ESCUELA POLITÉCNICA SUPERIOR DE CÓRDOBA
Universidad de Córdoba

EPSC

TRABAJO FIN DE GRADO

*Grado en Ingeniería Informática – Especialidad Computación*

# Herramienta para el análisis y la minería de datos en autoevaluación de alumnos

## Manual técnico

**Autor:** Antonio Rafael Carmona Mengual (i42camea@uco.es)

**DNI: 31012806-C** **Telf.: 679630458**

**Directores:**

**Prof. Dr. Cristóbal Romero Morales (cromero@uco.es)**

**Prof. Dr. Alberto Cano Rojas (acano@vcu.edu)**

UNIVERSIDAD D CÓRDOBA

Proyecto realizado por **D. Antonio Rafael Carmona Mengual**, alumno del Grado en Ingeniería Informática de la Escuela Politécnica Superior de Córdoba con especialidad en computación para que conste se ha firmado el siguiente documento en Córdoba a 5 de septiembre de 2018.

X

D. Antonio Rafael Carmona Mengual
Estudiante

# Certificado

Prof. Dr. Cristóbal Romero Morales, profesor titular del Departamento de Informática y Análisis Numérico de la Escuela Politécnica de la Universidad de Córdoba y el Prof. Dr. Alberto Cano, profesor ayudante del Departamento de Informática de la Virginia Commonwealth University, Richmond, Virginia, EE. UU..

## INFORMAN:

Que han dirigido el **Trabajo Fin de Grado** de la titulación **Grado en Ingeniería Informática** denominado "**Herramienta para el análisis y la minería de datos en autoevaluación de alumnos**" realizado por D. Antonio Rafael Carmona Mengual en la Escuela Politécnica Superior de la Universidad de Córdoba, reuniendo a su juicio, las condiciones establecidas en este tipo de trabajos.

Y para que conste, firman el presente informe en Córdoba a 5 de septiembre de 2018.

X

Prof. Dr. Cristóbal Romero Morales
Director de proyecto

X

Prof. Dr. Alberto Cano Rojas
Director de proyecto

*“Mira, no puedes diseñar tu vida como un edificio.*

*No funciona de esa manera. Solo tienes que vivirla …*

*Y se diseñara ella sola.”*

*-How I met your mother*

# Agradecimientos

Parece que fuera ayer cuando entre a mi primera clase y son ya cuatro años los que han pasado desde el comienzo de esta etapa de mi vida, la universitaria. Sabía que el camino no sería fácil pero las ganas por aprender superaron mis miedos por enfrentarme a lo desconocido y tras duros años de esfuerzo y trabajo conseguí mis objetivos y todo esto no lo habría conseguido sin vosotros y por eso os lo agradezco.

A los que dejasteis de ser compañeros de clase para convertirse en amigos y algunos llegasteis a ser familia, a los que compartís alguno de mis apellidos y no necesariamente en vuestro nombre, porque sin vosotros nada fuera sido posible. A los que estáis conmigo siempre apoyándome en los momentos malos y buenos. Siempre os llevare conmigo familia.

A ti guapa, por haber dado todo lo que llevas dentro para que yo pudiera recoger una pequeña parte, por tu apoyo, por todas esas noches hasta las tantas de la madrugada hablando, por todos tus consejos, ya que mi título en parte también es tuyo porque sin ti no fuera llegado a donde estoy ahora mismo.

Para acabar agradecer a lo más importante de mi vida, a ti mamá, por ser la estrella que me guía y ha hecho que se la persona que soy ahora y a ti papá, por no estar siempre hay detrás de mí para ayudarme a levantarme cuando me he caído o apoyarme cuando lo he necesitado, gracias a los dos por todo porque sin vosotros y vuestro apoyo este sueño nunca fuera sido realidad.

# Tabla de contenido

# Índice de ilustraciones:

# Índice de tablas:

# Sección I: Introducción e investigación.

# 1. Introducción

Entendemos como aprendizaje automático [1] a la rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras *"aprender"* de forma autónoma. La forma en la que aprende sería gracias al uso de ciertas bases de datos, las cuales están formadas por un conjunto de ejemplos de entrenamiento etiquetados. Estas colecciones de datos son la mayoría de las veces proporcionadas por una serie de expertos. El objetivo del sistema de aprendizaje es inferir un modelo de predicción a partir de los datos de entrenamiento, siendo capaz de dar una salida a nuevos datos de entrada.

