus_1_time_thirdclick_3_6
16. mus_1_time_thirdclick_4_4
17. mus_1_totalclicks_3_4
18. vis_1_firstclick_2_9
19. vis_1_hits_1_9
20. vis_1_hits_4_4
21. vis_1_out_1_9
22. vis_1_out_2_4
23. vis_1_out_3_9
24. vis_1_out_4_4
25. vis_1_time_fourthclick_1_9
26. vis_1_time_fourthclick_2_9
27. vis_1_time_secondclick_2_4
28. vis_1_time_secondclick_3_4
29. vis_1_time_sixclick_3_9
30. vis_1_timelastclick_1_9
31. vis_1_totalclicks_1_9
32. vis_1_totalclicks_4_4

- 20 % de caracteristicas ANOVA

1. lang_music
2. mus_1_averageclicktime_1_4
3. mus_1_averageclicktime_1_6
4. mus_1_averageclicktime_2_6
5. mus_1_averageclicktime_4_6
6. mus_1_duration_4_4
7. mus_1_duration_4_6
8. mus_1_firstclick_1_4
9. mus_1_firstclick_2_6
10. mus_1_instrucplayed_3_6
11. mus_1_precis_duration_1_6
12. mus_1_precis_duration_4_4
13. mus_1_precis_duration_4_6
14. mus_1_time_fifthclick_1_4
15. mus_1_time_fifthclick_2_4
16. mus_1_time_fifthclick_2_6
17. mus_1_time_fourthclick_2_4
18. mus_1_time_fourthclick_4_6
19. mus_1_time_secondclick_1_4
20. mus_1_time_sixclick_2_4
21. mus_1_time_sixclick_4_6
22. mus_1_time_thirdclick_1_4
23. mus_1_time_thirdclick_3_4
24. mus_1_time_thirdclick_3_6
25. mus_1_time_thirdclick_4_4
26. mus_1_totalclicks_3_4
27. vis_1_firstclick_2_9
28. vis_1_hits_1_9
29. vis_1_hits_4_4
30. vis_1_out_1_9
31. vis_1_out_2_4
32. vis_1_out_3_9
33. vis_1_out_4_4
34. vis_1_time_fourthclick_1_9
35. vis_1_time_fourthclick_2_9
36. vis_1_time_fourthclick_3_4
37. vis_1_time_secondclick_2_4
38. vis_1_time_secondclick_3_4
39. vis_1_time_secondclick_4_4
40. vis_1_time_sixclick_2_4
41. vis_1_time_sixclick_3_9
42. vis_1_time_sixclick_4_9
43. vis_1_timelastclick_1_9
44. vis_1_timelastclick_4_4
45. vis_1_totalclicks_1_9
46. vis_1_totalclicks_2_9
47. vis_1_totalclicks_4_4

- 30 % de caracteristicas ANOVA

1. lang_music
2. mus_1_averageclicktime_1_4
3. mus_1_averageclicktime_1_6
4. mus_1_averageclicktime_2_4
5. mus_1_averageclicktime_2_6
6. mus_1_averageclicktime_3_6
7. mus_1_averageclicktime_4_6
8. mus_1_duration_1_6
9. mus_1_duration_2_6
10. mus_1_duration_3_4
11. mus_1_duration_4_4
12. mus_1_duration_4_6
13. mus_1_firstclick_1_4
14. mus_1_firstclick_1_6
15. mus_1_firstclick_2_6
16. mus_1_instrucplayed_3_6
17. mus_1_precis_duration_1_6
18. mus_1_precis_duration_2_6
19. mus_1_precis_duration_3_4
20. mus_1_precis_duration_4_4
21. mus_1_precis_duration_4_6
22. mus_1_time_fifthclick_1_4
23. mus_1_time_fifthclick_2_4
24. mus_1_time_fifthclick_2_6
25. mus_1_time_fourthclick_1_6
26. mus_1_time_fourthclick_2_4
27. mus_1_time_fourthclick_4_6
28. mus_1_time_secondclick_1_4
29. mus_1_time_secondclick_1_6
30. mus_1_time_secondclick_2_6
31. mus_1_time_secondclick_4_4
32. mus_1_time_sixclick_2_4
33. mus_1_time_sixclick_3_4
34. mus_1_time_sixclick_4_6
35. mus_1_time_thirdclick_1_4
36. mus_1_time_thirdclick_3_4
37. mus_1_time_thirdclick_3_6
38. mus_1_time_thirdclick_4_4
39. mus_1_totalclicks_3_4
40. mus_1_totalclicks_4_4
41. mus_1_totalclicks_4_6
42. vis_1_firstclick_1_4
43. vis_1_firstclick_2_9
44. vis_1_hits_1_9
45. vis_1_hits_4_4
46. vis_1_miss_1_4
47. vis_1_miss_2_9
48. vis_1_out_1_9
49. vis_1_out_2_4
50. vis_1_out_3_9

51. vis_1_out_4_4
52. vis_1_time_fourthclick_1_9
53. vis_1_time_fourthclick_2_9
54. vis_1_time_fourthclick_3_4
55. vis_1_time_secondclick_1_9
56. vis_1_time_secondclick_2_4
57. vis_1_time_secondclick_3_4
58. vis_1_time_secondclick_4_4
59. vis_1_time_sixclick_2_4
60. vis_1_time_sixclick_3_9
61. vis_1_time_sixclick_4_9
62. vis_1_time_thirdclick_1_9
63. vis_1_timelastclick_1_9
64. vis_1_timelastclick_4_4
65. vis_1_totalclicks_1_9
66. vis_1_totalclicks_2_4
67. vis_1_totalclicks_2_9
68. vis_1_totalclicks_3_4
69. vis_1_totalclicks_3_9
70. vis_1_totalclicks_4_4

- 40 % de caracteristicas ANOVA

1. gender
2. memory
3. lang_music
4. mus_1_averageclicktime_1_4
5. mus_1_averageclicktime_1_6
6. mus_1_averageclicktime_2_4
7. mus_1_averageclicktime_2_6
8. mus_1_averageclicktime_3_6
9. mus_1_averageclicktime_4_4
10. mus_1_averageclicktime_4_6
11. mus_1_duration_1_6
12. mus_1_duration_2_6
13. mus_1_duration_3_4
14. mus_1_duration_4_4
15. mus_1_duration_4_6
16. mus_1_firstclick_1_4
17. mus_1_firstclick_1_6
18. mus_1_firstclick_2_6
19. mus_1_instrucplayed_1_4
20. mus_1_instrucplayed_2_4
21. mus_1_instrucplayed_3_6
22. mus_1_precis_duration_1_4
23. mus_1_precis_duration_1_6
24. mus_1_precis_duration_2_6
25. mus_1_precis_duration_3_4
26. mus_1_precis_duration_4_4
27. mus_1_precis_duration_4_6
28. mus_1_time_fifthclick_1_4

