



Universidad Católica  
**San Pablo**

**FACULTAD DE INGENIERÍA Y COMPUTACIÓN**  
**DEPARTAMENTO DE CIENCIA DE LA**  
**COMPUTACIÓN**

**Escuela Profesional de Ciencia de la**  
**Computación**

**Una herramienta de visualización interactiva**  
**para la generación de ensamblados óptimos de**  
**clasificadores**

**Tesis**

Presentado por el Bachiller:

**Dany Mauro Diaz Espino**

Para Optar por el Título Profesional de:

**Licenciado en Ciencia de la Computación**

**Asesor: Dr. Erick Mauricio Gomez Nieto**

**Arequipa, agosto 2022**

# Abreviaturas

**OCEG** *Optimal Classifier Ensemble Generation*

**SVM** *Support Vector Machine*

**CNB** *Complement Naive Bayes*

**UMAP** *Uniform Manifold Approximation and Projection for Dimension Reduction*

**LAMP** *Local Affine Multidimensional Projection*

**PEP** *Pareto Ensemble Pruning*

**COMEP** *Centralized Objection Maximization for Ensemble Pruning*

# Agradecimientos

---

Deseo agradecer a Dios por haberme guiado a lo largo de estos cinco años de estudio.

Agradezco a mis padres por el apoyo incondicional brindado en todo momento para formarme como un profesional.

Agradezco a la universidad, mi *alma matter*, por haberme cobijado y brindado la formación que ahora me permitirá ayudar a construir una mejor sociedad.

Agradezco de forma muy especial a mi orientador Dr. Erick Mauricio Gómez Nieto por haberme guiado en el proceso de planeamiento, investigación y evaluación de la presente tesis. Sus conocimientos aportados serán de mucha ayuda para mi carrera.

Deseo agradecer al personal administrativo de la universidad: Nercy Rosario Zevallos Gutierrez de Montes. Muchas gracias por la atención brindada y porque siempre estuvo dispuesta a ayudarnos.

# Resumen

---

En la actualidad existen muchos avances en el área de aprendizaje de máquina. A veces los modelos usados para realizar una determinada tarea no proporcionan los mejores resultados. Una alternativa de solución a este problema es el uso de ensambles de clasificadores, ya que la decisión final estará basada en el conjunto de probabilidades generadas por varios modelos. De esta forma, aunque uno de los clasificadores falle, los resultados tendrán cierto grado de confiabilidad si es que los demás clasificadores aciertan. Con este enfoque también surge la demanda por herramientas que permitan realizar un análisis detallado de cada clasificador perteneciente al ensamble. Es por este motivo que la presente tesis propone el desarrollo de una herramienta de generación y comparación de ensambles óptimos en términos de rendimiento. La herramienta propuesta se apoya en técnicas de visualización para realizar dos tareas principales: proporcionar un análisis detallado de cada clasificador y mostrar las métricas asociadas a cada clasificador para ayudar al usuario en la toma de decisiones.

***Palabras Claves*** - Ensamble de clasificadores, visualización interactiva, ciencia de datos.

# Abstract

---

Currently there are many advances in the area of machine learning. Sometimes the models used to perform a certain task do not provide the best results. An alternative solution to this problem is the use of classifier ensembles, because the final decision will be based on the set of probabilities generated by several models. In this way, even if one of the classifiers fails, the results will have a certain degree of reliability if the other classifiers are correct. With this approach, the demand for tools that allow a detailed analysis of each classifier belonging to the ensemble also arises. This is the reason why this thesis proposes the development of a tool for generating and comparing optimal ensembles in terms of performance. The proposed tool is based on visualization techniques to perform two main tasks: to provide a detailed analysis of each classifier and to show the metrics associated with each classifier to help the user in decision making.

**Keywords** - Classifier ensemble, interactive visualization, data science.

# Índice general

<b>1. Introducción</b>	<b>2</b>
1.1. Motivación y Contexto	3
1.2. Planteamiento del Problema	3
1.3. Objetivos	4
1.3.1. Objetivo general	4
1.3.2. Objetivos específicos	4
1.4. Organización del Plan de Tesis	5
<b>2. Marco Teórico</b>	<b>6</b>
2.1. Consideraciones Iniciales	6
2.2. Clasificación	6
2.2.1. Definición	6
2.2.2. Ensamblados de clasificadores	6
2.2.3. Métodos para unir resultados de los miembros de un ensamble	7
2.3. Métricas para evaluar la eficacia	8
2.3.1. Accuracy	8
2.3.2. Precision	8
2.3.3. Recall	8
2.3.4. F1 Score	9
2.3.5. Confusion matrix	9
2.4. Métodos tradicionales de clasificación	10

2.4.1. Redes neuronales	10
2.4.2. Árbol de decisión	10
2.4.3. Bosque aleatorio	10
2.4.4. Regresión logística	11
2.4.5. Support vector machine	12
2.4.6. Complement naive bayes	13
2.5. Métodos de proyección multidimensional	14
2.5.1. Uniform Manifold Approximation and Projection for Dimension Reduction	14
2.5.2. Local Affine Multidimensional Projection	14
2.6. Discretización de coordenadas en un espacio visual	15
2.6.1. Sistema de coordenadas baricéntrico	15
2.6.2. Coordenadas de Wachspres	16
2.7. Optimización de ensambles	16
2.7.1. <i>Pruning</i> de ensambles	16
2.7.2. <i>Pareto Ensemble Pruning</i>	17
2.7.3. <i>Centralized Objection Maximization for Ensemble Pruning</i>	17
2.8. Consideraciones Finales	18
<b>3. Estado del Arte</b>	<b>19</b>
3.1. Consideraciones iniciales	19
3.2. Trabajos relacionados	19
3.2.1. Causality Explorer	19
3.2.2. SeqCausal	19
3.2.3. Prospector	21
3.2.4. Exploración visual de ensambles de clasificadores	21
3.2.5. ICE	22
3.2.6. Boxer	23

---

3.2.7. Visualización de regiones de decisión	23
3.2.8. Visualización del análisis exploratorio de modelos	23
3.2.9. ComDia+	25
3.2.10. Integración visual de conjuntos de datos y modelos	25
3.3. Consideraciones Finales	26
<b>4. OCEG: Una herramienta de visualización interactiva para la generación de ensambles óptimos de clasificadores.</b>	<b>27</b>
4.1. Consideraciones iniciales	27
4.2. Análisis y diseño	27
4.2.1. Objetivos	27
4.2.2. Requerimientos	28
4.2.3. Tareas analíticas	28
4.3. OCEG	29
4.3.1. Componentes Visuales	29
4.3.2. Análisis de componentes visuales y tareas analíticas	33
4.3.3. Detalles de implementación	34
4.4. Consideraciones finales	34
<b>5. Pruebas y Resultados</b>	<b>35</b>
5.1. Consideraciones iniciales	35
5.2. Configuración de experimentos	35
5.3. Evaluación cuantitativa	35
5.3.1. Descripción del experimento	35
5.3.2. Discusión de los resultados	37
5.4. Estudios de caso DS16 y DS17	37
5.4.1. Estudio de caso DS16: Predecir si un paciente tiene COVID-19	38
5.4.2. Estudio de caso DS17: Predecir si una persona pagará o no su deuda de tarjeta de crédito	39

---

5.5. Consideraciones finales . . . . .	41
<b>6. Conclusiones y Trabajos Futuros</b>	<b>42</b>
6.1. Problemas Encontrados . . . . .	42
6.2. Recomendaciones . . . . .	43
6.3. Trabajos Futuros . . . . .	43
<b>Bibliografía</b>	<b>46</b>

# Índice de figuras

2.1. Ejemplo de <i>confusion matrix</i> generada por la herramienta propuesta, con un conjunto de 10 clases y usando un clasificador del tipo <i>árbol de decisión</i> . Imagen producida por el autor. . . . .	9
2.2. Estructura de un <i>árbol de decisión</i> . Imagen extraída de [Jijo and Mohsin Abdulazeez, 2021].	11
2.3. Estructura de un <i>bosque aleatorio</i> . Imagen extraída de [Denisko and Hoffman, 2018]	11
2.4. Ejemplo de <i>Support Vector Machine (SVM)</i> con kernel polinomial. Imagen extraída de [Géron, 2019]. . . . .	12
2.5. Ejemplo de coordenadas baricéntricas, el triángulo izquierdo se balancea hacia el lado derecho porque no posee masas (coeficientes), mientras el derecho sí está balanceado sobre el centro de masa P. Imagen producida por el autor . . . . .	15
2.6. Ejemplo de coordenadas coordenadas de <i>Wachspress</i> sobre un polígono de 6 lados. Para calcular el coeficiente del punto A, se divide el área del triángulo FAB entre el producto de los triángulos FAP y ABP. Imagen producida por el autor. . . . .	16
3.1. Componentes del sistema <i>Causality Explorer</i> . Imagen extraída de [Xie et al., 2021]. . . . .	20
3.2. Sistema <i>SeqCausal</i> . Imagen extraída de [Jin et al., 2020]. . . . .	20
3.3. Sistema <i>Prospector</i> . Imagen extraída de [Krause et al., 2016]. . . . .	21
3.4. Vista general de la interfaz del trabajo de [Ribeiro et al., 2019]. Imagen extraída de [Ribeiro et al., 2019]. . . . .	22
3.5. Vista general de la herramienta ICE. Imagen extraída de [Castro-Ochante et al., 2021]. . . . .	22
3.6. Vista de los componentes de la herramienta Boxer. Imagen extraída de [Gleicher et al., 2020]. . . . .	23

---

3.7. Ejemplo de proyecciones y mapas densos generados usando la propuesta de [Rodrigues et al., 2018]. Imagen extraída de [Rodrigues et al., 2018]. . . . .	24
3.8. Flujo de trabajo propuesto por [Cashman et al., 2019]. Imagen extraída de [Cashman et al., 2019]. . . . .	24
3.9. Vista general de ComDia+. Imagen extraída de [Park et al., 2019]. . . . .	25
3.10. Vista general de la herramienta propuesta por [Schneider et al., 2017] para la exploración de ensamblajes de clasificadores. Imagen extraída de [Schneider et al., 2017]. . . . .	26
4.1. Vista general de la herramienta: (a) vista de clasificadores, (b) vista de generación de ensamble, (c) vista de selección de pesos, (d) vista de <i>pruning</i> y (e) vista de comparación. Imagen producida por el autor. . . . .	29
4.2. Ejemplo de discretización del espacio 2d. Imagen producida por el autor. . . . .	30
4.3. Coloreo del espacio discretizado en 2d. Imagen producida por el autor. . . . .	30
4.4. Coloreo de las regiones del espacio 2d. Imagen producida por el autor. . . . .	30
4.5. Colocación de los elementos del conjunto de prueba en las regiones de decisión. Imagen producida por el autor. . . . .	31
4.6. Coloreo de los elementos del conjunto de prueba en las regiones de decisión. Imagen producida por el autor. . . . .	31
4.7. Procedimiento de coloreo de polígono. Imagen producida por el autor. . . . .	32
4.8. Asignación de pesos. Imagen producida por el autor. . . . .	33
5.1. Resumen del primer estudio de caso: (a) cargado e inspección de modelos, (b) creación del primer ensamble, (c-e) asignación de pesos y optimización de ensamblajes, y (f) exportar el mejor ensamble. Imagen producida por el autor. . . . .	40
5.2. Proyección de los clasificadores individuales y ensamblajes en la vista de comparación para el segundo estudio de caso. Imagen producida por el autor. . . . .	41

# Capítulo 1

## Introducción

Actualmente las tareas de clasificación llevadas a cabo sobre los crecientes volúmenes de datos se vuelven cada vez más complejas. Esto se debe a varios factores, entre los cuales se incluyen: la gran cantidad de atributos que poseen los conjuntos de datos, la existencia de pocos mecanismos para entender el comportamiento de los clasificadores entrenados con los conjuntos de datos y la poca variedad de métodos de comparación entre los clasificadores [Ribeiro et al., 2019]. Por otro lado, se ha visto un creciente interés en el desarrollo de nuevas técnicas de visualización, por lo que resulta factible buscar la forma de aplicar estos conocimientos para resolver las limitaciones actuales [Yuan et al., 2021]. Gracias al desarrollo de nuevas técnicas de visualización de datos es posible crear potentes herramientas para la toma de decisiones, las cuales no sólo mejoran los aspectos mencionados anteriormente, sino que también permiten obtener resultados con mayor eficiencia [Castro-Ochante et al., 2021].

En el área de clasificación, que pertenece al dominio de aprendizaje de máquina, a veces un solo clasificador no proporciona los mejores resultados para un determinado conjunto de datos. Es por este motivo que, en muchos trabajos recientes, se usan diversos clasificadores de forma conjunta para abordar una misma tarea [Sagi and Rokach, 2018]. Estas agrupaciones de clasificadores, conocidas como ensambles, son capaces de proporcionar mejores resultados si se configuran adecuadamente [Schneider et al., 2017], lo cual se pudo demostrar en el presente trabajo. Para lograr una correcta configuración, es necesario brindar la flexibilidad suficiente al usuario en la decisión de qué clasificadores usar y en qué medida el resultado individual de cada clasificador debe contribuir al resultado final. Estas decisiones pueden tomarse en base al criterio personal del usuario o a la métrica que desea maximizar [Chatzimparmpas et al., 2021].

Adicionalmente para el desarrollo de herramientas de visualización útiles se deben considerar diversos factores de diseño: facilidad de uso, volumen de información presentada al usuario, mecanismos de interacción entre usuario y herramienta, etc [Gleicher et al., 2020]. De los factores mencionados, el de mayor importancia es quizá la facilidad de uso, lo cual se evidencia en la mayoría trabajos analizados en el Capítulo 3 de la presente tesis, pues hacen énfasis en este aspecto.

---

## 1.1. Motivación y Contexto

La importancia de usar ensambles de clasificadores radica en que es posible alcanzar un mejor desempeño al usar varios clasificadores trabajando en simultáneo en vez de usar un clasificador individual para una tarea determinada. Sin embargo, esta potencial mejora está condicionada a la correcta configuración del ensamble, ya sea a través de una selección óptima de clasificadores para ser miembros del ensamble y de una buena asignación de pesos para cada miembro. Este es el principal motivo para la realización de un trabajo de generación, evaluación y optimización de ensambles de clasificadores. Cada clasificador individual posee ciertas ventajas y desventajas, por lo que la herramienta propuesta debe ser capaz de determinar cuáles son los mejores clasificadores para cada caso y guiar al usuario en la creación del ensamble.

El creciente avance en técnicas de visualización, interacción y proyección multidimensional es otro motivo importante para la realización de la herramienta propuesta en esta tesis. Varias de las propuestas analizadas en el Capítulo 2 de ésta tesis lograron incorporar, de forma exitosa, técnicas de visualización a los mecanismos tradicionales del aprendizaje de máquina, obteniendo opiniones favorables por parte de los usuarios que evaluaron las propuestas. Este éxito en la combinación de técnicas de visualización y de aprendizaje de máquina sienta un precedente como inspiración para futuros trabajos que sigan temáticas similares. La presente tesis se fundamenta en el éxito demostrado por las propuestas analizadas, pero enfocándose en un tema muy poco mencionado en la literatura actual: la generación visual e interactiva de ensambles de clasificadores.

Es por los motivos anteriormente mencionados que se justifica la inclusión de técnicas de visualización e interacción para los procedimientos usados en el área de aprendizaje de máquina. Una metodología que incorpore técnicas de visualización e interacción permitirá acelerar y facilitar los procesos que anteriormente se consideraban tediosos y extensos, lo cual queda demostrado por los trabajos analizados en el estado del arte de la presente tesis.

## 1.2. Planteamiento del Problema

Si bien es cierto que es posible obtener el mejor clasificador por medio de diversos métodos (como uso de ensambles, optimización de parámetros o selección de atributos dentro del conjunto de datos), el procedimiento usado para obtener los mejores resultados a menudo resulta complicado y extenso. En el caso de ensambles, el procedimiento para obtener el mejor ensamble consiste en experimentos de ensayo y error hasta encontrar una solución que cumpla las expectativas del usuario [Castro-Ochante et al., 2021]. La falta de eficiencia de este proceso es uno de los principales problemas que se abordarán en la presente tesis.