Por lo tanto, gracias a esta rama conseguimos poder facilitar el día a día a usuarios en algunos de sus trabajos, como por ejemplo el poder diagnosticar de formas más rápida y precisa ciertas enfermedades o cosas tan sencillas como el poder clasificar correos para decir si son spam o no.

La minería de datos [2] es un campo de la estadística y de las ciencias de la computación, con esta ciencia intentamos descubrir ciertos patrones en un gran conjunto de datos, es decir tiene como objetivo el poder obtener información y transformarla en una estructura compresible para su uso posterior en la enseñanza de la computadora.

Entonces después de haber explicado lo que sería aprendizaje automático y minería de datos, decir que en estas ciencias se centrará parte de nuestro proyecto, pasaremos a hablar de la base de datos que usaremos que en nuestro caso será un conjunto de datos en un fichero de formato Excel.

Este fichero será proporcionado por uno de los directores del proyecto (*Dr. Cristóbal Romero*). Este conjunto de información está basado en las notas de un grupo de alumnos de la asignatura de *Sistemas Inteligentes* de 2º del Grado en Ingeniería Informática de la Escuela Politécnica superior de Córdoba, las cuales han sido anonimizadas para mantener oculta la identidad de cada alumno. Todos los datos han sido recolectados por el Prof. Dr. Carlos García Martínez, profesor titular del Departamento de Informática y Análisis Numérico de la Escuela Politécnica Superior de la Universidad de Córdoba.

Este conjunto de datos no se basa solo en la nota obtenida en el examen final, sino que también podemos encontrar notas que el alumno ha ido obteniendo a lo largo de los meses que ha durado el periodo lectivo de la asignatura.

Además, tenemos ciertas variables las cuales consideramos como más importantes estas son, la nota de autoevaluación, esto es la evaluación que alguien hace de sí mismo o de algún aspecto o actividad que ha realizado, esta es una nota que cada alumno se pone antes de realizar el examen final. También tenemos otra nota que sería la que el alumno se pone justo después de la realización del examen, es decir calificación que espera obtener el alumno referente a cómo piensa que a echo su examen y también podemos encontrar una evaluación por pares, esta nota se atribuye por realizar una serie de actividades y por evaluar las actividades de sus compañeros.

La realización de estas bases de datos o estos informes de autoevaluación son cada vez más comunes entre el sector de la docencia, ya que se pueden utilizar para facilitar la labor del profesorado y reforzar el aprendizaje. Con esto los docentes pueden evaluar la información obtenida y usarla si lo creen pertinente en el cálculo de la nota final del alumnado matriculado en su curso.

Pero el uso de la autoevaluación no solo ayuda al docente a realizar la acción de calificar a los alumnos, sino que también le puede ser útil para detectar posibles fallos en la asignatura ya que puede observar que puntos le resulta más difíciles a los estudiantes, pudiendo entonces realizar cambios en la asignatura.

Entonces para finalizar podemos exponer que mediante el uso de la autoevaluación y el conjunto de calificaciones del alumnado podemos llegar a sustituir las calificaciones asignadas por un examen final, siempre y cuando ambas calificaciones tanto la del examen como la propuesta por el alumno lleguen a un punto las cuales sean muy similares. Si estas dos notas son muy diferentes entonces el profesorado debería usar el resto de las calificaciones que tuviera para el trabajo de calificar a los alumnos.

Sin embargo, este proceso, aunque parece fácil de aplicar no lo es, pudiendo llegar hasta un punto el cual se vuelva muy complejo. Sobre todo, si se tiene en cuenta el enorme número de alumnos que se puede tener y la cantidad de información que esto genera. Por esto surge la necesidad de utilizar algún tipo de técnica avanzada y automática como es la minería de datos.

# 2. Definición del problema

## Identificación del problema real

Mediante la realización de este proyecto pretendemos realizar una investigación utilizando minería de datos, para que el personal docente pueda analizar mediante una información recogida durante el curso el poder calificar al alumno mediante la técnica de autoevaluación.

Se va a realizar la investigación mediante tres enfoques o direcciones estas serían las siguientes:

- **Correlación** de los datos, es decir que relaciones de dependencia podemos encontrar entre todo el conjunto de datos que tenemos y cuales están más relacionadas entre sí.
- Mediante **regresión** por la cual utilizaremos diferentes algoritmos que nos proporciona Weka de los cuales ya hablaremos a lo largo de este documento.
- Mediante algoritmos de **clasificación**, lo cuales también no será proporcionados por el programa Weka.