29. mus_1_time_fifthclick_1_6
30. mus_1_time_fifthclick_2_4
31. mus_1_time_fifthclick_2_6
32. mus_1_time_fifthclick_4_6
33. mus_1_time_fourthclick_1_6
34. mus_1_time_fourthclick_2_4
35. mus_1_time_fourthclick_2_6
36. mus_1_time_fourthclick_3_4
37. mus_1_time_fourthclick_4_6
38. mus_1_time_secondclick_1_4
39. mus_1_time_secondclick_1_6
40. mus_1_time_secondclick_2_6
41. mus_1_time_secondclick_4_4
42. mus_1_time_sixclick_1_4
43. mus_1_time_sixclick_1_6
44. mus_1_time_sixclick_2_4
45. mus_1_time_sixclick_3_4
46. mus_1_time_sixclick_3_6
47. mus_1_time_sixclick_4_6
48. mus_1_time_thirdclick_1_4
49. mus_1_time_thirdclick_3_4
50. mus_1_time_thirdclick_3_6
51. mus_1_time_thirdclick_4_4
52. mus_1_totalclicks_3_4
53. mus_1_totalclicks_4_4
54. mus_1_totalclicks_4_6
55. vis_1_firstclick_1_4
56. vis_1_firstclick_2_4
57. vis_1_firstclick_2_9
58. vis_1_firstclick_4_9
59. vis_1_hits_1_9
60. vis_1_hits_2_4
61. vis_1_hits_3_9
62. vis_1_hits_4_4
63. vis_1_miss_1_4
64. vis_1_miss_2_9
65. vis_1_out_1_9
66. vis_1_out_2_4
67. vis_1_out_3_4
68. vis_1_out_3_9
69. vis_1_out_4_4
70. vis_1_time_fifthclick_2_4
71. vis_1_time_fourthclick_1_9
72. vis_1_time_fourthclick_2_9
73. vis_1_time_fourthclick_3_4
74. vis_1_time_fourthclick_3_9
75. vis_1_time_secondclick_1_9
76. vis_1_time_secondclick_2_4
77. vis_1_time_secondclick_3_4
78. vis_1_time_secondclick_4_4
79. vis_1_time_sixclick_2_4
80. vis_1_time_sixclick_3_4

81. vis_1_time_sixclick_3_9
82. vis_1_time_sixclick_4_4
83. vis_1_time_sixclick_4_9
84. vis_1_time_thirdclick_1_9
85. vis_1_timelastclick_1_9
86. vis_1_timelastclick_4_4
87. vis_1_totalclicks_1_4
88. vis_1_totalclicks_1_9
89. vis_1_totalclicks_2_4
90. vis_1_totalclicks_2_9
91. vis_1_totalclicks_3_4
92. vis_1_totalclicks_4_4
93. vis_1_totalclicks_3_9

- 60 % de caracteristicas ANOVA

1. gender
2. memory
3. db_radiodifficult_mus
4. lang_music
5. mus_1_averageclicktime_1_4
6. mus_1_averageclicktime_1_6
7. mus_1_averageclicktime_2_4
8. mus_1_averageclicktime_2_6
9. mus_1_averageclicktime_3_6
10. mus_1_averageclicktime_4_4
11. mus_1_averageclicktime_4_6
12. mus_1_duration_1_4
13. mus_1_duration_1_6
14. mus_1_duration_2_6
15. mus_1_duration_3_4
16. mus_1_duration_4_4
17. mus_1_duration_4_6
18. mus_1_firstclick_1_4
19. mus_1_firstclick_1_6
20. mus_1_firstclick_2_6
21. mus_1_firstclick_3_6
22. mus_1_instrucplayed_1_4
23. mus_1_instrucplayed_2_4
24. mus_1_instrucplayed_3_6
25. mus_1_logic_2_4
26. mus_1_logic_4_6
27. mus_1_precis_duration_1_4
28. mus_1_precis_duration_1_6
29. mus_1_precis_duration_2_6
30. mus_1_precis_duration_3_4
31. mus_1_precis_duration_3_6
32. mus_1_precis_duration_4_4
33. mus_1_precis_duration_4_6
34. mus_1_time_fifthclick_1_4

35. mus_1_time_fifthclick_1_6
36. mus_1_time_fifthclick_2_4
37. mus_1_time_fifthclick_2_6
38. mus_1_time_fifthclick_4_6
39. mus_1_time_fourthclick_1_6
40. mus_1_time_fourthclick_2_4
41. mus_1_time_fourthclick_2_6
42. mus_1_time_fourthclick_3_4
43. mus_1_time_fourthclick_3_6
44. mus_1_time_fourthclick_4_6
45. mus_1_time_secondclick_1_4
46. mus_1_time_secondclick_1_6
47. mus_1_time_secondclick_2_6
48. mus_1_time_secondclick_4_4
49. mus_1_time_secondclick_4_6
50. mus_1_time_sixclick_1_4
51. mus_1_time_sixclick_1_6
52. mus_1_time_sixclick_2_4
53. mus_1_time_sixclick_3_4
54. mus_1_time_sixclick_3_6
55. mus_1_time_sixclick_4_6
56. mus_1_time_thirdclick_1_4
57. mus_1_time_thirdclick_1_6
58. mus_1_time_thirdclick_2_4
59. mus_1_time_thirdclick_2_6
60. mus_1_time_thirdclick_3_4
61. mus_1_time_thirdclick_3_6
62. mus_1_time_thirdclick_4_4
63. mus_1_time_thirdclick_4_6
64. mus_1_totalclicks_3_4
65. mus_1_totalclicks_4_4
66. mus_1_totalclicks_4_6
67. vis_1_accuarcy_1_4
68. vis_1_accuarcy_1_9
69. vis_1_efficiency_3_4
70. vis_1_efficiency_4_4
71. vis_1_firstclick_1_4
72. vis_1_firstclick_1_9
73. vis_1_firstclick_2_4
74. vis_1_firstclick_2_9
75. vis_1_firstclick_3_4
76. vis_1_firstclick_4_9
77. vis_1_hits_1_9
78. vis_1_hits_2_4
79. vis_1_hits_3_4
80. vis_1_hits_3_9
81. vis_1_hits_4_4
82. vis_1_latin_1_4
83. vis_1_latin_1_9
84. vis_1_latin_2_4
85. vis_1_latin_2_9
86. vis_1_latin_3_4