Otro aspecto a tomar en cuenta al momento de usar ensambles es que, en muchos casos, los ensambles creados contienen demasiados miembros, lo cual hace que su desempeño sea lento. Esto se puede corregir a través de algoritmos de optimización de ensambles, los cuales apuntan a reducir el número de clasificadores miembros de un ensamble sin perder

mucha exactitud en sus predicciones [Sagi and Rokach, 2018]. En la presente tesis se toma en cuenta esta necesidad de optimización y se aplica al flujo de trabajo propuesto.

Un problema recurrente en varios de los trabajos actuales es la escalabilidad. Desarrollar una herramienta escalable de visualización es un desafío persistente en la actualidad [Xie et al., 2021]. Los autores de los trabajos analizados en el Capítulo 3 de la presente tesis mencionan que al trabajar con conjuntos de datos reales, que poseen una alta cantidad de datos de alta dimensionalidad, los métodos descritos en sus trabajos no presentan las soluciones óptimas. Por ese motivo consideran abordar este problema en sus próximos trabajos.

Basado en los antecedentes mencionados, el problema a abordar por este trabajo se centra en generar una nueva metodología que combine estrategias automáticas e interactivas, con la finalidad de extender la capacidad examinadora de un analista para la generación de ensambles de clasificación.

## 1.3. Objetivos

### 1.3.1. Objetivo general

El objetivo de esta tesis es la creación de una herramienta interactiva que permita generar diversos ensambles óptimos de clasificadores.

### 1.3.2. Objetivos específicos

- Seleccionar clasificadores de un conjunto de opciones para poder generar un ensamble inicial. El analista podrá generar un ensamble seleccionando por lo menos dos clasificadores del conjunto de opciones.
- Permitir la modificación de los ensambles generados ya sea añadiendo o eliminando clasificadores. La herramienta permitirá que el usuario seleccione de la lista de opciones cuáles clasificadores agregará al ensamble y cuáles clasificadores quitará del ensamble.
- Permitir la modificación del peso asociado a cada clasificador. La asignación de pesos para cada miembro del ensamble se hará a través de una gráfica interactiva.
- Realizar una optimización para reducir el número de clasificadores que componen el ensamble generado. El analista podrá usar un algoritmo de pruning para la optimización de los ensambles generados.
- Mostrar los resultados y métricas asociadas a cada ensamble de clasificadores de forma visual. La herramienta generará una gráfica de comparación usando algoritmos de proyección multidimensional.

---

## 1.4. Organización del Plan de Tesis

En el Capítulo 2 se explican los conceptos necesarios para el completo entendimiento del problema que se desea resolver. Este capítulo es el marco teórico de la tesis.

En el Capítulo 3 se analizan diversos trabajos similares al de esta tesis, para tener una idea general del estado del arte en esta área.

En el Capítulo 4 se encuentra la propuesta de la tesis, a la vez que se describe la metodología usada para resolver el problema propuesto.

En el Capítulo 5 se detallan las evaluaciones hechas a la propuesta y los resultados obtenidos.

En el Capítulo 6 se presentan las conclusiones obtenidas y los trabajos futuros de la propuesta.

# Capítulo 2

## Marco Teórico

### 2.1. Consideraciones Iniciales

En este capítulo se describirán brevemente todas las técnicas usadas para el presente trabajo, de modo que pueda facilitarse su comprensión.

### 2.2. Clasificación

#### 2.2.1. Definición

Una de las tareas más comunes dentro del área de aprendizaje de máquina (incluida dentro de la inteligencia artificial, cuyo objetivo es generar modelos que 'aprendan' a realizar diversas tareas) es la de clasificación. Clasificar es sinónimo de predecir la clase a la que pertenece un dato determinado. Se encuentra en la categoría de aprendizaje supervisado, ya que para el entrenamiento del modelo se usan datos cuyas clases ya son conocidas de antemano. La clasificación se aplica a una amplia gama de problemas, como la detección de spam, el reconocimiento de escritura a mano, el diagnóstico médico, la clasificación de documentos y las tareas de detección de imágenes, por nombrar algunos [Ribeiro et al., 2019](#).

#### 2.2.2. Ensamblados de clasificadores

En la mayoría de casos, dejar la tarea de clasificación a un solo modelo produce buenos resultados. Sin embargo, se ha visto una creciente tendencia a usar más de un modelo para realizar una tarea determinada. A este uso en conjunto de varios modelos se le denomina ensamble. En otras palabras, un ensamble es un conjunto de modelos individuales trabajando en paralelo. Los resultados obtenidos por cada modelo son combinados usando una función de fusión, la cual produce la respuesta a un problema determinado. Una forma de definir el grado de 'importancia' que se debe asignar a cada algoritmo es

colocarle un peso dentro del ensamble [Castro-Ochante et al., 2021].

Formalmente, un ensamble de clasificadores se puede definir según la ecuación 2.1 [Sagi and Rokach, 2018]:

Dado un conjunto de datos de  $n$  instancias y  $m$  atributos:

$$D = \{(x_i, y_i)\} (|D| = n, x_i \in R^m, y_i \in R) \quad (2.1)$$

Para combinar los resultados de los clasificadores individuales, un ensamble  $\phi$  usa una función de agregación  $G$  que incorpora  $K$  modelos  $\{f_1, f_2, \dots, f_k\}$ , definida usando la ecuación 2.2.

$$y_i = \phi(x_i) = G(f_1, f_2, \dots, f_k) \quad (2.2)$$

### 2.2.3. Métodos para unir resultados de los miembros de un ensamble

Existen varios métodos para fusionar los resultados obtenidos por los miembros de un ensamble en una sola predicción. En el presente trabajo usamos dos. El primero, llamado *hard voting*, consiste en contar las predicciones individuales realizadas por cada modelo y dar como resultado la clase más frecuente de estos resultados individuales. El segundo, llamado *soft voting*, consiste en obtener un promedio de todas las probabilidades generadas por cada modelo individual y dar como resultado la clase con la mayor probabilidad obtenida [Géron, 2019].

El uso de estas técnicas puede producir resultados diferentes. Por ejemplo, supongamos que tenemos un conjunto de datos que contiene dos clases, llamadas A y B, y tenemos un ensamble de tres clasificadores que producen los siguientes resultados para un dato individual:

- Clasificador 1: Probabilidades: 80 % clase A, 20 % clase B. Resultado: clase A.
- Clasificador 2: Probabilidades: 45 % clase A, 55 % clase B. Resultado: clase B.
- Clasificador 3: Probabilidades: 45 % clase A, 55 % clase B. Resultado: clase B.

Si el ensamble usa *hard voting*, se obtendría como resultado la clase B, debido a que la mayoría de clasificadores votaron por esta clase (los clasificadores 2 y 3). Por otro lado, si el ensamble usa *soft voting* entonces se obtendría como resultado la clase A debido a que el promedio obtenido para la clase A es mayor al promedio obtenido para la clase B.

$$\begin{aligned} (80\% + 45\% + 45\%) \div 3 &= 56,66\% \text{ para la clase A} \\ (20\% + 55\% + 55\%) \div 3 &= 43,33\% \text{ para la clase B} \end{aligned}$$

Una de las ventajas del algoritmo *soft voting* frente a *hard voting* es que se también se toma en cuenta la incertidumbre de cada clasificador en la decisión final. En el ejemplo anterior, la alta incertidumbre en los clasificadores 2 y 3 hizo que la decisión final esté basada en mayor medida en el clasificador 1.

## 2.3. Métricas para evaluar la eficacia

Existen diferentes métricas que nos permiten tener una idea general de la eficacia de un clasificador. Las más usadas son: *accuracy*, *precision*, *recall*, *F1* y *confusion matrix* [Géron, 2019]. Estas métricas dependen de los siguientes valores:

- Verdaderos positivos (VP): Valores a los cuales el clasificador indica que pertenecen a una clase cuando efectivamente sí pertenecen a esa clase.
- Falsos positivos (FP): Valores a los cuales el clasificador indica que pertenecen a una clase cuando en realidad no pertenecen a esa clase.
- Falsos negativos (FN): Valores a los cuales el clasificador indica que no pertenecen a una clase cuando en realidad sí pertenecen a esa clase.
- Verdaderos negativos (VN): Valores a los cuales el clasificador indica que no pertenecen a una clase cuando efectivamente no pertenecen a esa clase.

### 2.3.1. Accuracy

Es el ratio de predicciones correctas con respecto al total de predicciones [Géron, 2019]. Se calcula usando la ecuación 2.3.

$$Accuracy = \frac{VP + VN}{VP + FP + FN + VN} \quad (2.3)$$

### 2.3.2. Precision

Es definida como la precisión de las predicciones positivas [Géron, 2019]. Se calcula usando la ecuación 2.4.

$$Precision = \frac{VP}{VP + FP} \quad (2.4)$$

### 2.3.3. Recall

Es definida como el ratio de instancias positivas que fueron detectadas correctamente por el clasificador [Géron, 2019]. Se calcula usando la ecuación 2.5.

$$Recall = \frac{VP}{VP + FN} \quad (2.5)$$

### 2.3.4. F1 Score

Proporciona una forma más simple de comparar dos clasificadores. Es definida como la media armónica de *precision* y *recall* que da más peso a los valores bajos. Es por este motivo que un clasificador obtendrá un valor *F1* alto sólo si *precision* y *recall* tienen valores altos [Géron, 2019]. Se calcula usando la ecuación 2.6.

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{2.6}$$

### 2.3.5. Confusion matrix

Esta matriz proporciona información sobre el ratio de elementos clasificados tanto de forma correcta e incorrecta de todas las clases. Cada fila representa una clase real, mientras que cada columna representa una clase predicha. También se pueden mostrar los valores normalizados (al dividir la cantidad de elementos clasificados por clase entre el total de elementos) en lugar de mostrar la cantidad real de predicciones hechas. Idealmente, sólo en la diagonal de la matriz se deberían encontrar los valores diferentes de cero [Luque et al., 2019].

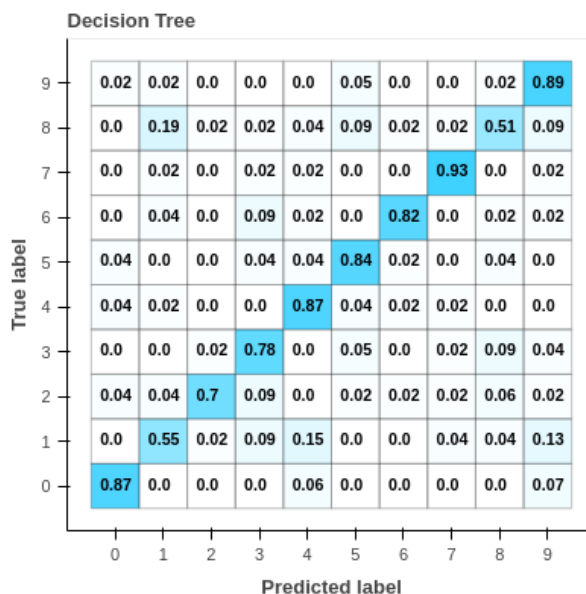


Figura 2.1: Ejemplo de *confusion matrix* generada por la herramienta propuesta, con un conjunto de 10 clases y usando un clasificador del tipo *árbol de decisión*. Imagen producida por el autor.

---

## 2.4. Métodos tradicionales de clasificación

### 2.4.1. Redes neuronales

Son modelos que actualmente gozan de una amplia popularidad pues son útiles para la clasificación, agrupamiento, reconocimiento de patrones y predicción en muchas disciplinas. Se componen principalmente de unidades llamadas neuronas que se encuentran conectadas unas con otras a través de unos enlaces [Elsheikh et al., 2019]. Cada neurona realiza un procesamiento matemático. El valor de entrada de cada neurona es determinado por el valor de salida de la neurona anterior multiplicado por un valor de peso. Debido a los avances en modelos de *redes neuronales* es que se han vuelto relativamente competitivo con los modelos estadísticos y de regresión convencionales, en el ámbito de la utilidad. El gran potencial de las *redes neuronales* es el procesamiento de alta velocidad, que se debe a que son implementados de forma paralela masiva y esto, a su vez, ha aumentado la necesidad de investigación en este dominio. Dentro de la amplia gama de tareas que pueden realizar se incluyen: reconocimiento de imágenes, el procesamiento del lenguaje natural, etc [Abiodun et al., 2018].

### 2.4.2. Árbol de decisión

Este tipo de modelos contienen reglas conceptuales que son mucho más fáciles de construir, en comparación con los pesos numéricos en las *redes neuronales* de conexiones entre nodos. Principalmente son usados para fines de agrupación y minería de datos. Este modelo se compone por nodos que representan características en una categoría para ser clasificado. Los datos son procesados en cada nodo y luego de verificar ciertas reglas son derivados hacia un nodo inferior correspondiente. Este proceso se realiza hasta alcanzar un nodo hoja, que contiene la información de la clase a la cual pertenece el dato procesado [Patel and Prajapati, 2018]. Debido a su análisis simple y su precisión en múltiples conjuntos de datos, los árboles de decisión se implementan como métodos de solución en una amplia gama de tareas [Jijo and Mohsin Abdulazeez, 2021].

### 2.4.3. Bosque aleatorio

Este tipo de modelos consisten en conjuntos de varios *árboles de decisión*. Desde su apogeo en 2001, han demostrado ser útiles en muchos campos debido a que poseen una alta precisión. Por ejemplo, en biología y medicina, los *bosques aleatorios* han logrado abordar con éxito una variedad de problemas, incluida la predicción de la respuesta a los medicamentos, la identificación de proteínas en el ADN y la localización del cáncer en tejidos particulares. Estos modelos también han sido usados con éxito en tareas de reconocimiento del habla y de dígitos escritos a mano. A diferencia de un *árbol de decisión*, en este tipo de modelos se debe seleccionar el mejor atributo del conjunto de datos para dividir en cada nodo de un subconjunto aleatorio de los atributos disponibles disponibles. Esta selección aleatoria permite que los árboles de decisión individuales de un *bosque aleatorio* se enfoquen en diferentes atributos y no sólo en uno. Esta diversidad resultante

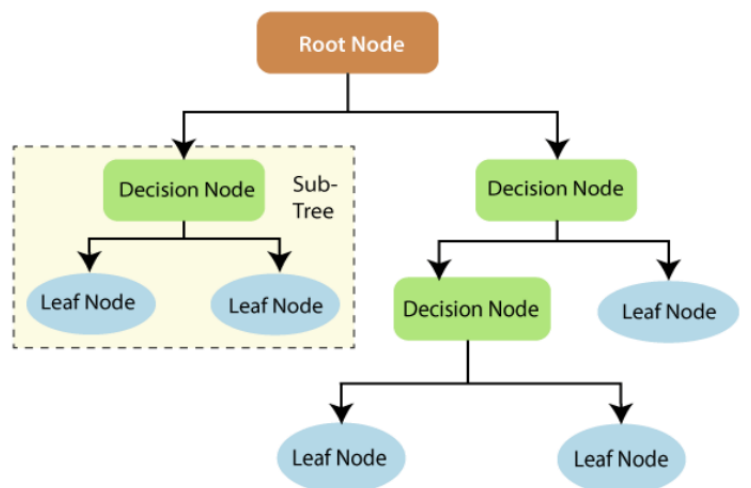


Figura 2.2: Estructura de un árbol de decisión. Imagen extraída de [Jijo and Mohsin Abdulazeez, 2021].

de árboles puede reducir la posibilidad de sobreajuste a los datos de entrenamiento. De esta forma, el *bosque aleatorio* mejora la precisión [Denisko and Hoffman, 2018].

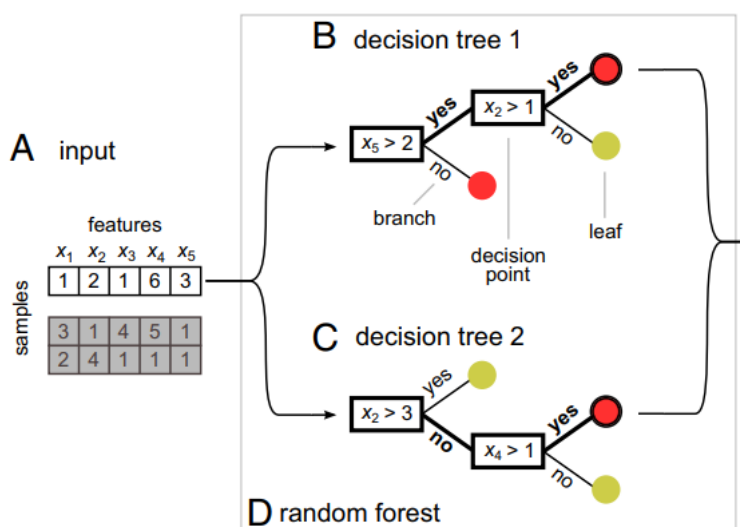


Figura 2.3: Estructura de un bosque aleatorio. Imagen extraída de [Denisko and Hoffman, 2018].