87. vis_1_latin_3_9
88. vis_1_latin_4_4
89. vis_1_latin_4_9
90. vis_1_miss_1_4
91. vis_1_miss_1_9
92. vis_1_miss_2_9
93. vis_1_miss_3_4
94. vis_1_miss_3_9
95. vis_1_miss_4_9
96. vis_1_out_1_9
97. vis_1_out_2_4
98. vis_1_out_3_4
99. vis_1_out_3_9
100. vis_1_out_4_4
101. vis_1_out_4_9
102. vis_1_round_number_1_4
103. vis_1_round_number_1_9
104. vis_1_round_number_4_4
105. vis_1_round_number_4_9
106. vis_1_time_fifthclick_2_4
107. vis_1_time_fifthclick_2_9
108. vis_1_time_fifthclick_4_9
109. vis_1_time_fourthclick_1_4
110. vis_1_time_fourthclick_1_9
111. vis_1_time_fourthclick_2_9
112. vis_1_time_fourthclick_3_4
113. vis_1_time_fourthclick_3_9
114. vis_1_time_secondclick_1_9
115. vis_1_time_secondclick_2_4
116. vis_1_time_secondclick_2_9
117. vis_1_time_secondclick_3_4
118. vis_1_time_secondclick_4_4
119. vis_1_time_sixclick_1_9
120. vis_1_time_sixclick_2_4
121. vis_1_time_sixclick_3_4
122. vis_1_time_sixclick_3_9
123. vis_1_time_sixclick_4_4
124. vis_1_time_sixclick_4_9
125. vis_1_time_thirdclick_1_9
126. vis_1_time_thirdclick_3_9
127. vis_1_timelastclick_1_4
128. vis_1_timelastclick_1_9
129. vis_1_timelastclick_2_9
130. vis_1_timelastclick_3_9
131. vis_1_timelastclick_4_4
132. vis_1_totalclicks_1_4
133. vis_1_totalclicks_1_9
134. vis_1_totalclicks_2_4
135. vis_1_totalclicks_2_9
136. vis_1_totalclicks_3_4
137. vis_1_totalclicks_3_9
138. vis_1_totalclicks_4_4
139. vis_1_totalclicks_4_9

- 80 % de caracteristicas ANOVA

1. age
2. gender
3. memory
4. db_radiodifficult_mus
5. lang_music
6. mus_1_averageclicktime_1_4
7. mus_1_averageclicktime_1_6
8. mus_1_averageclicktime_2_4
9. mus_1_averageclicktime_2_6
10. mus_1_averageclicktime_3_6
11. mus_1_averageclicktime_4_4
12. mus_1_averageclicktime_4_6
13. mus_1_duration_1_4
14. mus_1_duration_1_6
15. mus_1_duration_2_6
16. mus_1_duration_3_4
17. mus_1_duration_3_6
18. mus_1_duration_4_4
19. mus_1_duration_4_6
20. mus_1_firstclick_1_4
21. mus_1_firstclick_1_6
22. mus_1_firstclick_2_6
23. mus_1_firstclick_3_4
24. mus_1_firstclick_3_6
25. mus_1_firstclick_4_4
26. mus_1_firstclick_4_6
27. mus_1_instrucplayed_1_4
28. mus_1_instrucplayed_1_6
29. mus_1_instrucplayed_2_4
30. mus_1_instrucplayed_2_6
31. mus_1_instrucplayed_3_4
32. mus_1_instrucplayed_3_6
33. mus_1_instrucplayed_4_6
34. mus_1_logic_1_4
35. mus_1_logic_1_6
36. mus_1_logic_2_4
37. mus_1_logic_3_4
38. mus_1_logic_3_6
39. mus_1_logic_4_6
40. mus_1_precis_duration_1_4
41. mus_1_precis_duration_1_6
42. mus_1_precis_duration_2_6
43. mus_1_precis_duration_3_4
44. mus_1_precis_duration_3_6
45. mus_1_precis_duration_4_4
46. mus_1_precis_duration_4_6
47. mus_1_round_number_1_4
48. mus_1_round_number_1_6
49. mus_1_round_number_2_4
50. mus_1_round_number_2_6

51. mus_1_round_number_3_4
52. mus_1_round_number_3_6
53. mus_1_round_number_4_4
54. mus_1_round_number_4_6
55. mus_1_time_fifthclick_1_4
56. mus_1_time_fifthclick_1_6
57. mus_1_time_fifthclick_2_4
58. mus_1_time_fifthclick_2_6
59. mus_1_time_fifthclick_3_4
60. mus_1_time_fifthclick_4_4
61. mus_1_time_fifthclick_4_6
62. mus_1_time_fourthclick_1_4
63. mus_1_time_fourthclick_1_6
64. mus_1_time_fourthclick_2_4
65. mus_1_time_fourthclick_2_6
66. mus_1_time_fourthclick_3_4
67. mus_1_time_fourthclick_3_6
68. mus_1_time_fourthclick_4_6
69. mus_1_time_secondclick_1_4
70. mus_1_time_secondclick_1_6
71. mus_1_time_secondclick_2_4
72. mus_1_time_secondclick_2_6
73. mus_1_time_secondclick_3_4
74. mus_1_time_secondclick_3_6
75. mus_1_time_secondclick_4_4
76. mus_1_time_secondclick_4_6
77. mus_1_time_sixclick_1_4
78. mus_1_time_sixclick_1_6
79. mus_1_time_sixclick_2_4
80. mus_1_time_sixclick_3_4
81. mus_1_time_sixclick_3_6
82. mus_1_time_sixclick_4_4
83. mus_1_time_sixclick_4_6
84. mus_1_time_thirdclick_1_4
85. mus_1_time_thirdclick_1_6
86. mus_1_time_thirdclick_2_4
87. mus_1_time_thirdclick_2_6
88. mus_1_time_thirdclick_3_4
89. mus_1_time_thirdclick_3_6
90. mus_1_time_thirdclick_4_4
91. mus_1_time_thirdclick_4_6
92. mus_1_totalclicks_2_6
93. mus_1_totalclicks_3_4
94. mus_1_totalclicks_4_4
95. mus_1_totalclicks_4_6
96. vis_1_accuarcy_1_4
97. vis_1_accuarcy_1_9
98. vis_1_accuarcy_4_4
99. vis_1_accuarcy_4_9
100. vis_1_efficiency_1_9
101. vis_1_efficiency_2_4
102. vis_1_efficiency_2_9

103. vis_1_efficiency_3_4
104. vis_1_efficiency_3_9
105. vis_1_efficiency_4_4
106. vis_1_firstclick_1_4
107. vis_1_firstclick_1_9
108. vis_1_firstclick_2_4
109. vis_1_firstclick_2_9
110. vis_1_firstclick_3_4
111. vis_1_firstclick_3_9
112. vis_1_firstclick_4_9
113. vis_1_hits_1_9
114. vis_1_hits_2_4
115. vis_1_hits_3_4
116. vis_1_hits_3_9
117. vis_1_hits_4_4
118. vis_1_hits_4_9
119. vis_1_latin_1_4
120. vis_1_latin_1_9
121. vis_1_latin_2_4
122. vis_1_latin_2_9
123. vis_1_latin_3_4
124. vis_1_latin_3_9
125. vis_1_latin_4_4
126. vis_1_latin_4_9
127. vis_1_miss_1_4
128. vis_1_miss_1_9
129. vis_1_miss_2_9
130. vis_1_miss_3_4
131. vis_1_miss_3_9
132. vis_1_miss_4_9
133. vis_1_out_1_9
134. vis_1_out_2_4
135. vis_1_out_3_4
136. vis_1_out_3_9
137. vis_1_out_4_4
138. vis_1_out_4_9
139. vis_1_round_number_1_4
140. vis_1_round_number_1_9
141. vis_1_round_number_4_4
142. vis_1_round_number_4_9
143. vis_1_time_fifthclick_1_4
144. vis_1_time_fifthclick_1_9
145. vis_1_time_fifthclick_2_4
146. vis_1_time_fifthclick_2_9
147. vis_1_time_fifthclick_3_9
148. vis_1_time_fifthclick_4_4
149. vis_1_time_fifthclick_4_9
150. vis_1_time_fourthclick_1_4
151. vis_1_time_fourthclick_1_9
152. vis_1_time_fourthclick_2_9
153. vis_1_time_fourthclick_3_4
154. vis_1_time_fourthclick_3_9

155. vis_1_time_fourthclick_4_4
156. vis_1_time_fourthclick_4_9
157. vis_1_time_secondclick_1_9
158. vis_1_time_secondclick_2_4
159. vis_1_time_secondclick_2_9
160. vis_1_time_secondclick_3_4
161. vis_1_time_secondclick_4_4
162. vis_1_time_secondclick_4_9
163. vis_1_time_sixclick_1_9
164. vis_1_time_sixclick_2_4
165. vis_1_time_sixclick_3_4
166. vis_1_time_sixclick_3_9
167. vis_1_time_sixclick_4_4
168. vis_1_time_sixclick_4_9
169. vis_1_time_thirdclick_1_4
170. vis_1_time_thirdclick_1_9
171. vis_1_time_thirdclick_3_9
172. vis_1_timelastclick_1_4
173. vis_1_timelastclick_1_9
174. vis_1_timelastclick_2_9
175. vis_1_timelastclick_3_9
176. vis_1_timelastclick_4_4
177. vis_1_timelastclick_4_9
178. vis_1_totalclicks_1_4
179. vis_1_totalclicks_1_9
180. vis_1_totalclicks_2_4
181. vis_1_totalclicks_2_9
182. vis_1_totalclicks_3_4
183. vis_1_totalclicks_3_9
184. vis_1_totalclicks_4_4
185. vis_1_totalclicks_4_9

- 100 % de caracteristicas ANOVA

1. mus_1_averageclicktime_1_4
2. mus_1_averageclicktime_1_6
3. mus_1_averageclicktime_2_4
4. mus_1_averageclicktime_2_6
5. mus_1_averageclicktime_3_4
6. mus_1_averageclicktime_3_6
7. mus_1_averageclicktime_4_4
8. mus_1_averageclicktime_4_6
9. mus_1_duration_1_4
10. mus_1_duration_1_6
11. mus_1_duration_2_4
12. mus_1_duration_2_6
13. mus_1_duration_3_4
14. mus_1_duration_3_6
15. mus_1_duration_4_4
16. mus_1_duration_4_6

17. mus_1_firstclick_1_4
18. mus_1_firstclick_1_6
19. mus_1_firstclick_2_4
20. mus_1_firstclick_2_6
21. mus_1_firstclick_3_4
22. mus_1_firstclick_3_6
23. mus_1_firstclick_4_4
24. mus_1_firstclick_4_6
25. mus_1_instrucplayed_1_4
26. mus_1_instrucplayed_1_6
27. mus_1_instrucplayed_2_4
28. mus_1_instrucplayed_2_6
29. mus_1_instrucplayed_3_4
30. mus_1_instrucplayed_3_6
31. mus_1_instrucplayed_4_4
32. mus_1_instrucplayed_4_6
33. mus_1_logic_1_4
34. mus_1_logic_1_6
35. mus_1_logic_2_4
36. mus_1_logic_2_6
37. mus_1_logic_3_4
38. mus_1_logic_3_6
39. mus_1_logic_4_4
40. mus_1_logic_4_6
41. mus_1_precis_duration_1_4
42. mus_1_precis_duration_1_6
43. mus_1_precis_duration_2_4
44. mus_1_precis_duration_2_6
45. mus_1_precis_duration_3_4
46. mus_1_precis_duration_3_6
47. mus_1_precis_duration_4_4
48. mus_1_precis_duration_4_6
49. mus_1_round_number_1_4
50. mus_1_round_number_1_6
51. mus_1_round_number_2_4
52. mus_1_round_number_2_6
53. mus_1_round_number_3_4
54. mus_1_round_number_3_6
55. mus_1_round_number_4_4
56. mus_1_round_number_4_6
57. mus_1_time_fifthclick_1_4
58. mus_1_time_fifthclick_1_6
59. mus_1_time_fifthclick_2_4
60. mus_1_time_fifthclick_2_6
61. mus_1_time_fifthclick_3_4
62. mus_1_time_fifthclick_3_6
63. mus_1_time_fifthclick_4_4
64. mus_1_time_fifthclick_4_6
65. mus_1_time_fourthclick_1_4
66. mus_1_time_fourthclick_1_6
67. mus_1_time_fourthclick_2_4
68. mus_1_time_fourthclick_2_6