### 2.4.4. Regresión logística

Este modelo es similar a una *regresión lineal*. Ambos modelos realizan una suma ponderada de las variables de entrada acorde a la ecuación 2.7, donde  $y$  es el valor predicho.

$$y = \theta_0 + \theta_1x_1 + \theta_2x_2 + \dots + \theta_nx_n \tag{2.7}$$

La diferencia radica es que en la *regresión lineal* genera el resultado directamente mientras que la *regresión logística* genera las probabilidades de pertenecer a cada clase a través del uso de una función sigmoide, la cual se define usando la ecuación 2.8

$$\theta(t) = \frac{1}{1 - \exp(-t)} \quad (2.8)$$

Obteniendo la función de resultados [2.9](#), basada en la probabilidad estimada.

$$y = \begin{cases} 0 & \text{if } p < 0,5 \\ 1 & \text{if } p \geq 0,5 \end{cases} \quad (2.9)$$

Por último, el entrenamiento se enfoca en obtener un vector  $\theta$  optimizado de tal forma que al realizar la suma ponderada, se estimen altas probabilidades para instancias positivas y bajas probabilidades para las instancias negativas [Géron, 2019](#).

### 2.4.5. Support vector machine

Son modelos de aprendizaje de máquina adecuados para la tarea de clasificación en conjuntos de datos complejos pero de tamaño pequeño o mediano. La idea principal es generar un hiperplano que pueda separar los datos pertenecientes a clases distintas de forma óptima, maximizando la distancia entre el hiperplano y los puntos que estén más cercanos a él. Esta búsqueda de la distancia máxima posible es llamada *hard margin classification* [Jahed Armaghani et al., 2020](#).

El principal problema es que este enfoque puede no dar buenos resultados si los datos no son linealmente separables. La mejor forma de solucionar esto es encontrar un balance entre la distancia del hiperplano a los puntos y la penalización por las violaciones al límite definido por el hiperplano (que se dan cuando los puntos intersectan el hiperplano o son clasificados en el lado incorrecto del hiperplano). A este nuevo enfoque flexible se le conoce como *soft margin classification* [Géron, 2019](#).

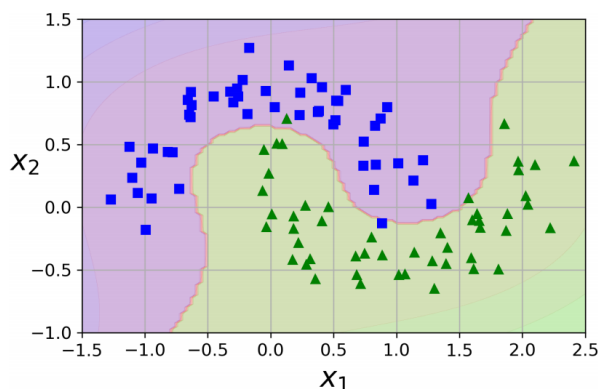


Figura 2.4: Ejemplo de [SVM](#) con kernel polinomial. Imagen extraída de [Géron, 2019](#).

### 2.4.6. Complement naive bayes

El clasificador *Complement Naive Bayes* (CNB) es una mejora del modelo *naive bayes*. Este clasificador no es tan eficaz como los clasificadores del estado del arte, sin embargo es usado por su rapidez y su facilidad de implementación. Los algoritmos más eficientes tienden a ser más complejos y lentos. Se fundamenta en el teorema de Bayes, descrito en la ecuación 2.10, que define la probabilidad de que ocurra un evento aleatorio dada la ocurrencia de otro evento.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2.10)$$

$P(A|B)$  es la probabilidad de A sea verdadero dado que B es verdadero.

$P(B|A)$  es la probabilidad de B sea verdadero dado que A es verdadero.

$P(A)$  es la probabilidad de que A sea verdadero.

$P(B)$  es la probabilidad de que B sea verdadero.

Este teorema, en conjunto con la suposición de que cada una de las variables contribuye de forma independiente a la predicción de una clase, conforman las bases para el clasificador. La clasificación se realiza usando la fórmula 2.11

$$\text{classify}(f_1, \dots, f_n) = \underset{c}{\operatorname{argmax}} p(C = c) \prod_{i=1}^n p(F_i = f_i | C = c). \quad (2.11)$$

Donde se obtiene el argumento (en este caso la clase) que maximiza la función de probabilidad para que el vector de atributos  $(f_1, \dots, f_n)$  pertenezca a la clase  $C$ . Esto se realiza multiplicando las probabilidades de cada elemento dado que la clase es  $C$ .

El principal problema surge cuando existe un número desigual de muestras por cada clase en el conjunto de entrenamiento. Esto puede provocar que los pesos de los límites de decisión estén sesgados, causando que el clasificador asigne más probabilidades a la clase con más muestras en el conjunto de entrenamiento.

Para solucionar este problema, se usa una función de estimación del tipo 'uno contra todos menos uno', que asume que cada dato puede tener más de una clase. Es por esto que éste clasificador estima los pesos usando datos de todas las clases. Esta heurística genera estimaciones más efectivas, a diferencia del modelo *naive bayes*, que estima pesos usando datos de una sola clase [Rennie et al., 2003].

---

## 2.5. Métodos de proyección multidimensional

### 2.5.1. Uniform Manifold Approximation and Projection for Dimension Reduction

*Uniform Manifold Approximation and Projection for Dimension Reduction* (**UMAP**) es una técnica de aprendizaje múltiple que tiene como objetivo la reducción de dimensiones. Se puede describir como un algoritmo para la creación y operación de grafos dirigidos.

El procedimiento se divide en dos fases:

- **Creación de grafo:** Primero se crea un grafo ponderado que contiene los  $k$  vecinos de un dato. Luego se aplica una transformación en las aristas para obtener la distancia local entre los nodos. Por último se debe preservar la asimetría inherente del gráfico de  $k$  vecinos.
- **Diseño de grafo:** Se debe definir una función que preserve las características deseadas de este grafo con los  $k$  vecinos. Una vez formulada la función, el último paso es encontrar una representación de baja dimensionalidad que optimice esta función objetivo.

El algoritmo asume que los siguientes axiomas son verdaderos:

- Existe una variedad (objeto que permite representar un espacio de  $n$  dimensiones parecido localmente al conjunto  $R^n$ ) en la que los datos se distribuirían uniformemente.
- La variedad de interés está conectada localmente.
- El objetivo principal es preservar la estructura topológica de esta variedad.

La información detallada del algoritmo creado por **McInnes et al., 2018** así como su implementación completa se pueden encontrar en **[9]**. Esta técnica de reducción de dimensiones es usada en el presente trabajo para proporcionar una visualización en un espacio de dos dimensiones de los conjuntos de datos y las regiones de decisión de los modelos.

### 2.5.2. Local Affine Multidimensional Projection

El algoritmo *Local Affine Multidimensional Projection* (**LAMP**) surge como método de visualización enfocado a aplicaciones completamente interactivas. Este algoritmo busca solucionar el problema que poseen la mayoría de los métodos de proyección multidimensional, ya que hacen uso de un solo mapeo global para proyectar todas las instancias de datos a un espacio visual más simple. Este carácter global hace difícil la experiencia del usuario al impedir que se produzcan ajustes locales, ya que, los cambios locales afectan

a la proyección en su conjunto. Se basa en la teoría del mapeo ortogonal para construir transformaciones precisas de conjuntos de datos que pueden modificarse dinámicamente acorde al conocimiento del usuario.

Este algoritmo usa un subconjunto de puntos, llamados puntos de control, para generar mapeos afines ortogonales para cada instancia a proyectar. Encontrar estos mapeos se reduce a un problema de minimización conocido como *Problema de Procrustes ortogonal*.

Los autores mencionan que se producen mejores resultados cuando se normalizan los datos en el espacio original y la posición de los puntos de control en el espacio visual. Aunque no es una limitación seria, se debe tener en consideración al momento de implementar *LAMP* [Joia et al., 2011].

## 2.6. Discretización de coordenadas en un espacio visual

### 2.6.1. Sistema de coordenadas baricéntrico

Es un sistema de coordenadas en el que la ubicación de un punto  $P$  dentro de un triángulo  $ABC$  se representa según los coeficientes  $\alpha$ ,  $\beta$  y  $\gamma$  asignados a cada vértice  $A$ ,  $B$  y  $C$ .

Se pueden interpretar de la siguiente forma: para equilibrar este triángulo  $ABC$  sin masa en su interior sobre un fulcro en un punto  $P$ , debemos colocar masas en cada vértice ( $\alpha$ ,  $\beta$  y  $\gamma$ ) según la proximidad del punto  $P$  a cada vértice [Lidberg, 2011].

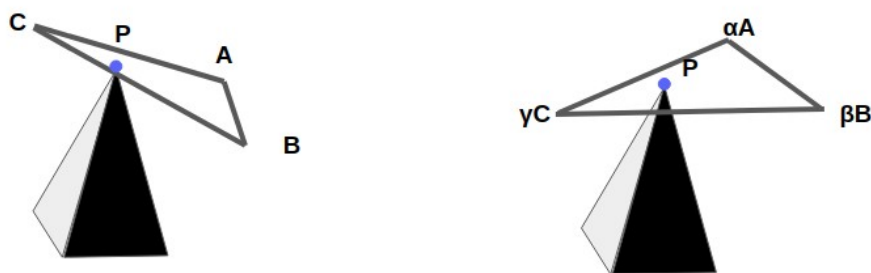


Figura 2.5: Ejemplo de coordenadas baricéntricas, el triángulo izquierdo se balancea hacia el lado derecho porque no posee masas (coeficientes), mientras el derecho sí está balanceado sobre el centro de masa  $P$ . Imagen producida por el autor

Entonces, este punto se puede representar como  $P = \alpha A + \beta B + \gamma C$

## 2.6.2. Coordenadas de Wachspress

Este sistema una extensión de las coordenadas baricéntricas pero aplicadas sobre polígonos convexos con un número de vértices mayor a igual a 3 [Floater, 2014]. El primer paso es formar un polígono colocando los vértices deseados en sentido antihorario. Luego se debe usar la fórmula de Wachspress para calcular los coeficientes asociados a cada vértice, calculando el área de los triángulos usando el vértice anterior, posterior y el punto de referencia P [Lidberg, 2011]. En la fórmula 2.12, la función  $A(X,Y,Z)$  representa el área del triángulo XYZ.

$$\gamma(P_i) = \frac{A(V_{i-1}, V_i, V_{i+1})}{A(V_{i-1}, V_i, P) * A(V_i, V_{i+1}, P)} \quad (2.12)$$

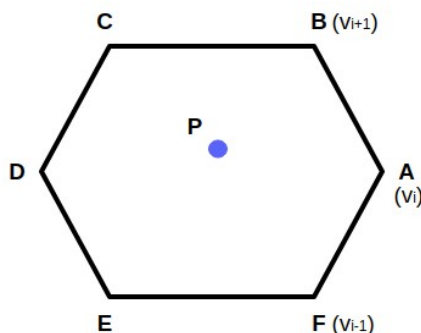


Figura 2.6: Ejemplo de coordenadas de *Wachspress* sobre un polígono de 6 lados. Para calcular el coeficiente del punto A, se divide el área del triángulo FAB entre el producto de los triángulos FAP y ABP. Imagen producida por el autor.

Este sistema de coordenadas será útil para la asignación de pesos en nuestra herramienta propuesta.

## 2.7. Optimización de ensamblajes

### 2.7.1. *Pruning* de ensamblajes

Son métodos usados para reducir el número de clasificadores miembros de un ensamblaje. Al reducir el número de clasificadores se puede ahorrar espacio de almacenamiento y acelerar la velocidad de predicción. El *pruning* de ensamblajes tiene dos objetivos específicos: maximizar (o mantener) el desempeño del ensamblaje y minimizar el número de clasificadores que componen el ensamblaje [Qian et al., 2015]. En algunos casos estos objetivos son contradictorios, ya que al reducir el número de clasificadores se puede obtener un desempeño menor. Es por este motivo que existen diferentes algoritmos y heurísticas para reducir el número de clasificadores de forma óptima, evitando la disminución del desempeño del ensamblaje.

### 2.7.2. *Pareto Ensemble Pruning*

*Pareto Ensemble Pruning* (PEP), es un algoritmo evolutivo de optimización que formula el problema de optimizar un ensemble como una función bi-objetiva [Qian et al., 2015]. El algoritmo consiste en definir una función bi-objetiva  $g(s) = (f(Hs), |s|)$ . Donde  $f(Hs)$  corresponde al error obtenido luego de evaluar el conjunto de validación y  $|s|$  corresponde al número de clasificadores que componen el ensemble. El algoritmo usa cadenas de unos y ceros para representar los ensembles generados. De esta forma, si se tiene un conjunto de clasificadores  $H = \{h1, h2, h3, h4, h5\}$ , la cadena  $S = 10110$  representa el ensemble  $S$  conformado por los clasificadores  $h1, h3$  y  $h4$ .

El algoritmo consiste en generar variaciones  $S'$  de la cadena  $S$  original y agregarlas a un conjunto de soluciones  $P$  siempre que cumplan la condición de dominación ( $f(S') < f(S)$  o  $|S'| < |S|$ ) mientras se eliminan las soluciones de  $P$  que cumplan la condición de dominación débil ( $f(S') \leq f(S)$  y  $|S'| \leq |S|$ ).

Por último, para cada solución  $S'$  generada se realiza una búsqueda local usando el algoritmo *variable depth search* para añadir las soluciones locales que cumplan la condición de dominación.

El resultado es un conjunto de ensembles  $P = \{S1, S2, \dots, Sn\}$  que puedan servir como alternativas al ensemble  $S$  original.

### 2.7.3. *Centralized Objection Maximization for Ensemble Pruning*

El algoritmo *Centralized Objection Maximization for Ensemble Pruning* (COMEP) se enfoca en la exactitud y la diversidad de los ensembles, que por lo general entran en conflicto entre sí, para equilibrar ambas y formalizar el problema de *pruning* de ensembles como un problema de maximización basado en la entropía de la información. Los autores mencionan que se debe incluir la diversidad (que puede ser considerada como la diferencia entre los miembros individuales en un ensemble) al momento de optimizar ensembles de clasificadores, por lo que definen una función objetivo que contiene tanto la métrica de exactitud y la diversidad entre los miembros del ensemble. Para lograr su objetivo, el algoritmo usa una búsqueda *greedy*, que selecciona el mejor ensemble en cada paso hasta lograr reducir el número de miembros del ensemble original a una cantidad arbitraria definida por el usuario.

Los autores mencionan que se puede abarcar diferentes tipos de problemas similares al reformular matemáticamente la función objetivo con tal para considerar un solo objetivo e incluso agregar el concepto de dominación (similar al de *Pareto Ensemble Pruning*), características que estudiarán en trabajos futuros [Bian et al., 2020].

## 2.8. Consideraciones Finales

En este capítulo se exponen brevemente las definiciones de las áreas de interés y algoritmos que serán usados en este proyecto. Adicionalmente se hizo una descripción básica de los modelos que serán usados para la generación de ensamblados en la herramienta propuesta. Cabe mencionar que cada uno de los modelos descritos son objeto de diversos estudios para realizarles mejoras, lo cual no es el objetivo de esta tesis, por lo que no se profundizará más en ellos. En el siguiente capítulo se mencionan algunos trabajos similares recientes.

## Capítulo 3

### Estado del Arte

#### 3.1. Consideraciones iniciales

En este capítulo se describirán algunos trabajos recientes similares a la propuesta de este proyecto. Se realizará un breve análisis sobre las fortalezas y debilidades de trabajos actuales.