69. mus_1_time_fourthclick_3_4
70. mus_1_time_fourthclick_3_6
71. mus_1_time_fourthclick_4_4
72. mus_1_time_fourthclick_4_6
73. mus_1_time_secondclick_1_4
74. mus_1_time_secondclick_1_6
75. mus_1_time_secondclick_2_4
76. mus_1_time_secondclick_2_6
77. mus_1_time_secondclick_3_4
78. mus_1_time_secondclick_3_6
79. mus_1_time_secondclick_4_4
80. mus_1_time_secondclick_4_6
81. mus_1_time_sixclick_1_4
82. mus_1_time_sixclick_1_6
83. mus_1_time_sixclick_2_4
84. mus_1_time_sixclick_2_6
85. mus_1_time_sixclick_3_4
86. mus_1_time_sixclick_3_6
87. mus_1_time_sixclick_4_4
88. mus_1_time_sixclick_4_6
89. mus_1_time_thirdclick_1_4
90. mus_1_time_thirdclick_1_6
91. mus_1_time_thirdclick_2_4
92. mus_1_time_thirdclick_2_6
93. mus_1_time_thirdclick_3_4
94. mus_1_time_thirdclick_3_6
95. mus_1_time_thirdclick_4_4
96. mus_1_time_thirdclick_4_6
97. mus_1_totalclicks_1_4
98. mus_1_totalclicks_1_6
99. mus_1_totalclicks_2_4
100. mus_1_totalclicks_2_6
101. mus_1_totalclicks_3_4
102. mus_1_totalclicks_3_6
103. mus_1_totalclicks_4_4
104. mus_1_totalclicks_4_6
105. vis_1_accuarcy_1_4
106. vis_1_accuarcy_1_9
107. vis_1_accuarcy_2_4
108. vis_1_accuarcy_2_9
109. vis_1_accuarcy_3_4
110. vis_1_accuarcy_3_9
111. vis_1_accuarcy_4_4
112. vis_1_accuarcy_4_9
113. vis_1_efficiency_1_4
114. vis_1_efficiency_1_9
115. vis_1_efficiency_2_4
116. vis_1_efficiency_2_9
117. vis_1_efficiency_3_4
118. vis_1_efficiency_3_9
119. vis_1_efficiency_4_4
120. vis_1_efficiency_4_9

121. vis_1_firstclick_1_4
122. vis_1_firstclick_1_9
123. vis_1_firstclick_2_4
124. vis_1_firstclick_2_9
125. vis_1_firstclick_3_4
126. vis_1_firstclick_3_9
127. vis_1_firstclick_4_4
128. vis_1_firstclick_4_9
129. vis_1_hits_1_4
130. vis_1_hits_1_9
131. vis_1_hits_2_4
132. vis_1_hits_2_9
133. vis_1_hits_3_4
134. vis_1_hits_3_9
135. vis_1_hits_4_4
136. vis_1_hits_4_9
137. vis_1_latin_1_4
138. vis_1_latin_1_9
139. vis_1_latin_2_4
140. vis_1_latin_2_9
141. vis_1_latin_3_4
142. vis_1_latin_3_9
143. vis_1_latin_4_4
144. vis_1_latin_4_9
145. vis_1_miss_1_4
146. vis_1_miss_1_9
147. vis_1_miss_2_4
148. vis_1_miss_2_9
149. vis_1_miss_3_4
150. vis_1_miss_3_9
151. vis_1_miss_4_4
152. vis_1_miss_4_9
153. vis_1_out_1_4
154. vis_1_out_1_9
155. vis_1_out_2_4
156. vis_1_out_2_9
157. vis_1_out_3_4
158. vis_1_out_3_9
159. vis_1_out_4_4
160. vis_1_out_4_9
161. vis_1_round_number_1_4
162. vis_1_round_number_1_9
163. vis_1_round_number_2_4
164. vis_1_round_number_2_9
165. vis_1_round_number_3_4
166. vis_1_round_number_3_9
167. vis_1_round_number_4_4
168. vis_1_round_number_4_9
169. vis_1_time_fifthclick_1_4
170. vis_1_time_fifthclick_1_9
171. vis_1_time_fifthclick_2_4
172. vis_1_time_fifthclick_2_9

173. vis_1_time_fifthclick_3_4
174. vis_1_time_fifthclick_3_9
175. vis_1_time_fifthclick_4_4
176. vis_1_time_fifthclick_4_9
177. vis_1_time_fourthclick_1_4
178. vis_1_time_fourthclick_1_9
179. vis_1_time_fourthclick_2_4
180. vis_1_time_fourthclick_2_9
181. vis_1_time_fourthclick_3_4
182. vis_1_time_fourthclick_3_9
183. vis_1_time_fourthclick_4_4
184. vis_1_time_fourthclick_4_9
185. vis_1_time_secondclick_1_4
186. vis_1_time_secondclick_1_9
187. vis_1_time_secondclick_2_4
188. vis_1_time_secondclick_2_9
189. vis_1_time_secondclick_3_4
190. vis_1_time_secondclick_3_9
191. vis_1_time_secondclick_4_4
192. vis_1_time_secondclick_4_9
193. vis_1_time_sixclick_1_4
194. vis_1_time_sixclick_1_9
195. vis_1_time_sixclick_2_4
196. vis_1_time_sixclick_2_9
197. vis_1_time_sixclick_3_4
198. vis_1_time_sixclick_3_9
199. vis_1_time_sixclick_4_4
200. vis_1_time_sixclick_4_9
201. vis_1_time_thirdclick_1_4
202. vis_1_time_thirdclick_1_9
203. vis_1_time_thirdclick_2_4
204. vis_1_time_thirdclick_2_9
205. vis_1_time_thirdclick_3_4
206. vis_1_time_thirdclick_3_9
207. vis_1_time_thirdclick_4_4
208. vis_1_time_thirdclick_4_9
209. vis_1_timelastclick_1_4
210. vis_1_timelastclick_1_9
211. vis_1_timelastclick_2_4
212. vis_1_timelastclick_2_9
213. vis_1_timelastclick_3_4
214. vis_1_timelastclick_3_9
215. vis_1_timelastclick_4_4
216. vis_1_timelastclick_4_9
217. vis_1_totalclicks_1_4
218. vis_1_totalclicks_1_9
219. vis_1_totalclicks_2_4
220. vis_1_totalclicks_2_9
221. vis_1_totalclicks_3_4
222. vis_1_totalclicks_3_9
223. vis_1_totalclicks_4_4
224. vis_1_totalclicks_4_9

# Experimento 2

### 5.4.2. Entrenamiento MusVis balanceado y selección de características Random Forest

Tabla 5.2: Resultados de ensambles con los datos balanceados, Validación cruzada (5 folders), escalamiento y selección de características Bosque aleatorio

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | S | F1 | B Acc | DE (S) | S | F1 | B Acc | DE. (S) |
| GB-Estandar | 100 % | 100 % | 100 % | 0 | 67.5 % | 67.2 % | 67.5 % | 0.045 |
| MLP-MinMax | 100 % | 100 % | 100 % | 0 | 66.8 % | 66.4 % | 66.8 % | 0.114 |
| GB | 100 % | 100 % | 100 % | 0 | 66.2 % | 65.9 % | 66.2 % | 0.088 |
| GB-MinMax | 100 % | 100 % | 100 % | 0 | 64.8 % | 64.9 % | 64.8 % | 0.082 |
| RL-Estandar | 100 % | 100 % | 100 % | 0 | 62.6 % | 62.4 % | 62.6 % | 0.142 |
| GB-Normalizer | 100 % | 100 % | 100 % | 0 | 60.7 % | 60.8 % | 60.7 % | 0.071 |
| RL-Normalizer | 86.4 % | 86.4 % | 86.4 % | 0.023 | 59.9 % | 59.8 % | 59.9 % | 0.117 |
| RL | 100 % | 100 % | 100 % | 0 | 60.0 % | 59.5 % | 60.0 % | 0.067 |
| MLP-Estandar | 100 % | 100 % | 100 % | 0 | 58.4 % | 58.3 % | 58.4 % | 0.112 |
| SVM-MinMax | 90.3 % | 90.3 % | 90.3 % | 0.017 | 56.8 % | 56.7 % | 56.8 % | 0.088 |
| ET-Estandar | 100 % | 100 % | 100 % | 0 | 56.9 % | 56.5 % | 56.9 % | 0.083 |
| ET | 100 % | 100 % | 100 % | 0 | 56.3 % | 55.7 % | 56.3 % | 0.059 |
| MLP-Normalizer | 100 % | 100 % | 100 % | 0 | 56.0 % | 55.7 % | 56.0 % | 0.052 |
| ET-MinMax | 100 % | 100 % | 100 % | 0 | 56.4 % | 55.6 % | 56.4 % | 0.065 |
| RF-Normalizer | 100 % | 100 % | 100 % | 0 | 55.8 % | 55.5 % | 55.8 % | 0.016 |
| RF | 100 % | 100 % | 100 % | 0 | 55.2 % | 55.1 % | 55.2 % | 0.072 |
| RL-MinMax | 85.4 % | 85.4 % | 0.854 | 0.028 | 54.2 % | 54.2 % | 54.2 % | 0.113 |
| ET-Normalizer | 100 % | 100 % | 100 % | 0 | 55.1 % | 53.5 % | 55.1 % | 0.052 |
| NB-Normalizer | 74.3 % | 74.2 % | 0.743 | 0.037 | 53.1 % | 52.4 % | 53.1 % | 0.044 |
| SVM-Estandar | 97.4 % | 97.4 % | 0.974 | 0.005 | 53.9 % | 51.9 % | 53.9 % | 0.063 |
| RF-Estandar | 100 % | 100 % | 100 % | 0 | 52.7 % | 51.8 % | 52.7 % | 0.072 |
| RF-MinMax | 100 % | 100 % | 100 % | 0 | 51.8 % | 51.4 % | 51.8 % | 0.063 |
| NB | 74.9 % | 74.5 % | 0.749 | 0.032 | 51.7 % | 51.1 % | 51.7 % | 0.037 |
| NB-Estandar | 78.1 % | 77.4 % | 0.781 | 0.059 | 50.1 % | 49.2 % | 50.1 % | 0.08 |
| NB-MinMax | 76.4 % | 75.7 % | 0.764 | 0.061 | 49.8 % | 48.9 % | 49.8 % | 0.082 |
| MLP | 86.1 % | 85.3 % | 0.861 | 0.119 | 46.1 % | 45.7 % | 46.1 % | 0.055 |
| SVM | 59.4 % | 57.6 % | 0.594 | 0.025 | 46.8 % | 45.2 % | 46.8 % | 0.054 |
| SVM-Normalizer | 66.7 % | 65.9 % | 0.667 | 0.021 | 47.3 % | 44.6 % | 47.3 % | 0.044 |

Nota: Resultados de Sensibilidad (S) y F1 de la validación cruzada con un k = 5 en los modelos: Bayes Ingenuo (NB), Regresión Logística (LR), Máquina de Soporte de Vectores (SVM), Árboles Extremadamente Aleatorios (ET), Bosque Aleatorio (RF), Perceptrón Multicapa (MLP) y Potenciación del Gradiente (GB) en el conjunto de datos MusVis balanceado utilizando selección de características mediante Bosque Aleatorio y con técnicas escalamiento: Min-Max, Estandar y L2-Normalizer.