#### 3.2. Trabajos relacionados

##### 3.2.1. Causality Explorer

La herramienta *Causality Explorer* [Xie et al., 2021] usa el algoritmo *Fast Greedy Equivalence Search* para la exploración de relaciones causales, aplicándolo en dos casos reales: análisis de estudiantes que se retiran de la universidad y análisis del impacto de campañas de marketing digital. Este trabajo también recalca la importancia de la elecciones que se tomen en el diseño de la herramienta, resaltando tres aspectos: facilidad de lectura de los resultados de la exploración causal, identificación de los atributos más influyentes y posibilidad de hacer un análisis *what-if*. Los investigadores encargados de hacer las pruebas y proporcionar sus comentarios a los autores de la herramienta señalaron que les agradó la representación de los nodos y apreciaron la animación del vínculo causal, ya que es intuitiva y estética. Este trabajo es útil para este proyecto pues su temática es muy similar, sin embargo todavía queda el reto de aplicar esta metodología en conjuntos de datos de alta dimensionalidad, ya que los conjuntos de datos usados en este trabajo contienen sólo treinta y dos dimensiones.

##### 3.2.2. SeqCausal

El framework *SeqCausal* [Jin et al., 2020] sirve para la visualización de datos secuenciales, los cuales guardan una relación causal a lo largo del tiempo. Su metodología consiste

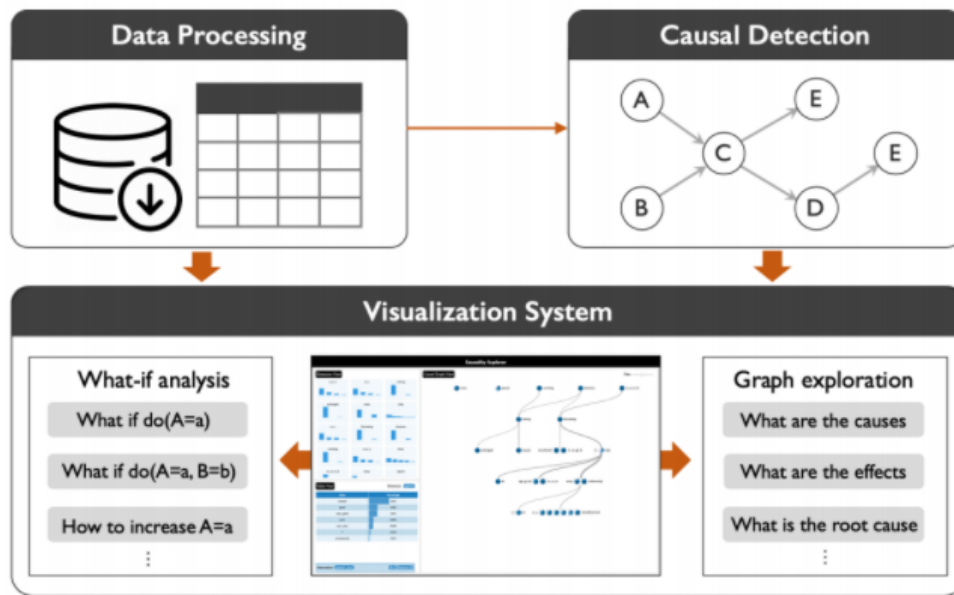


Figura 3.1: Componentes del sistema *Causality Explorer*. Imagen extraída de [Xie et al., 2021].

en tres pasos: el primero es modelar la secuencia de eventos en el tiempo, el segundo paso consiste en inferir la causalidad entre los atributos usando el modelo de procesamiento Hawkes y el último paso consiste en refinar el modelo usando la retroalimentación dada por el usuario. De forma similar al trabajo anterior, la herramienta provee un análisis what-if para poder ver los resultados de modificar alguna de las variables. Sin embargo, una limitación que mencionan los autores de esta herramienta es la pobre escalabilidad, porque probaron con nueve conjuntos de datos sintéticos de diferentes dimensiones, obteniendo tiempos de ejecución cada vez más elevados. Por último, los autores reconocen que su diseño actual no puede soportar completamente el análisis de datos de eventos secuenciales con una dimensionalidad muy alta.

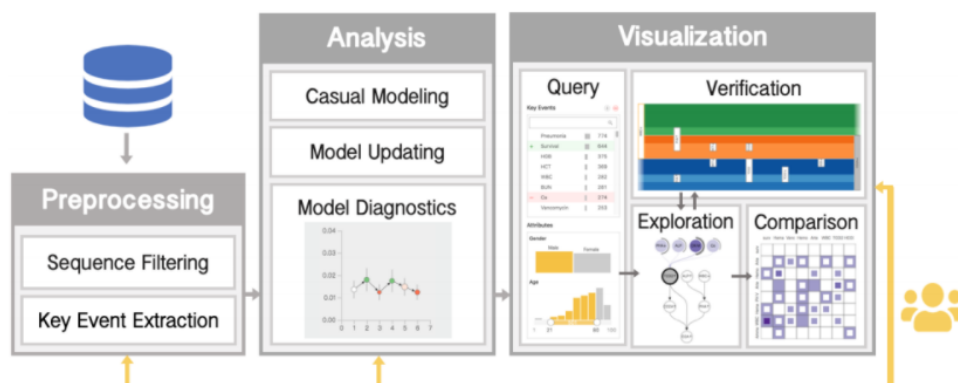


Figura 3.2: Sistema *SeqCausal*. Imagen extraída de [Jin et al., 2020].

### 3.2.3. Prospector

Por otro lado, el problema de la dificultad de entender los modelos de caja negra es abordado por la herramienta *Prospector* [Krause et al., 2016]. En este trabajo, el estudio de caso es la predicción de diagnósticos médicos usando conjuntos de datos de pacientes con diabetes. Esta herramienta sirve para poder determinar qué características de los pacientes son más importantes para determinar el riesgo real de diabetes. También cuentan con una interfaz que permite modificar los atributos y mostrar su impacto en el diagnóstico en tiempo real, a medida que se van modificando los atributos. De forma similar a los trabajos anteriores, esta herramienta también tiene sus propias limitaciones. Su principal limitación es que esta herramienta sólo puede manejar predicciones que resulten en una sola clase. Otra limitación de esta herramienta es que sólo admite la dependencia parcial de un atributo a la vez, lo cual no es algo común en conjuntos de datos reales, pues los atributos pueden estar correlacionados.

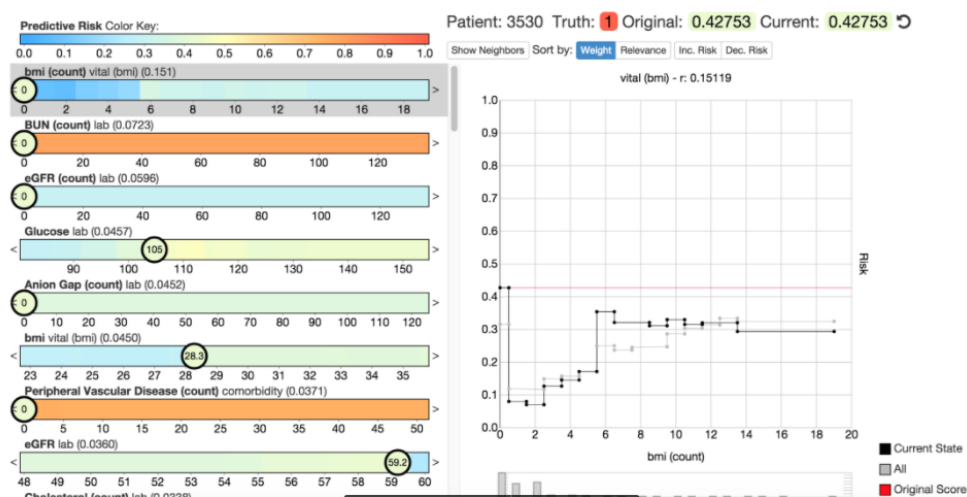


Figura 3.3: Sistema *Prospector*. Imagen extraída de [Krause et al., 2016].

### 3.2.4. Exploración visual de ensambles de clasificadores

En el área de visualización de ensambles, una tarea importante es la de proporcionar herramientas visuales para interpretar los resultados de un conjunto de clasificadores. Esta tarea es abordada en el trabajo de [Ribeiro et al., 2019] al mismo tiempo que brinda información sobre cómo cada clasificador contribuye a los resultados finales. Una de las principales contribuciones de este trabajo es que permite que la interpretación del por qué se tomó una decisión específica (clasificación) sea más entendible. Esto se logra mediante el uso de mapas de confusión y límites de decisión, permitiendo visualizar el comportamiento general del clasificador, así como inspeccionar regiones de interés. La principal limitación de este trabajo es la poca flexibilidad que ofrece con respecto a sus clasificadores, pues en este caso sólo existe un ensamble fijo (que contiene los clasificadores de regresión logística, red neuronal, árbol de decisión y bosque aleatorio). Este trabajo servirá como base para el presente proyecto, pues se buscará mejorarlo añadiendo más clasificadores con el objetivo de generar ensambles a medida del usuario y permitiendo la comparación en tiempo real de los mismos.

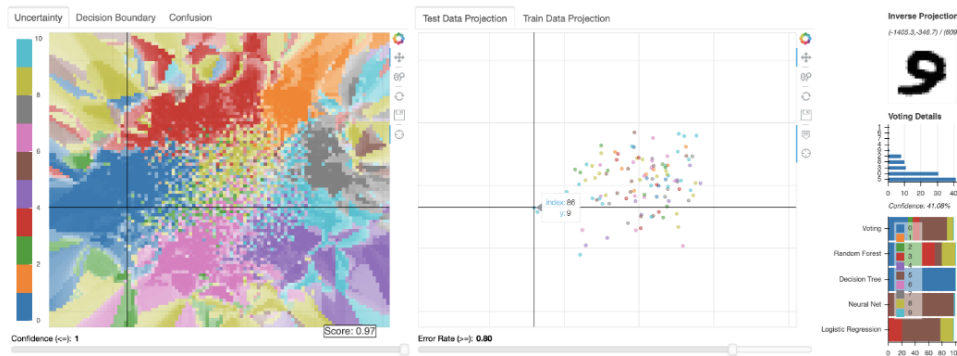


Figura 3.4: Vista general de la interfaz del trabajo de [Ribeiro et al., 2019]. Imagen extraída de [Ribeiro et al., 2019].

### 3.2.5. ICE

La herramienta analítica *ICE* [Castro-Ochante et al., 2021] no se enfoca en los clasificadores, sino que genera ensambles de métodos de agrupamiento de datos o *clustering*. Su enfoque muestra los diferentes resultados de los métodos de *clustering* (del estado del arte) con el objetivo de analizar su desempeño en función de métricas específicas y la inspección visual. Esta herramienta brinda muchas libertades a los analistas ya que (según su criterio) pueden asignar pesos de forma interactiva a los diferentes métodos para poder identificar sus contribuciones. Asimismo, les permite crear, almacenar, comparar y fusionar los ensambles generados. A pesar que el eje central de este trabajo es el clustering y no la clasificación, será de gran utilidad para el desarrollo de este proyecto, debido a que sus objetivos son muy similares a los objetivos propuestos en la presente tesis.

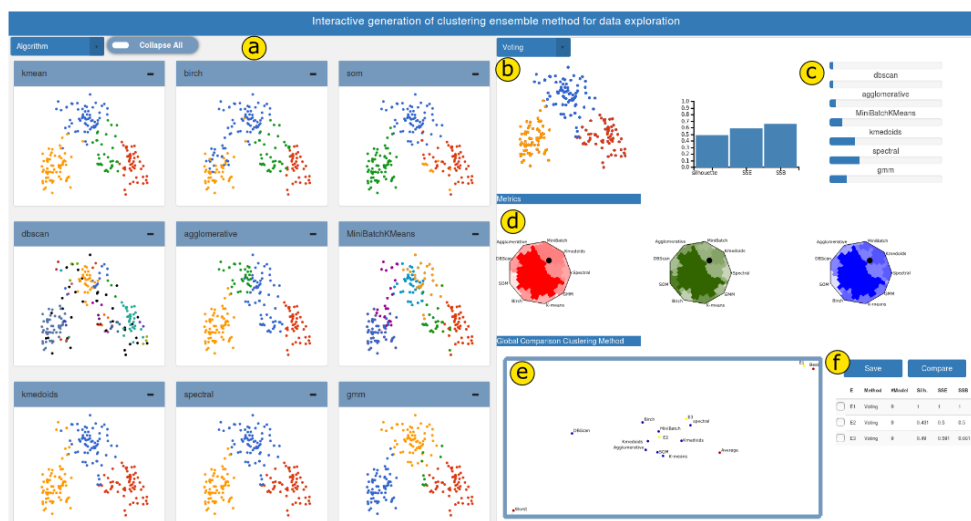


Figura 3.5: Vista general de la herramienta ICE. Imagen extraída de [Castro-Ochante et al., 2021].

### 3.2.6. Boxer

*Boxer* [Gleicher et al., 2020] es un sistema interactivo enfocado en comparar los resultados obtenidos por un conjunto de clasificadores. El sistema contiene varias vistas, entre las cuales se incluye: una vista de barras apiladas, usadas para comparar una métrica individual en todos los clasificadores; una gráfica con múltiples líneas para mostrar el desempeño de todos los clasificadores en todas las métricas, entre otros. Un factor clave para el desarrollo de esta propuesta fue la usabilidad, lo cual se refleja en la mínima complejidad de las interfaces de usuario. Este nivel bajo de complejidad resulta ser una limitación, para hacer análisis más complejos se deben combinar varias vistas y de formas más complejas.

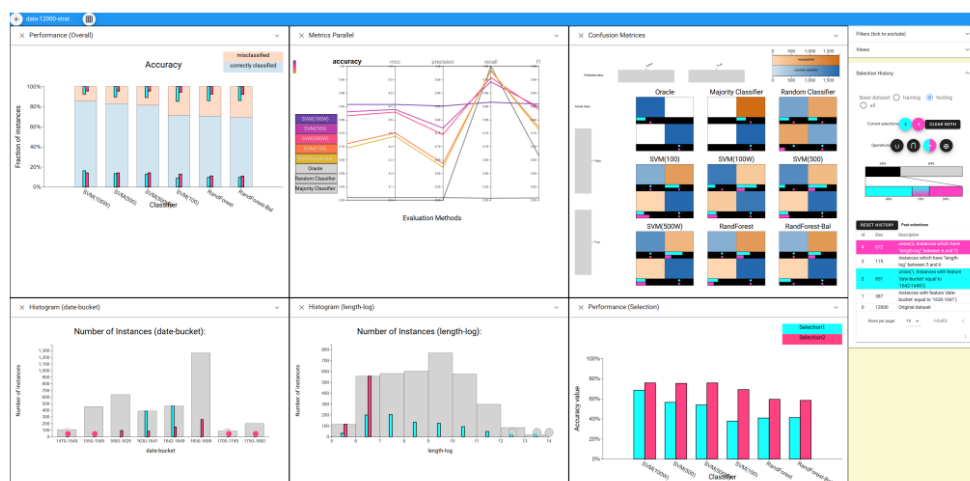


Figura 3.6: Vista de los componentes de la herramienta Boxer. Imagen extraída de [Gleicher et al., 2020].

### 3.2.7. Visualización de regiones de decisión

Otra propuesta emplea métodos de proyección multidimensional para generar mapas densos y poder inspeccionar interactivamente las regiones de decisión de modelos de clasificación. El objetivo de la propuesta de [Rodrigues et al., 2018] fue sobrepasar las deficiencias de las gráficas clásicas en dos dimensiones a través de un método de proyección multidimensional y una estrategia para combinar los resultados. La metodología propuesta incluye el uso de sobre muestrear una cantidad  $N$  de muestras para cada pixel, de tal forma que cada pixel muestre información de al menos  $N$  muestras de alta dimensionalidad. Aunque la evaluación a su metodología arrojó resultados positivos, la complejidad de los métodos de proyección multidimensional usados hicieron que su implementación sea incapaz de manejar rápidamente conjuntos de datos con decenas de miles de instancias.

### 3.2.8. Visualización del análisis exploratorio de modelos

Los autores Cashman et al. [Cashman et al., 2019] desarrollaron un flujo de trabajo que incorpora elementos de análisis visual para apoyar a los usuarios en el análisis de

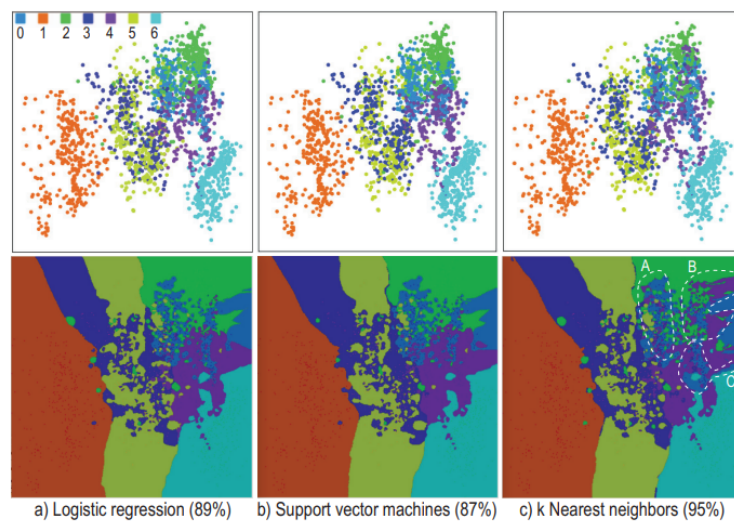


Figura 3.7: Ejemplo de proyecciones y mapas densos generados usando la propuesta de [Rodrigues et al., 2018](#). Imagen extraída de [Rodrigues et al., 2018](#).

conjuntos de datos, con el objetivo de encontrar clasificadores potencialmente útiles. Su flujo de trabajo propuesto contiene varios pasos, agrupados en tres categorías principales: Exploración del problema y datos, generación de modelos de clasificación y selección y exportación de modelos. Luego de realizar la exploración de datos, su metodología hace uso de la librería *autoML* para generar automáticamente una lista de candidatos a usar. Finalmente, el sistema provee diferentes visualizaciones para los candidatos generados, con el objetivo de ayudar al usuario a decidir cuáles usar y cuáles descartar. Aunque su trabajo fue alabado por su efectividad y facilidad de uso, también se hallaron limitación como la poca cantidad de opciones de personalización.

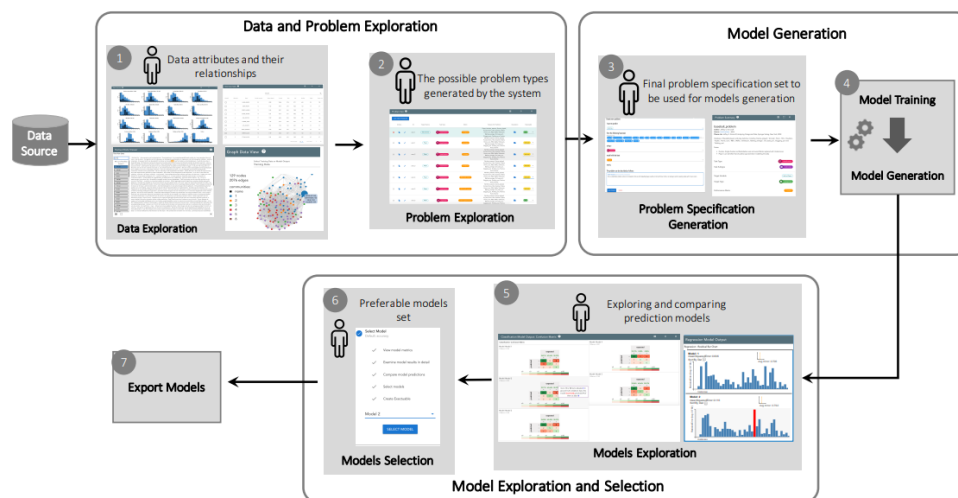


Figura 3.8: Flujo de trabajo propuesto por [Cashman et al., 2019](#). Imagen extraída de [Cashman et al., 2019](#).

### 3.2.9. ComDia+

*ComDia+* [Park et al., 2019] es un sistema interactivo de análisis visual para comparar el desempeño de modelos de clasificación multiclase en imágenes. Posee tres componentes: una vista de ranking para ordenar los clasificadores según su desempeño, una vista de diagnóstico que usa una variación de la *confusion matrix* para mostrar visualmente instancias mal clasificadas y una vista de corrección de malas clasificaciones para comparar los resultados (a nivel de instancias) entre varios modelos. Este sistema proporciona una forma efectiva de comparar el desempeño de varios clasificadores pero carece de escalabilidad para conjuntos más extensos y con imágenes de mayor complejidad.

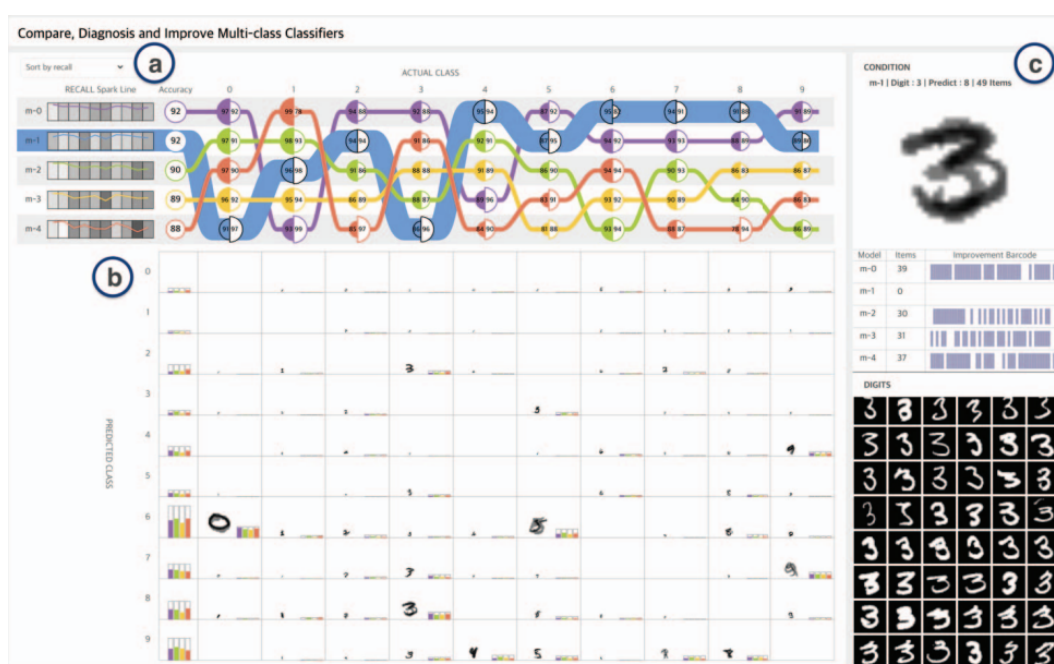


Figura 3.9: Vista general de ComDia+. Imagen extraída de [Park et al., 2019].

### 3.2.10. Integración visual de conjuntos de datos y modelos

Los autores Schneider et al [Schneider et al., 2017] propusieron un enfoque interactivo y visual para la exploración de resultados de clasificaciones hechas por un ensemble, pero que permita la interacción con los miembros del ensemble, de tal forma que se pueda experimentar con diferentes configuraciones de ensambles y reflejar esos cambios en los resultados. Esta metodología usa una técnica de proyección multidimensional para visualizar los elementos en un espacio de dos dimensiones. Las instancias proyectadas pueden seleccionarse para conocer cómo fueron clasificadas por cada modelo miembro del ensemble. De la misma forma, los usuarios pueden seleccionar o deseleccionar los clasificadores miembros del ensemble para conocer cómo serían clasificadas las instancias seleccionadas.

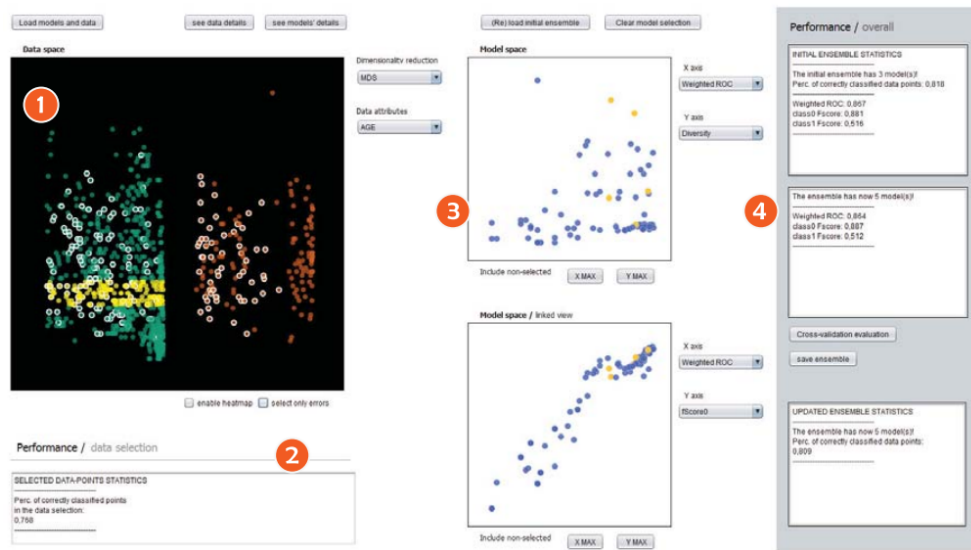


Figura 3.10: Vista general de la herramienta propuesta por [Schneider et al., 2017](#) para la exploración de ensambles de clasificadores. Imagen extraída de [Schneider et al., 2017](#).

### 3.3. Consideraciones Finales

En este capítulo se mencionaron algunos trabajos actuales con objetivos similares a la presente propuesta. Gracias a este breve análisis se pudieron identificar algunas debilidades en trabajos actuales que pueden servir como objetivo de mejora en el presente trabajo. En el siguiente capítulo se definirá la propuesta y los elementos que la componen

## Capítulo 4

# OCEG: Una herramienta de visualización interactiva para la generación de ensambles óptimos de clasificadores.

### 4.1. Consideraciones iniciales

Se explicará a profundidad los objetivos del presente trabajo, las tareas analíticas que se deben cumplir, los componentes principales de la herramienta propuesta y cómo se relacionan con las tareas analíticas definidas.

### 4.2. Análisis y diseño

#### 4.2.1. Objetivos

Esta propuesta consiste en desarrollar una herramienta visual que permita la agrupación de diferentes clasificadores en un ensamble, permitiendo el uso de las técnicas *soft voting* y *hard voting* para determinar la clase resultante en cada predicción. Los clasificadores que se pueden usar para la generación de ensambles serán cargados a la herramienta en un archivo comprimido, de tal forma que el usuario pueda usar los clasificadores que entrenó previamente. A diferencia del trabajo de [Ribeiro et al., 2019](#), la principal mejora será la capacidad de generar ensambles a medida del usuario, con una mayor gama de modelos disponibles, ya que el trabajo propuesto por este autor sólo contiene la visualización de un ensamble fijo (compuesto por modelos de *regresión logística*, *red neuronal*, *árbol de decisión* y *bosque aleatorio*) generado previamente. Asimismo, otra mejora que tiene esta herramienta con respecto al trabajo mencionado anteriormente es la capacidad de poder comparar diversos ensambles en forma simultánea con las métricas descritas en la Sección 2 (*accuracy*, *precision*, *recall* y *confusion matrix*).

### 4.2.2. Requerimientos

Como resultado del análisis de las fortalezas y debilidades de los trabajos actuales, se lograron formular los siguientes requerimientos para el desarrollo de una herramienta que cumpla con los objetivos propuestos.

- **R1 - Análisis de modelos de clasificación.** Explorar visualmente el rendimiento de los modelos de clasificación con respecto a múltiples características de poder predictivo, por ejemplo, regiones de decisión, métricas, entre otras.
- **R2 - Desarrollar ensambles de clasificadores.** Crear, explorar, comparar y exportar conjuntos de clasificadores de acuerdo con los conocimientos/requisitos del analista.
- **R3 - Optimizar ensambles de clasificadores.** Implementar mecanismos automáticos/interactivos para optimizar los ensambles generados.

### 4.2.3. Tareas analíticas

Para poder satisfacer los requerimientos mencionados previamente, se deben cumplir las siguientes tareas analíticas:

- **T1 - Visualizar las características de los modelos de clasificación.** El analista debe conocer los detalles del modelo para comprender sus capacidades, por ejemplo, el nombre del método, la *confusion matrix* y los resultados de las métricas de clasificación. Esta tarea busca satisfacer el requerimiento R1.
- **T2 - Generar interactivamente ensambles de clasificadores.** Los analistas deben ser capaces de agregar o eliminar dinámicamente cualquier modelo de clasificación para componer un nuevo conjunto clasificador. Esta tarea está relacionada con el requerimiento R2.
- **T3 - Asignar interactivamente pesos para cada miembro del ensamble basado en el desempeño individual.** La contribución de cada miembro del conjunto debe impactar positivamente. Por lo tanto, la herramienta debe guiar tales selecciones para maximizar el desempeño del ensamble. Esta tarea se relaciona con el requerimiento R2.
- **T4 - Realizar procedimientos para optimizar la composición del ensamble.** Los analistas deben ser capaces de afinar automáticamente los ensambles construidos, es decir, saber si todos los miembros que los componen son útiles para el ensamble. De lo contrario, la herramienta debe excluir a los miembros irrelevantes. Esta tarea se relaciona con el requerimiento R3.
- **T5 - Comparar el rendimiento de todos los modelos para evaluar la calidad del ensamble.** La herramienta debe introducir un mecanismo para mostrar visualmente todos los ensambles construidos durante la interacción del analista para comparar y extraer los mejores resultados. Esta tarea busca satisfacer los requerimientos R2 y R3.

- **T6 - Exportar conjuntos de clasificadores optimizados listos para usar.** Una vez que un ensamble específico muestre un rendimiento eficiente, se debe exportar en un formato común que cualquier biblioteca de aprendizaje automático pueda cargar e implementar. Esta tarea está relacionada con los requerimientos R2 y R3.

## 4.3. OCEG

### 4.3.1. Componentes Visuales

La herramienta propuesta, *Optimal Classifier Ensemble Generation* (**OCEG**), está conformada por cinco componentes principales:

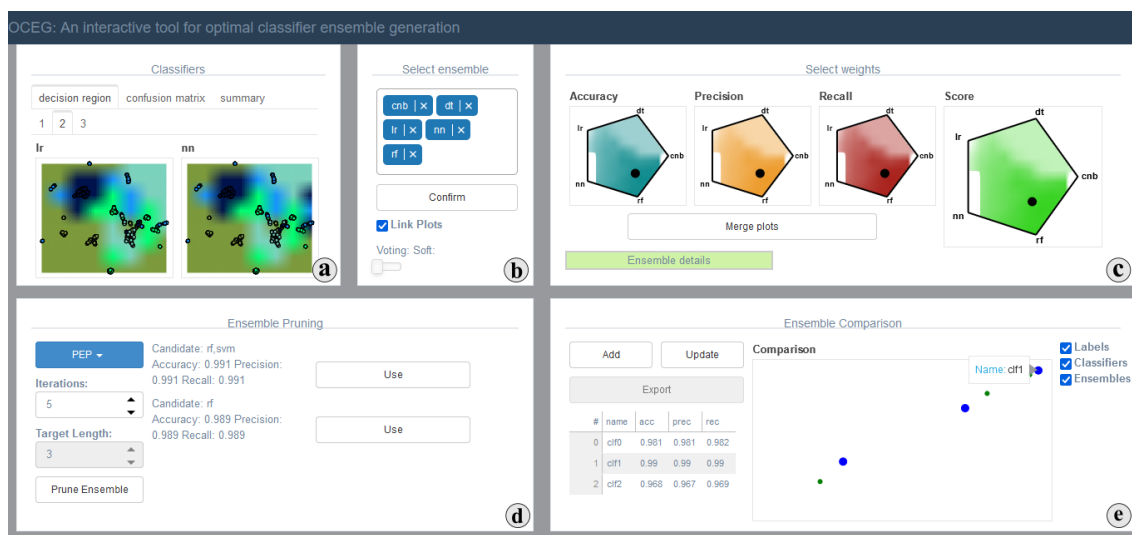


Figura 4.1: Vista general de la herramienta: (a) vista de clasificadores, (b) vista de generación de ensamble, (c) vista de selección de pesos, (d) vista de *pruning* y (e) vista de comparación. Imagen producida por el autor.