Tabla 5.3: Resultados con los datos balanceados, Validación cruzada (5 folders), escalamiento y selección de características Bosque aleatorio por clase.

| Modelo | Entrenamiento | | | | Prueba | | | |
|---|---|---|---|---|---|---|---|---|
| | Sin dislexia | | Con dislexia | | Sin dislexia | | Con dislexia | |
| | S | F1 | S | F1 | S | F1 | S | F1 |
| GB-Estandar | 100 % | 100 % | 100 % | 100 % | 75.8 % | 75.6 % | 59.2 % | 58.8 % |
| GB | 100 % | 100 % | 100 % | 100 % | 74.6 % | 74.7 % | 57.9 % | 57.1 % |
| MLP-MinMax | 100 % | 100 % | 100 % | 100 % | 76.5 % | 75.4 % | 57.1 % | 57.4 % |
| RL-Estandar | 100 % | 100 % | 100 % | 100 % | 71.4 % | 71.7 % | 53.8 % | 53.1 % |
| RL | 100 % | 100 % | 100 % | 100 % | 66.2 % | 68.2 % | 53.8 % | 50.7 % |
| SVM-Normalizer | 56.0 % | 62.1 % | 77.3 % | 69.7 % | 42.0 % | 48.4 % | 52.7 % | 40.7 % |
| GB-MinMax | 100 % | 100 % | 100 % | 100 % | 77.0 % | 75.1 % | 52.5 % | 54.6 % |
| NB-Normalizer | 67.8 % | 72.5 % | 80.8 % | 75.9 % | 54.7 % | 59.4 % | 51.5 % | 45.4 % |
| RL-Normalizer | 85.9 % | 86.3 % | 86.9 % | 86.5 % | 70.1 % | 69.8 % | 49.7 % | 49.7 % |
| GB-Normalizer | 100 % | 100 % | 100 % | 100 % | 73.2 % | 71.5 % | 48.2 % | 50.1 % |
| MLP-Normalizer | 100 % | 100 % | 100 % | 100 % | 65.6 % | 66.3 % | 46.3 % | 45.1 % |
| RL-MinMax | 85.2 % | 85.3 % | 85.6 % | 85.4 % | 64.3 % | 64.9 % | 44.2 % | 43.4 % |
| MLP-Estandar | 100 % | 100 % | 100 % | 100 % | 73.9 % | 71.2 % | 42.9 % | 45.5 % |
| MLP | 91.8 % | 87.6 % | 80.4 % | 82.9 % | 53.5 % | 56.3 % | 38.8 % | 35.1 % |
| NB-MinMax | 77.8 % | 77.0 % | 75.1 % | 74.4 % | 61.1 % | 61.1 % | 38.5 % | 36.7 % |
| NB | 74.8 % | 74.5 % | 74.9 % | 74.5 % | 66.0 % | 64.3 % | 37.4 % | 38.0 % |
| RF | 100 % | 100 % | 100 % | 100 % | 73.9 % | 69.5 % | 36.5 % | 40.7 % |
| SVM-MinMax | 92.5 % | 90.5 % | 88.2 % | 90.1 % | 77.1 % | 71.7 % | 36.4 % | 41.8 % |
| RF-Normalizer | 100 % | 100 % | 100 % | 100 % | 79.5 % | 72.3 % | 32.2 % | 38.6 % |
| ET | 100 % | 100 % | 100 % | 100 % | 81.5 % | 73.3 % | 31.1 % | 38.1 % |
| RF-MinMax | 100 % | 100 % | 100 % | 100 % | 72.55 | 67.5 % | 31.1 % | 35.2 % |
| ET-Estandar | 100 % | 100 % | 100 % | 100 % | 82.9 % | 73.9 % | 31.0 % | 39.0 % |
| ET-MinMax | 100 % | 100 % | 100 % | 100 % | 82.7 % | 73.8 % | 30.0 % | 37.4 % |
| SVM | 74.5 % | 64.3 % | 44.3 % | 50.8 % | 64.7 % | 61.5 % | 28.8 % | 29.0 % |
| ET-Normalizer | 100 % | 100 % | 100 % | 100 % | 82.2 % | 73.0 % | 28.0 % | 34.1 % |
| RF-Estandar | 100 % | 100 % | 100 % | 100 % | 79.6 % | 71.0 % | 25.7 % | 32.6 % |
| SVM-Estandar | 98.8 % | 97.4 % | 96.0 % | 97.4 % | 82.8 % | 72.7 % | 24.9 % | 31.2 % |
| NB-Estandar | 88.2 % | 80.1 % | 68.1 % | 74.7 % | 75.6 % | 68.2 % | 24.6 % | 30.2 % |

Nota: Resultados de Sensibilidad (S) y F1 por clase en la validación cruzada con un k = 5 en los modelos: Bayes Ingenuo (NB), Regresión Logística (LR), Máquina de Soporte de Vectores (SVM), Árboles Extremadamente Aleatorios (ET), Bosque Aleatorio (RF), Perceptrón Multicapa (MLP) y Potenciación del Gradiente (GB) en el conjunto de datos MusVis balanceado utilizando selección de características mediante Bosque aleatorio y con técnicas escalamiento: Min-Max, Estandar y L2-Normalizer.

### 5.4.3. Resultados del experimento 2: selección de características con bosque aleatorio

En la figura 5.2 se muestra la sensibilidad que logro cada algoritmo para predecir a los infantes con y sin dislexia utilizando selección de características con bosque aleatorio.