#### Vista de clasificadores

Esta vista permite visualizar las regiones de decisión y las *confusion matrix* de los clasificadores individuales cargados a la herramienta. El objetivo de esta vista es proporcionar información al usuario para que pueda decidir qué clasificadores usar en el ensamble.

Para el cálculo de las regiones de decisión primero se discretiza el espacio generando 225 puntos equidistantes entre sí de dos dimensiones.

Luego, cada uno de estos puntos de dos dimensiones es convertido a un vector de  $N$  dimensiones usando *UMAP*. La cantidad de dimensiones depende del conjunto de datos sobre el cual se entrenaron los clasificadores. Este proceso es llamado proyección inversa. A continuación se realizan predicciones con cada uno de estos vectores. Una vez completadas las predicciones, se colorean los puntos correspondientes a los vectores en base a la etiqueta predicha por cada clasificador.

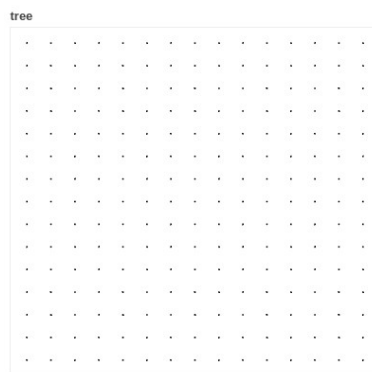


Figura 4.2: Ejemplo de discretización del espacio 2d. Imagen producida por el autor.

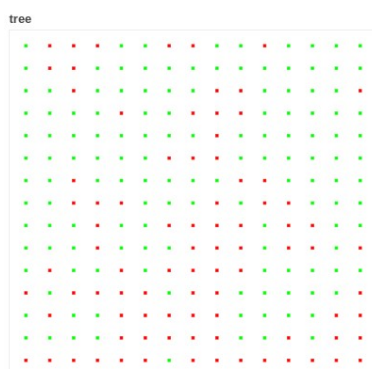


Figura 4.3: Coloreo del espacio discretizado en 2d. Imagen producida por el autor.

El siguiente paso es realizar una interpolación bilineal para hallar una representación aproximada de las regiones de decisión usando los puntos colocados anteriormente en la gráfica. Esta interpolación se realiza tres veces, una para cada canal (RGB) de los colores de los puntos.

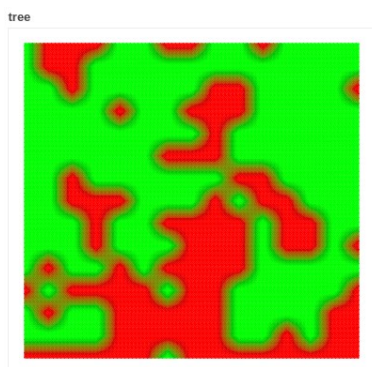


Figura 4.4: Coloreo de las regiones del espacio 2d. Imagen producida por el autor.

Una vez obtenidas las regiones, se aplica *UMAP* para reducir la dimensionalidad de los vectores del conjunto de prueba a puntos de dos dimensiones y se colocan en la gráfica generada previamente.

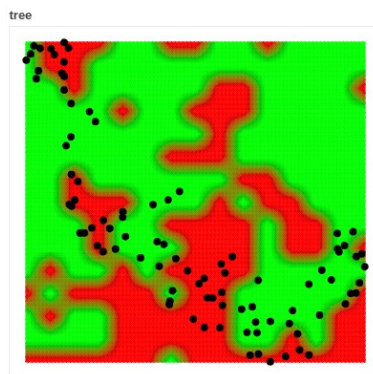


Figura 4.5: Colocación de los elementos del conjunto de prueba en las regiones de decisión. Imagen producida por el autor.

Por último, cada vector es usado para realizar predicciones y se colorean los puntos correspondientes según la etiqueta predicha por el clasificador.

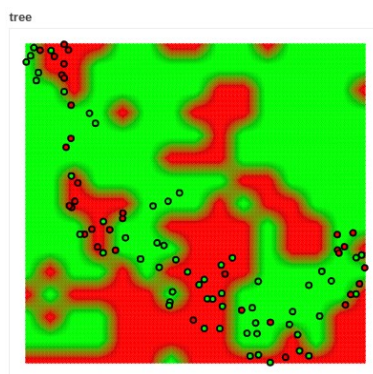


Figura 4.6: Coloreo de los elementos del conjunto de prueba en las regiones de decisión. Imagen producida por el autor.

El objetivo de esta vista es dar una idea inicial del comportamiento de los clasificadores por separado, para que el usuario pueda determinar cuáles usar en el ensamble. Con las regiones de decisión el usuario puede saber cómo estarían distribuidos los datos en un espacio de dos dimensiones y con la *confusión matrix* puede conocer qué clases son las que más confunde el clasificador al momento de realizar las predicciones.

### Vista de generación de ensamble

El objetivo de esta vista es realizar una selección de clasificadores individuales para generar un ensamble. Para lograr tal objetivo, se usan las siguientes herramientas:

**Selección de modelos:** es una selección múltiple que permite al usuario escoger qué clasificadores se usarán para el ensamble de clasificadores. Cuando el usuario hace clic en este componente, se despliega una lista con todos los clasificadores disponibles.

**Opciones de configuración:** estas opciones (Link plots y Voting) sirven para decidir si todas las gráficas que se generarán en el siguiente componente estarán enlazadas (Link plots) y para decidir si se usará *hard voting* o *soft voting* en el ensamble.

### Vista de selección de pesos

Esta vista contiene cuatro polígonos de asignación de pesos. El objetivo de cada polígono es brindar información de una métrica específica de forma visual.

**Polígono de asignación de pesos:** una vez seleccionados los clasificadores se procede a generar un polígono en el cual la cantidad de esquinas corresponde a la cantidad de clasificadores existentes en el ensamble.

Dentro del polígono existen diferentes áreas coloreadas en base al desempeño obtenido por un ensamble de clasificadores. Las áreas con un color más intenso son las que tienen un mayor desempeño. Existen cuatro polígonos, los tres primeros representan el *accuracy*, *precision* y *recall*, mientras que el último polígono representa la suma de estas tres métricas.

Para generar esta coloreo primero se realiza una discretización del espacio (de forma similar a la discretización de las regiones de decisión) generando puntos equidistantes a la largo de la gráfica. Se generan en total 100 puntos

Luego, se crean ensambles de clasificadores usando los pesos obtenidos al calcular las coordenadas de Wachspres (descritas en la Sección 2.13) de cada punto. Estos puntos son coloreados en base a la métrica seleccionada. Mientras mayor sea este valor, mayor será la intensidad con la que se colorea el punto.

Para terminar de llenar la figura, se realiza una interpolación bilineal en cada cuadrante de puntos (habiendo en total 81 cuadrantes). De esta forma se generan los puntos intermedios y su color aproximado sin necesidad de crear ensambles para cada combinación. Por último, sólo se muestran los puntos que se encuentran adentro del polígono.

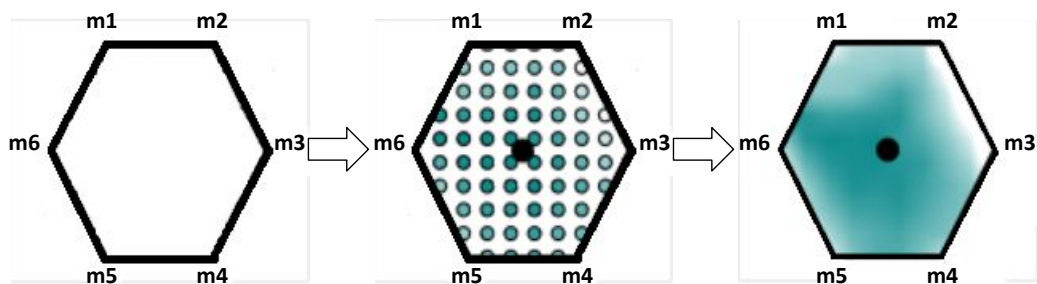


Figura 4.7: Procedimiento de coloreo de polígono. Imagen producida por el autor.

Dentro de este polígono se coloca un punto de mayor tamaño que servirá de cursor, su función es permitir la navegación entre las diversas distribuciones de pesos. El punto se posiciona por defecto en el centro del polígono.

En la vista desplegable de *Ensemble Details* se muestra de forma visual la contribución de cada clasificador dentro del ensamble y las métricas obtenidas. Esta vista auxiliar se actualiza a medida que el usuario arrastra el cursor dentro del polígono.

### Vista de pruning

Esta vista permite seleccionar un ensamble generado previamente usando los componentes anteriores y realizar una optimización usando los algoritmos *Pareto Ensemble*

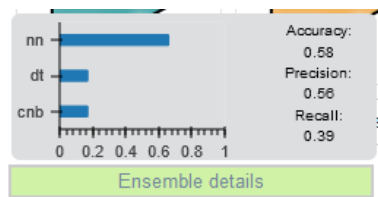


Figura 4.8: Asignación de pesos. Imagen producida por el autor.

*Pruning (PEP)* y *Centralized Objection Maximization for Ensemble Pruning (CO-MEP)*. Al finalizar el algoritmo, la lista de candidatos se presenta al analista. El analista puede seleccionar alguno de los candidatos para una posterior exploración de pesos usando nuevamente la vista de selección de pesos.

### Vista de comparación

Esta vista requiere que se hayan creado ensambles de clasificadores previamente. Se evalúan las métricas de cada ensamble generado y se grafican en un espacio de dos dimensiones. Cada punto dentro de esta vista representa las métricas *accuracy*, *precision* y *recall* obtenidas por el ensamble o clasificador, proyectadas a un punto en dos dimensiones usando *LAMP*. Adicionalmente se grafican dos puntos en las esquinas de la vista, uno representa el mínimo valor que pueden tener las métricas (*Worst*) y el otro que representa el máximo valor que pueden tener las métricas (*Best*). El objetivo es mostrar al usuario cómo se comparan los ensambles generados entre sí y con los clasificadores individuales de forma sencilla y fácil de entender.

#### 4.3.2. Análisis de componentes visuales y tareas analíticas

La implementación de cada una de las cinco vistas que componen la herramienta corresponde a una o varias tareas analíticas específicas, las cuales fueron definidas en la sección 4.3. La vista de clasificadores satisface la tarea analítica T1 porque, a través de sus elementos visuales, nos permite conocer los detalles de los modelos a usar. La vista de generación de ensamble satisface la tarea analítica T2, debido a que nos permite agregar o eliminar fácilmente clasificadores miembros de un ensamble. La vista de selección de pesos satisface las tareas analíticas T2 y T3 al proporcionar la información necesaria para direccionar la configuración de pesos y miembros del nuevo ensamble. La tarea analítica T4 es realizada por la vista de pruning, ya que ésta vista permite reducir el número de clasificadores miembros del ensamble, optimizando efectivamente su velocidad. Por último, la vista de comparación cumple las tareas analíticas T5 y T6, ya que permite comparar de manera sencilla de entender y rápida los ensambles generados. Asimismo, permite seleccionar cualquiera de los ensambles para exportarlos, con el fin de que puedan ser usados posteriormente. En el cuadro 4.1 se resume la relación de las vistas y las tareas analíticas que buscan satisfacer.

---

	<i>T1</i>	<i>T2</i>	<i>T3</i>	<i>T4</i>	<i>T5</i>	<i>T6</i>
Vista de clasificadores	✓					
Vista de generación de ensamble		✓				
Vista de selección de pesos		✓	✓			
Vista de <i>pruning</i>				✓		
Vista de comparación					✓	✓

---

Cuadro 4.1: Relación de los componentes visuales y las tareas analíticas que satisfacen.

### 4.3.3. Detalles de implementación

El apartado visual e interactivo del prototipo presentado en este trabajo se implementó usando la librería *Bokeh* de *Python*, que a su vez usa el framework *Tornado* para back end. Para la realización de las tareas de reducción de dimensionalidad y la proyección inversa se usó la librería del estado del arte *UMAP*, debido a que es más rápida y escalable que los métodos *Principal Component Analysis (PCA)*, *t-distributed Stochastic Neighbor Embedding (t-SNE)* y predecesores [Becht et al., 2018]. Se usó el algoritmo *LAMP* para la creación de la vista de comparación. Los clasificadores individuales utilizados para las pruebas se implementaron mediante la librería *scikit-learn* y los métodos usados para los ensambles fueron proporcionados por la librería *mlxtend*.

## 4.4. Consideraciones finales

En este capítulo se explicaron los objetivos del presente trabajo y la metodología usada para cumplirlos. En el siguiente capítulo se mencionará la metodología usada para la evaluación de la herramienta propuesta y los resultados obtenidos por la misma.

# Capítulo 5

## Pruebas y Resultados

### 5.1. Consideraciones iniciales

En el presente capítulo se detallará la metodología usada para evaluar la facilidad de uso y calidad de resultados de la herramienta propuesta. También se discutirán los resultados obtenidos.

### 5.2. Configuración de experimentos

Para evaluar la efectividad de la propuesta se realizaron dos estudios de caso usando conjuntos de datos relevantes en la actualidad (*COVID-19* y deudas de tarjetas de crédito). El propósito de los estudios de caso fue generar de forma fácil e interactiva ensambles de clasificadores, haciendo énfasis en el procedimiento utilizado para la generación y selección. Adicionalmente se realizó una evaluación cuantitativa usando 15 conjuntos de datos. La evaluación cuantitativa tuvo como objetivo analizar el desempeño de los ensambles generados, evaluando si pudieron superar o no a los clasificadores individuales que los componen.

### 5.3. Evaluación cuantitativa

#### 5.3.1. Descripción del experimento

Se usaron 15 conjuntos de datos recopilados del Repositorio de aprendizaje automático de UC Irvine [\[1\]](https://archive.ics.uci.edu/ml/index.php) para evaluar la eficiencia y utilidad de la propuesta. El Cuadro [5.1](#) resume los experimentos: cada columna representa un conjunto de datos diferentes, mientras que cada fila (en el nivel más a la izquierda) representa un modelo de clasificación diferente. Para ésta evaluación, se ejecutaron seis modelos de clasificación diferentes: *árbol de decisión* (**dt**), *bosque leatorio* (**rf**), *regresión logística* (**lr**), *red neuronal* (**nn**), *support vector machine* (**svm**) y *complement naive bayes* (**cnb**). La última fila describe el mejor conjunto producido por la propuesta.

---

<sup>1</sup>Conjuntos de datos del Repositorio de aprendizaje automático de UC Irvine: <https://archive.ics.uci.edu/ml/index.php>

Cuadro 5.1: Resultados en los conjuntos de datos (DS1-DS15).

ID	DS1	DS2	DS3	DS4	DS5	DS6	DS7	DS8	DS9	DS10	DS11	DS12	DS13	DS14	DS15
Nombre	<i>letter recognition</i>		<i>breast cancer</i>	<i>heart statlog</i>	<i>ionosphere</i>	<i>kr us kp</i>	<i>opt digits</i>	<i>landset statlog</i>	<i>tic tac toe</i>	<i>dry bean</i>	<i>spambase</i>	<i>phishing</i>	<i>segment statlog</i>	<i>wireless</i>	<i>australian statlog</i>
	<i>letter</i>	<i>magic</i>													
dt	acc	77.40%	90.70%	71.90%	86.20%	98.00%	84.90%	83.50%	87.70%	88.50%	86.40%	94.10%	94.90%	97.10%	77.20%
	prec	95.50%	77.40%	93.30%	71.10%	84.20%	98.00%	79.90%	87.30%	90.10%	86.50%	94.10%	94.90%	97.10%	77.20%
	rec	95.50%	77.40%	87.50%	73.20%	88.40%	98.10%	79.60%	86.30%	90.10%	86.50%	94.00%	94.80%	97.00%	76.90%
rf	acc	98.90%	85.20%	96.20%	79.80%	97.60%	97.60%	89.60%	89.90%	<b>91.60%</b>	94.50%	95.70%	96.50%	97.70%	85.10%
	prec	98.90%	85.20%	96.20%	79.00%	97.60%	97.60%	84.80%	88.80%	<b>92.60%</b>	94.50%	95.60%	96.50%	97.70%	84.90%
	rec	98.90%	85.20%	94.40%	82.30%	95.90%	97.70%	<b>88.60%</b>	89.10%	93.00%	94.50%	95.70%	96.50%	97.70%	84.90%
lr	acc	94.10%	78.40%	94.30%	<b>87.60%</b>	95.10%	96.20%	80.90%	69.60%	69.00%	92.80%	93.00%	89.80%	96.70%	86.80%
	prec	93.90%	78.30%	94.30%	<b>86.90%</b>	94.90%	96.20%	73.70%	63.80%	70.10%	92.80%	92.80%	89.80%	96.60%	87.00%
	rec	94.40%	78.70%	93.30%	<b>90.50%</b>	95.30%	96.30%	77.60%	66.60%	71.20%	92.80%	92.90%	90.00%	96.50%	86.60%
nn	acc	96.40%	82.40%	93.30%	78.70%	98.80%	97.10%	70.80%	70.90%	35.70%	93.40%	94.80%	92.50%	97.00%	84.60%
	prec	96.40%	82.40%	94.40%	78.40%	98.80%	97.20%	65.70%	66.00%	28.00%	93.30%	94.60%	92.50%	96.90%	84.60%
	rec	96.50%	82.50%	94.30%	78.70%	98.80%	97.30%	69.90%	68.10%	28.50%	93.30%	94.90%	92.40%	96.90%	84.40%
svm	acc	98.40%	83.10%	94.90%	52.80%	92.20%	98.70%	88.30%	74.40%	63.00%	69.60%	94.20%	93.20%	97.90%	86.80%
	prec	98.40%	83.00%	94.70%	50.00%	94.60%	98.70%	82.60%	69.80%	62.20%	69.70%	94.00%	93.00%	97.80%	87.00%
	rec	98.50%	83.20%	96.20%	26.40%	93.40%	94.90%	86.60%	72.40%	55.70%	69.80%	94.30%	93.30%	97.80%	86.60%
cub	acc	69.80%	77.90%	88.40%	74.20%	87.90%	81.60%	62.60%	52.50%	29.50%	76.80%	86.20%	48.00%	65.60%	76.30%
	prec	70.10%	77.70%	87.50%	74.00%	87.90%	82.10%	54.50%	52.30%	28.60%	76.80%	86.00%	49.20%	63.70%	77.30%
	rec	75.60%	78.30%	94.70%	74.10%	87.80%	85.00%	44.30%	52.10%	7.10%	76.80%	86.00%	49.80%	54.90%	77.10%
Ensemble	acc	<b>99.30%</b>	<b>85.40%</b>	<b>97.30%</b>	86.50%	<b>99.10%</b>	<b>99.00%</b>	<b>90.20%</b>	<b>91.10%</b>	91.50%	<b>94.80%</b>	<b>96.00%</b>	<b>96.90%</b>	<b>98.60%</b>	<b>87.70%</b>
	prec	<b>99.30%</b>	<b>85.30%</b>	<b>97.20%</b>	85.80%	<b>99.00%</b>	<b>99.00%</b>	<b>85.90%</b>	<b>90.20%</b>	92.40%	<b>94.80%</b>	<b>95.90%</b>	<b>96.90%</b>	<b>98.60%</b>	<b>87.80%</b>
	rec	<b>99.30%</b>	<b>85.40%</b>	<b>97.20%</b>	88.80%	<b>96.80%</b>	<b>99.10%</b>	<b>99.00%</b>	<b>90.50%</b>	<b>93.10%</b>	<b>94.80%</b>	<b>96.00%</b>	<b>96.90%</b>	<b>98.60%</b>	<b>87.50%</b>
	<i>Members:</i> dt, rf, lr, nn, svm, cub	<i>Members:</i> rf, svm	<i>Members:</i> lr, nn, svm, cub	<i>Members:</i> cub, dt, lr, nn, rf, svm	<i>Members:</i> cub, svm	<i>Members:</i> dt, nn, rf	<i>Members:</i> dt, nn, svm	<i>Members:</i> dt, nn, rf	<i>Members:</i> dt, lr, rf	<i>Members:</i> cub, dt, lr, nn, rf, svm	<i>Members:</i> dt, lr, nn, rf	<i>Members:</i> dt, nn, rf	<i>Members:</i> cub, dt, rf	<i>Members:</i> cub, dt, lr, nn, rf, svm	<i>Members:</i> cub, dt, lr, nn, rf, svm
	<i>Pesos:</i> 0.3, 0.17, 0.09, 0.08, 0.12, 0.25	<i>Pesos:</i> 0.9, 0.1 <i>Pruning:</i> PEP	<i>Pesos:</i> 0.15, 0.06, 0.05, 0.09, 0.28, 0.38	<i>Pesos:</i> 0.0, 0.04, 0.75, 0.2, 0.01, 0.0	<i>Pesos:</i> 0.7, 0.3 <i>Pruning:</i> PEP	<i>Pesos:</i> 0.11, 0.83, 0.06 <i>Pruning:</i> PEP	<i>Pesos:</i> 0.11, 0.72, 0.17 <i>Pruning:</i> COMEP	<i>Pesos:</i> 0.11, 0.06, 0.83 <i>Pruning:</i> PEP	<i>Pesos:</i> 0.11, 0.06, 0.83 <i>Pruning:</i> COMEP	<i>Pesos:</i> 0.13, 0.02, 0.01, 0.02, 0.18, 0.63 <i>Pruning:</i> None	<i>Pesos:</i> 0.05, 0.01, 0.16, 0.79 <i>Pruning:</i> PEP	<i>Pesos:</i> 0.22, 0.11, 0.67 <i>Pruning:</i> PEP	<i>Pesos:</i> 0.33, 0.06, 0.61 <i>Pruning:</i> COMEP	<i>Pesos:</i> 0.3, 0.17, 0.09, 0.08, 0.12, 0.25 <i>Pruning:</i> None	<i>Pesos:</i> 0.08, 0.02, 0.02, 0.05, 0.34, 0.49 <i>Pruning:</i> None

Se emplearon las métricas *accuracy* (*acc*), *precision* (*acc*) y *recall* (*rec*) para medir la calidad de los resultados. En la parte inferior, se detallan la configuración del ensamble clasificador generado, que describe los miembros del ensamble, los pesos de cada clasificador y el nombre del método de *pruning* utilizado. Finalmente, fueron resaltados los mejores valores por conjunto de datos en gris para mejorar la legibilidad de los resultados.

### 5.3.2. Discusión de los resultados

Como se puede observar, los ensambles obtenidos usando *OCEG* superaron (en términos de las tres métricas) a los modelos de clasificación en la mayoría de los casos, excepto en *DS4* y *DS10*. Para tales casos, los métodos *lr* y *rf* funcionan mejor, lo que confirma que los ensambles no funcionan para todos los problemas de clasificación. Sin embargo, el uso de ensambles de clasificadores con una configuración adecuada proporciona un nuevo modelo general que mejora potencialmente su poder predictivo.

El número de clasificadores que deben incluirse en un ensamble es un detalle específico a tomar en cuenta para cada conjunto de datos. Los resultados obtenidos mostraron que en algunos casos no es necesario crear un ensamble con varios miembros, por ejemplo en *DS2* el ensamble óptimo que obtuvo mejores resultados emplea solamente dos miembros, mientras que en otros casos como *DS3* y *DS4* los mejores resultados fueron obtenidos por ensambles que usaron todos los clasificadores disponibles.

Los mejores ensambles se obtuvieron mediante una inspección continua y guiada. Este proceso de generación de ensambles óptimos es el principal beneficio de la herramienta propuesta. Los resultados obtenidos evidencian lo útil que puede ser la herramienta. En la siguiente sección se detallará el flujo de trabajo que usa la herramienta para conseguir estos resultados, tomando como ejemplo dos estudios de caso.

## 5.4. Estudios de caso DS16 y DS17

El primero consiste en predecir si un paciente tiene *COVID-19* a partir de la información de síntomas (*DS16*). El segundo estudio de caso utiliza un conjunto de datos (*DS17*) de deudas de tarjetas de crédito para predecir si es más probable que una persona pague su deuda de tarjeta de crédito en el futuro. Cada conjunto de datos se distribuyó de la siguiente manera: un tercio para el conjunto de entrenamiento, un tercio para el conjunto de prueba y un tercio para el conjunto de validación (utilizado en la fase de optimización). Además, se entrenó un modelo *UMAP* con el conjunto de datos de prueba para poder visualizar datos de prueba en un espacio 2D.

### 5.4.1. Estudio de caso DS16: Predecir si un paciente tiene COVID-19

#### Características del conjunto de datos

Este conjunto de datos, que se encuentra en *Kaggle*, contiene información anónima recopilada de fuentes del gobierno mexicano disponibles públicamente. Este conjunto de datos originalmente contenía 23 atributos y 566602 instancias, pero para este experimento se redujo a 17 atributos y 72374 instancias. Se entrenaron los siguientes clasificadores: *árbol de decisión* (dt), *bosque aleatorio* (rf), *regresión logística* (lr), *bernoulli naive bayes* (bnb) y *complement naive bayes* (cnb). Todos los modelos y conjuntos de entrenamiento/prueba fueron incluidos en un archivo (.zip).

#### Procedimiento

Todo el procedimiento se resume en la Fig 5.1. El primer paso fue cargar el archivo zip que contiene todos los clasificadores a la herramienta. Luego, se obtiene una descripción general rápida de cada clasificador al observar la vista de clasificadores, que indica las regiones de decisión, las matrices de confusión y las métricas *accuracy*, *precision* y *recall*. De esta forma, se facilita la generación de las métricas estándar que utilizan los analistas para determinar la calidad de sus clasificadores.

Después de esta exploración inicial, se encontró que los mejores clasificadores para el conjunto de datos *DS16* eran *regresión logística* y *bosque aleatorio*; sin embargo, por motivos de demostración, se usaron todos los clasificadores disponibles para generar el primer ensamble. Este primer ensamble (que utilizó *soft voting* como método de votación) superó a todos los clasificadores individuales, con una puntuación de 0,744 en *accuracy*, *precision* y *recall*, mientras que el desempeño individual más alto lo obtuvieron los modelos de *regresión logística* y *red neuronal*, cada uno con una puntuación de 0,719 en *accuracy*, *precision* y *recall*. Como resultado, el punto que representa el primer ensamble (nombrado *Clf0*) dentro de la vista de comparación de ensambles está más cerca del mejor punto artificial que los puntos asociados a cada clasificador individual, lo que representa el rendimiento superior de *Clf0* sobre los clasificadores individuales. A pesar de obtener una solución factible, todavía se pueden obtener mejoras.

Después de obtener el primer ensamble, se usó la vista de selección de pesos para determinar los pesos óptimos para cada clasificador miembro del ensamble. El nuevo ensamble resultante de esta distribución de pesos (llamado *Clf1*) obtuvo 0,746 en *accuracy*, *precision* y *recall*, alcanzando una mejora mínima sobre *Clf0* (que asignaba el mismo peso a todos los clasificadores individuales). También se repitió el mismo procedimiento, pero utilizando *hard voting* para generar el ensamble *Clf2*. Este nuevo ensamble obtuvo resultados inferiores a los producidos por *Clf0* y *Clf1*, por lo que fue descartado.

El siguiente paso fue generar un ensamble más rápido mediante la reducción del número de clasificadores miembros, pero sin reducir considerablemente el desempeño. Se usó la vista de *pruning* para encontrar candidatos que pudieran reemplazar a *Clf1*. El primer método utilizado fue el de *Pareto Ensemble Pruning (PEP)*. La implementación realizada de este algoritmo necesita el parámetro definido por el

usuario *Número de iteraciones*. En este experimento se decidió usar tres iteraciones porque con iteraciones más altas; ésta implementación redujo significativamente el número de clasificadores. Sin embargo, con menos iteraciones, no pudo realizar una exploración óptima. El algoritmo seleccionó una combinación de cinco clasificadores individuales (*complement naive bayes*, *árbol de decisión*, *regresión logística*, *red neuronal* y *bosque aleatorio*) para el nuevo ensamble (nombrado *Clf3*). *Clf3* obtuvo 0,747 en *accuracy*, *precision* y *recall*, superando ligeramente los resultados de *Clf1* pero con un clasificador menos, lo que lo hace más rápido.

También se utilizó el algoritmo *COMEP* para encontrar candidatos que pudieran reemplazar a *Clf1*. Este algoritmo requiere el parámetro definido por el usuario *Longitud final*, que indica el número de clasificadores que debe tener el nuevo ensamble. Esta vez se estableció en tres. El algoritmo seleccionó los clasificadores *complement naive bayes*, *árbol de decisión* y *regresión logística* para el nuevo ensamble, nombrado *Clf4*. *Clf4* obtuvo 0,737 en *accuracy*, *precision* y *recall*. Después de generar *Clf4*, se intentó optimizarlo utilizando la vista Selección de pesos para generar una distribución óptima de pesos. El nuevo ensamble (*Clf5*) obtuvo una puntuación de 0,744 en *accuracy*, *precision* y *recall*, superando a *Clf4* pero por debajo de *Clf1*. Finalmente, se usó la vista de selección de pesos para optimizar *Clf3*; el creado con *PEP*, con una puntuación de 0,748 en *accuracy*, *precision* y *recall*. Se concluye que los ensambles generados *Clf3* y *Clf5* fueron más rápidos que el ensamble *Clf0* inicial (ya que se construyeron con menos clasificadores) y mantuvieron resultados similares (*Clf3* lo superó ligeramente).

### 5.4.2. Estudio de caso DS17: Predecir si una persona pagará o no su deuda de tarjeta de crédito

#### Características del conjunto de datos

Este conjunto de datos se encontró en el repositorio de aprendizaje automático de UC Irvine. Contiene 23 atributos y 30000 instancias.

#### Procedimiento

El primer paso fue cargar los clasificadores entrenados y los datos en la herramienta a través de la vista de clasificadores. Después de una exploración rápida de la información obtenida en la vista de clasificadores, se llegó a la conclusión de que los mejores clasificadores para esta tarea eran *red neuronal*, *bosque aleatorio* y *support vector machine*, por lo que fueron escogidos para el primer ensamble generado. Este primer ensamble (nombrado *Clf0*) logró obtener 0,711 en *accuracy*, 0,709 en *precision* y 0,716 en *recall*. El *accuracy* y *precision* obtenidos por el conjunto *Clf0* fueron superiores a las obtenidos por todos los clasificadores individuales. Sin embargo, su *recall* fue superado por el clasificador individual *support vector machine* (0.723).