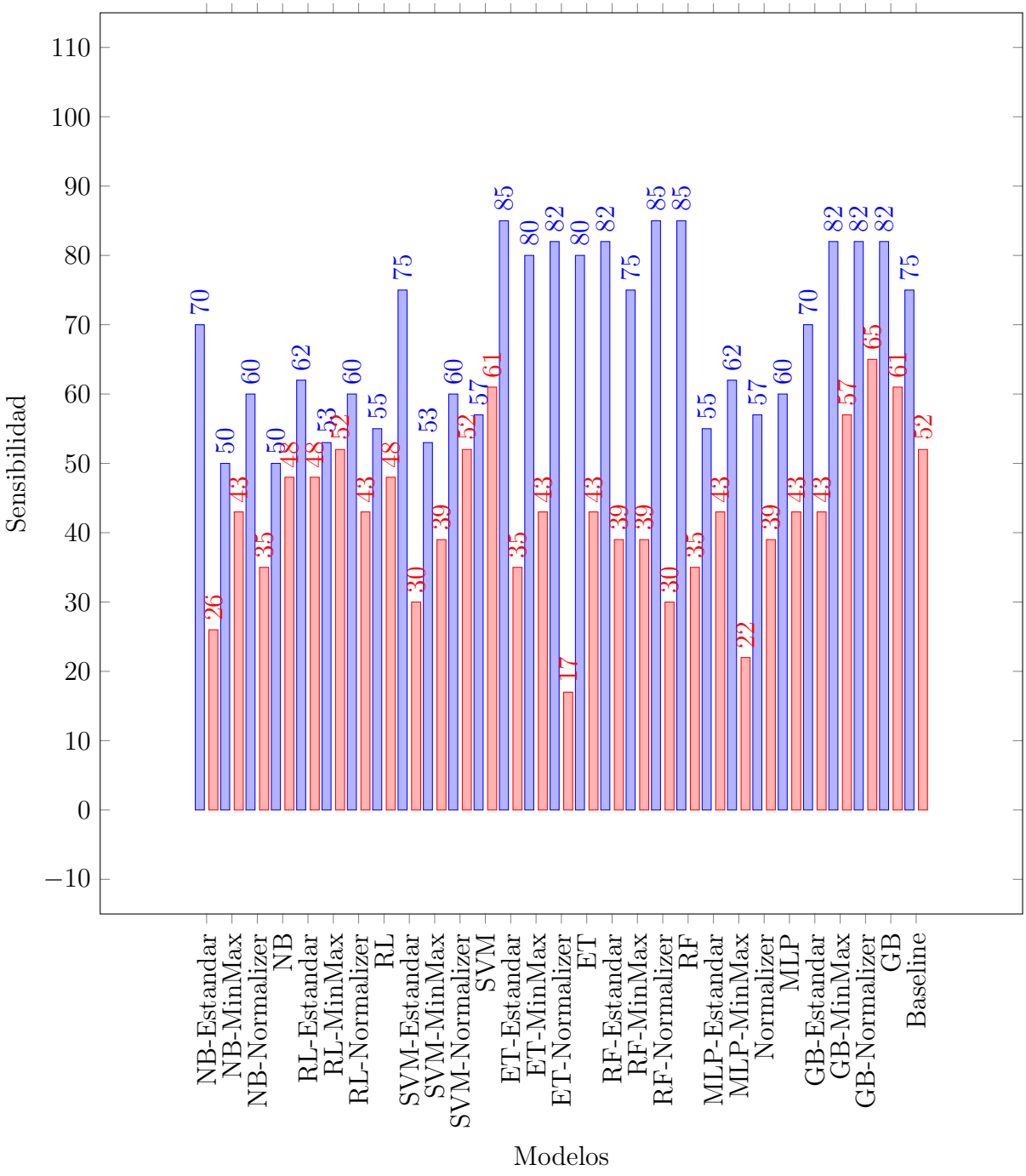


Figura 5.2: Sensibilidad del modelo 2.3

# Referencias


- Definition of Dyslexia - International Dyslexia Association. (2014, September 11). International Dyslexia Association. https://dyslexiaida.org/definition-of-dyslexia/
- Do I Have Dyslexia? - International Dyslexia Association. (2016, March 29). International Dyslexia Association. https://dyslexiaida.org/dyslexia-test/
- Al-Lamki L. Dyslexia: Its impact on the Individual, Parents and Society. Sultan Qaboos Univ Med J. 2012 Aug;12(3):269-72. doi: 10.12816/0003139. Epub 2012 Jul 15. PMID: 23269947; PMCID: PMC3529660.
- En México, siete por ciento de los niños presenta alteraciones del lenguaje. (2018). Unam.mx. https://www.dgcs.unam.mx/boletin/bdboletin/2018_134.html
- Cuetos, F., Ramos, JL y Ruano, E. (2002). PROESC. Evaluación de los procesos de escritura . Madrid: TÉ .
- Cuetos, F., Rodríguez, B., Ruano, E. y Arribas, D. (2007). PROLEC-R: Batería de Evaluación de los Procesos Lectores, Revisada.
- Hulme, C., & Snowling, M. J. (2011). Children's Reading Comprehension Difficulties: Nature, Causes, and Treatments. Current Directions in Psychological Science, 20(3), 139–142. https://doi.org/10.1177/0963721411408673

- Drigas, A. & Politi-Georgousi, S. (2019). ICTs as a Distinct Detection Approach for Dyslexia Screening: A Contemporary View. International Association of Online Engineering. Retrieved August 14, 2023 from https://www.learntechlib.org/p/217923/.
- Lakretz, Y., Chechik, G., Friedmann, N., & Rosen-Zvi, M. (2015, August). Probabilistic graphical models of dyslexia. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1919-1928).
- Detecting readers with dyslexia using machine learning with eye tracking measures | Proceedings of the 12th International Web for All Conference. (2015). ACM Conferences. https://dl.acm.org/doi/abs/10.1145/2745555.2746644
- Rello, L., & Ballesteros, M. (2015, May). Detecting readers with dyslexia using machine learning with eye tracking measures. In Proceedings of the 12th International Web for All Conference (pp. 1-8).
- Cui, Z., Xia, Z., Su, M., Shu, H., & Gong, G. (2016). Disrupted white matter connectivity underlying developmental dyslexia: a machine learning approach. Human brain mapping, 37(4), 1443-1458.
- Rauschenberger, M., Baeza-Yates, R., & Rello, L. (2022). A Universal Screening Tool for Dyslexia by a Web-Game and Machine Learning. Frontiers in Computer Science, 3, 628634.
- Pérez-de-Arrilucea-Cabanillas, I. (2014). La detección temprana de la dislexia y su intervención en educación (Bachelor's thesis).
- Villagra, A. (2022). Dislexia, Detección temprana e impacto emocional.
- British Dyslexia Association. (2023). Assessments - British Dyslexia Association. British Dyslexia Association. https://www.bdadyslexia.org.uk/services/assessments

- Bishop, C. M. (2016). Pattern Recognition and Machine Learning. Springer.
- Kelleher, J. D. (2020). FUNDAMENTALS OF MACHINE LEARNING FOR PREDICTIVE DATA ANALYTICS : algorithms, worked examples, ... and case studies.
- Müller, A. C., & Guido, S. (2017). Introduction to machine learning with Python : a guide for data scientists. O'reilly.
- Andriy Burkov. (2019). THE HUNDRED-PAGE MACHINE LEARNING BOOK. Andriy Burkov.
- Kotsiantis, SB, Kanellopoulos, D. y Pintelas, PE (2006). Preprocesamiento de datos para aprendizaje supervisado. Revista internacional de informática , 1 (2), 111-117.
- Zheng, A., & Casari, A. (2018). Feature engineering for machine learning: Principles and Techniques for Data Scientists. O'Reilly Media.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
- Vanderplas, J. T. (2017). Python data science handbook : essential tools for working with data. O'reilly, Cop.