Después de generar el primer ensamble, se usó la vista de selección de pesos para optimizarlo seleccionando los pesos óptimos para cada clasificador. El ensamble resultante (*Clf1*) obtuvo 0,717 en *accuracy*, 0,716 en *precision* y 0,721 en *recall*. Aunque estos resultados superaron los obtenidos por *Clf0*, el *recall* todavía estuvo por debajo del obtenido por *support vector machine*. La vista de comparación de

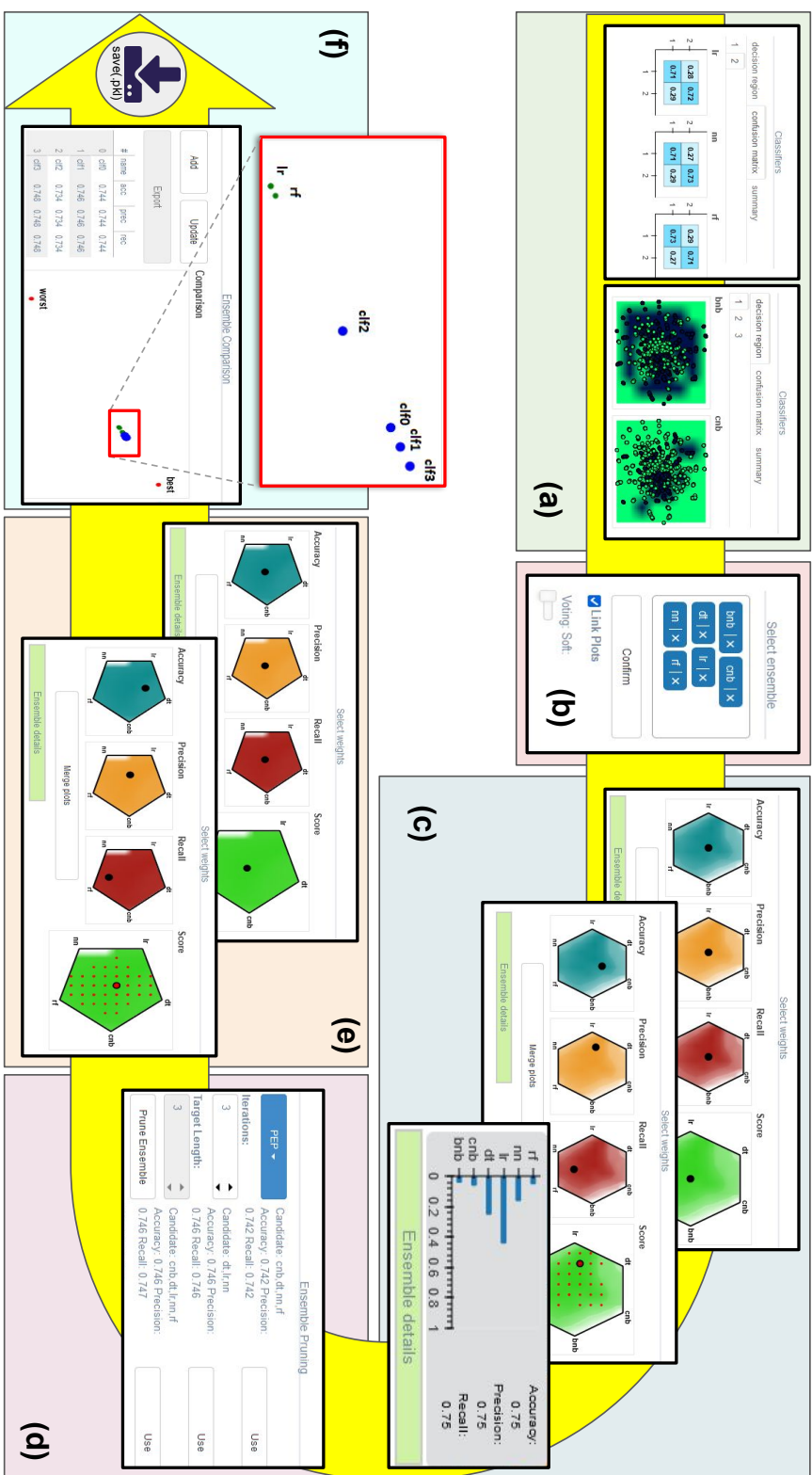


Figura 5.1: Resumen del primer estudio de caso: (a) cargado e inspección de modelos, (b) creación del primer ensamble, (c-e) asignación de pesos y optimización de ensambles, y (f) exportar el mejor ensamble. Imagen producida por el autor.

ensambles reveló este contraste después de agregar *Clf0* y *Clf1*. *Clf0* está mapeado entre los puntos de *bosque aleatorio* y *support vector machine*, mientras que *Clf1* se acercó al mejor punto ya que superó a *Clf0* en todas las métricas obtenidas. Sin embargo, todavía se encuentra en el lado izquierdo de *support vector machine* porque obtuvo 0.723 en *recall*, mientras que *Clf1* solo logró 0.721.

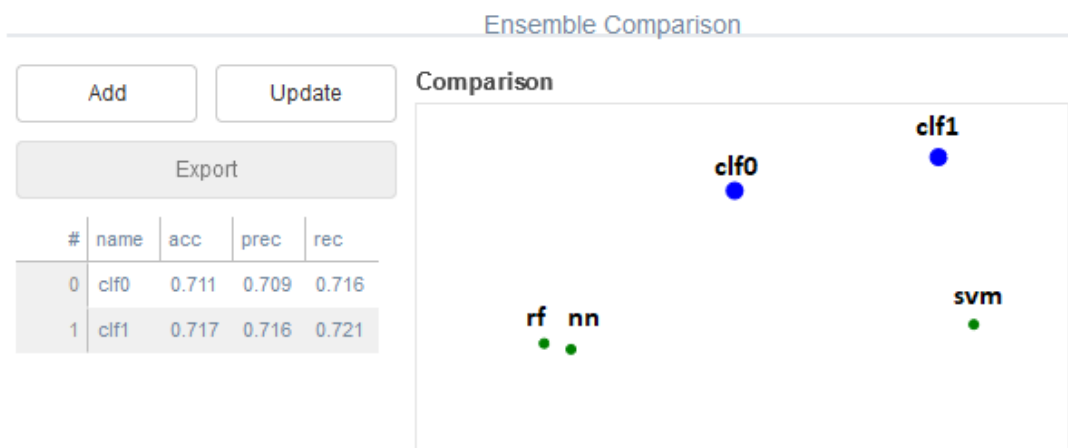


Figura 5.2: Proyección de los clasificadores individuales y ensambles en la vista de comparación para el segundo estudio de caso. Imagen producida por el autor.

Aunque no logró superar el *recall* de *support vector machine*, el ensamble *Clf1* fue una mejora con respecto a todos los clasificadores individuales, superándolos en términos de *accuracy* y *precision*. Finalmente, se intentó reducir el número de clasificadores individuales en este ensamble utilizando la vista de *pruning*. Aún así, los resultados obtenidos por ambos algoritmos fueron inferiores.

## 5.5. Consideraciones finales

En este capítulo se expusieron los resultados obtenidos por la herramienta propuesta al ser evaluada. Gracias a la evaluación cuantitativa se puede concluir que, en la mayoría de casos, los ensambles generados por esta propuesta lograrán superar a los modelos de clasificación individuales; y gracias a los estudios de caso se pudo evidenciar la facilidad de uso de la propuesta. En el siguiente capítulo se enunciarán las conclusiones con mayor detalle.

## Capítulo 6

# Conclusiones y Trabajos Futuros

Este trabajo presentó una herramienta para la creación, evaluación y optimización de ensamblajes de clasificadores. Gracias a los resultados obtenidos tanto por los estudios de caso como por la evaluación cuantitativa, podemos concluir que una buena asignación de pesos conlleva a un desempeño superior en comparación al desempeño de cada clasificador por separado. La herramienta propuesta cumplió con su objetivo, ya que permitió crear ensamblajes interactivamente y brindar una mejora ante el uso individual de clasificadores. Cabe mencionar que para ambos estudios de caso se lograron crear ensamblajes que superaban a los clasificadores individuales que los componían, debido a que estos clasificadores individuales no eran los más eficientes, pero en conjunto brindaron mejores resultados.

Aunque la herramienta produce una distribución de pesos óptima aproximada, esta resulta útil para mejorar el desempeño general del ensamblaje. Asimismo, esta aproximación podría tomarse como punto de partida para realizar una exploración exhaustiva de distribuciones de pesos para cada clasificador, con la ventaja de poder limitar el espacio de búsqueda a distribuciones cuya variación no sea demasiado extensa, con respecto a la distribución óptima aproximada generada por la herramienta.

### 6.1. Problemas Encontrados

El principal problema, así como en muchos trabajos dentro de esta área, es el de la búsqueda de optimizaciones para maximizar ciertas métricas. Alterar los parámetros para incrementar una determinada métrica puede causar que otra sea disminuida, por lo que debe priorizarse cada métrica según la tarea que desee realizarse. Por ejemplo, en un sistema de filtrado de videos aptos para niños se debe priorizar *precision*, ya que el impacto de un dato clasificado incorrectamente es muy alto (mostrar un video no apto a niños). Por otro lado, en un sistema de detección de enfermedades debe priorizarse *recall*, porque el impacto de un falso positivo (se clasifica a un paciente como portador de una enfermedad cuando en realidad no la tiene) no es tan significativo como el impacto de un falso negativo (se clasifica a un paciente como no portador de una enfermedad cuando en realidad sí la tiene, causando que la enfermedad no sea tratada). En este trabajo se usó como criterios de evaluación las métricas descritas en la Sección 2. Estas métricas no son las únicas

que existen y puede ser necesario, dependiendo del tipo de tarea que se realiza, el uso de otras métricas complementarias.

Otro problema encontrado es evitar el incremento del tiempo de ejecución total de la herramienta. Hay diversos factores que contribuyen al aumento del tiempo de ejecución. El primero es la cantidad de iteraciones máximas definidas para los clasificadores. Aumentar el número de iteraciones puede brindar mejores resultados en algunos casos, pero también aumenta el tiempo total de ejecución. El segundo factor es el ajuste de parámetros del algoritmo de reducción dimensional (de *UMAP*). Mientras más preciso sea el algoritmo en la realización de las reducciones, más tiempo demora en alcanzar la convergencia. El último factor es la cantidad de puntos evaluados para la generación de las regiones de decisión. En ésta herramienta no se usó una cantidad demasiado amplia de puntos para tal fin, por lo que las regiones sólo proveen una representación aproximada. La importancia de los tiempos de ejecución impacta en la usabilidad, ya que una herramienta lenta puede resultar tediosa y terminar siendo descartada.

## 6.2. Recomendaciones

La principal y más importante recomendación al trabajar con ensambles de clasificadores es asignar pesos en base a las métricas que se deseen maximizar y al desempeño de los clasificadores individuales para evitar una disminución en la eficacia.

Otro aspecto a tomar en cuenta es el propósito de cada clasificador, pues cada clasificador tiene ventajas frente a los demás en determinadas tareas, mientras que en otras tareas puede tener menor efectividad. Por tal motivo se deben seleccionar los clasificadores que mejor se adecúen a la tarea en cuestión.

## 6.3. Trabajos Futuros

Para trabajos futuros debería desarrollarse una herramienta que permita trabajar tareas de regresión. De esta forma la herramienta podrá ser usada en un número mayor de tareas analíticas. Otro aspecto importante que se debe considerar para un trabajo futuro es la realización de pruebas de uso con personas que trabajen en áreas relacionadas. El objetivo de estas pruebas será medir la usabilidad, tomando como base las opiniones de diversos usuarios. Este paso será crucial para el éxito o fracaso de la herramienta propuesta en la presente tesis, ya que en base a estos comentarios se podrá refinar la implementación y posterior distribución de la herramienta.

Por último, un framework a desarrollarse posteriormente debería permitir la asignación de parámetros iniciales a cada clasificador, ya que algunos tienen varios modos (por ejemplo los kernels disponibles en el clasificador de Máquina de Vectores Soporte) que tienen una mayor o menor efectividad dependiendo del conjunto de datos que usen.

## Bibliografía

- [Abiodun et al., 2018] Abiodun, O., Jantan, A., Omolara, O., Dada, K., Mohamed, N., and Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4:e00938.
- [Becht et al., 2018] Becht, E., McInnes, L., Healy, J., Dutertre, C.-A., Kwok, I. W. H., Ng, L. G., Ginhoux, F., and Newell, E. W. (2018). Dimensionality reduction for visualizing single-cell data using umap. *Nature Biotechnology*.
- [Bian et al., 2020] Bian, Y., Wang, Y., Yao, Y., and Chen, H. (2020). Ensemble pruning based on objection maximization with a general distributed framework. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3766–3774.
- [Cashman et al., 2019] Cashman, D., Humayoun, S. R., Heimerl, F., Park, K., Das, S., Thompson, J., Saket, B., Mosca, A., Stasko, J., Endert, A., et al. (2019). A user-based visual analytics workflow for exploratory model analysis. In *Computer Graphics Forum*, volume 38, pages 185–199. Wiley Online Library.
- [Castro-Ochante et al., 2021] Castro-Ochante, J., Camara-Chavez, G., and Gomez-Nieto, E. (2021). Ice: A visual analytic tool for interactive clustering ensembles generation. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing, SAC '21*, page 400–408, New York, NY, USA. Association for Computing Machinery.
- [Chatzimparmpas et al., 2021] Chatzimparmpas, A., Martins, R. M., Kucher, K., and Kerren, A. (2021). Stackgenvis: Alignment of data, algorithms, and models for stacking ensemble learning using performance metrics. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1547–1557.
- [Denisko and Hoffman, 2018] Denisko, D. and Hoffman, M. M. (2018). Classification and interaction in random forests. *Proceedings of the National Academy of Sciences*, 115(8):1690–1692.
- [Elsheikh et al., 2019] Elsheikh, A. H., Sharshir, S. W., Abd Elaziz, M., Kabeel, A., Guilan, W., and Haiou, Z. (2019). Modeling of solar energy systems using artificial neural network: A comprehensive review. *Solar Energy*, 180:622–639.
- [Floater, 2014] Floater, M. S. (2014). Wachspress and mean value coordinates. In Fasshauer, G. E. and Schumaker, L. L., editors, *Approximation Theory XIV: San Antonio 2013*, pages 81–102, Cham. Springer International Publishing.
- [Gleicher et al., 2020] Gleicher, M., Barve, A., Yu, X., and Heimerl, F. (2020). Boxer: Interactive comparison of classifier results. In *Computer Graphics Forum*, volume 39, pages 181–193. Wiley Online Library.

- 
- [Géron, 2019] Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and Tensorflow*. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA.
- [Jahed Armaghani et al., 2020] Jahed Armaghani, D., Asteris, P. G., Askarian, B., Hasanipanah, M., Tarinejad, R., and Huynh, V. V. (2020). Examining hybrid and single svm models with different kernels to predict rock brittleness. *Sustainability*, 12(6).
- [Jijo and Mohsin Abdulazeez, 2021] Jijo, B. and Mohsin Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2:20–28.
- [Jin et al., 2020] Jin, Z., Guo, S., Chen, N., Weiskopf, D., Gotz, D., and Cao, N. (2020). Visual causality analysis of event sequence data. *CoRR*, abs/2009.00219.
- [Joia et al., 2011] Joia, P., Coimbra, D., Cuminato, J. A., Paulovich, F. V., and Nonato, L. G. (2011). Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2563–2571.
- [Krause et al., 2016] Krause, J., Perer, A., and Ng, K. (2016). Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 5686–5697, New York, NY, USA. Association for Computing Machinery.
- [Lidberg, 2011] Lidberg, P. (2011). Barycentric and wachspres coordinates in two dimensions: theory and implementation for shape transformations.
- [Luque et al., 2019] Luque, A., Carrasco, A., Martín, A., and de las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91:216–231.
- [McInnes et al., 2018] McInnes, L., Healy, J., and Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction.
- [Park et al., 2019] Park, C., Lee, J., Han, H., and Lee, K. (2019). Comdia+: An interactive visual analytics system for comparing, diagnosing, and improving multiclass classifiers. In *2019 IEEE Pacific Visualization Symposium (PacificVis)*, pages 313–317. IEEE.
- [Patel and Prajapati, 2018] Patel, H. and Prajapati, P. (2018). Study and analysis of decision tree based classification algorithms. *International Journal of Computer Sciences and Engineering*, 6:74–78.
- [Qian et al., 2015] Qian, C., Yu, Y., and Zhou, Z.-H. (2015). Pareto ensemble pruning. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*.
- [Rennie et al., 2003] Rennie, J. D. M., Shih, L., Teevan, J., and Karger, D. R. (2003). Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 616–623. AAAI Press.
- [Ribeiro et al., 2019] Ribeiro, P. C., Schardong, G. G., Barbosa, S. D., de Souza, C. S., and Lopes, H. (2019). Visual exploration of an ensemble of classifiers. *Computers Graphics*, 85:23–41.
- [Rodrigues et al., 2018] Rodrigues, F. C. M., Hirata, R., and Telea, A. C. (2018). Image-based visualization of classifier decision boundaries. In *2018 31st SIBGRA-*

*PI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 353–360. IEEE.

- [Sagi and Rokach, 2018] Sagi, O. and Rokach, L. (2018). Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4):e1249.
- [Schneider et al., 2017] Schneider, B., Jäckle, D., Stoffel, F., Diehl, A., Fuchs, J., and Keim, D. (2017). Visual integration of data and model space in ensemble learning. In *2017 IEEE Visualization in Data Science (VDS)*, pages 15–22.
- [Xie et al., 2021] Xie, X., Du, F., and Wu, Y. (2021). A visual analytics approach for exploratory causal analysis: Exploration, validation, and applications. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1448–1458.
- [Yuan et al., 2021] Yuan, J., Chen, C., Yang, W., Liu, M., Xia, J., and Liu, S. (2021). A survey of visual analytics techniques for machine learning. *Computational Visual Media*, 7(1):3–36.