Una parte importante de nuestro problema será los datos y el preprocesamiento de estos. La información que utilizaremos debe estar limpia de "basura", es decir, esta información debe ser filtrada antes de ser puesta al análisis o introducida en los algoritmos. Por decir un ejemplo de preprocesamiento podríamos decir que se han buscado y eliminado todo el ruido que pudiera tener nuestro fichero de datos, con ruido queremos decir datos erróneos o en blanco.

Ahora se va a hablar sobre el Dataset utilizado para que el lector tenga siempre claro con que datos hemos contado. Primero decir como ya se ha comentado antes que se trata de una serie de datos recogidos de un conjunto de alumnos y alumnas de la asignatura de *Sistemas Inteligentes* de 2º del Grado en Ingeniería Informática de la Escuela Politécnica superior de Córdoba, dado que los datos provienen de entornos educativos, a nuestro Dataset se le suele denominar Dataset educacional.

El Dataset consta de 43 atributos de entrada los cuales se pueden dividir como en dos secciones los datos obtenidos durante el curso y los obtenidos en el examen final.

A continuación, se describirán resumidamente todos estos atributos.

- **Identificador:** Es un número que utilizaremos para identificar cada conjunto de datos, a cada alumno, para no tener que usar el nombre ya que en todo el estudio se ha mantenido anónima la identidad de los alumnos.

-**Notas obtenidas durante el curso:**

- **NumPartTaller:** Esto sería el número de ejercicios que ha enviado el alumno durante el curso, con ejercicios nos referimos a pequeñas actividades mandadas por el profesor para ver si ha sido comprendido por los alumnos la materia.

- **NumPartEval:** Número de ejercicios que han evaluado los alumnos durante el curso, evalúan las actividades descritas antes de otros compañeros.

- **[Taller1:(envió)]:** Conjunto de notas obtenidas en cada uno de los diferentes ejercicios/actividades enviados durante el curso.

- **[Taller1:(Evaluaciones)]:** Conjunto de notas obtenidas por evaluar cada uno de los ejercicios/actividades de otros compañeros durante el curso, esta nota en parte es puesta por la plataforma Moodle mediante una de sus herramientas.

- **Media recibida:** Puntación media de las actividades, esta puntación se obtiene calculando la media aritmética de todo el conjunto de pruebas que se realiza durante el curso por el alumno. Cada actividad que hemos hablado antes es calificada con un número comprendido ente 0 y 100 (ambos inclusive). Una vez hemos obtenido el valor de todas las actividades es cuando hemos podido obtener el valor de este atributo aplicando una media aritmética.

- **Media de evaluación:** Este atributo se puede ver igual que el dato llamado “media recibida” pero en este hace referencia a las calificaciones por evaluar los trabajos de los compañeros.

- **Media talleres:** Nota final por las actividades enviadas y evaluadas esta nota se saca usando los dos atributos antes explicados y mediante unos porcentajes aplicados los cuales son propuestos por el profesor para su asignatura.

-**Notas obtenidas en el examen final:**

- **Nota Ex Teoría:** Es la nota que cada alumno ha obtenido a la realización de un examen oficial, esta nota puede tener un valor en el rango de 0 a 10 (ambos inclusive).

- **Nota de Heteroevaluación:** Se basa en la nota que se pone entre los diferentes alumnos por el trabajo que han realizado durante el curso, esta calificación está comprendida entre los valores de 0 y 10.

- **Autoevaluación:** Puntación propuesta por el alumno o la alumna, esta es la puntación que el alumno o alumna considera que se merece en la asignatura como calificación final del curso. El valor de esta puntación está comprendido entre los valores de 0 y 10 (ambos inclusive).

- **Nota después de examen:** Nota que el alumno predice que ha obtenido después de la realización del examen oficial, es un valor entre 0 y 10 (ambos incluidos).

- **[Test, 2.a.Bayer..]:** Es un conjunto de variables con diferentes nombres que hacen referencia a la calificación de las diferentes partes de las que consta el examen oficial, de todas estas calificaciones sale la nota de la variable descrita antes con el nombre "Nota Ex teoría", cada una de ellas es un valor comprendido entre 0 y 1 (ambos incluidos).

- **Azure:** Es un valor ente 0 y 100 el cual se ha sacado de la siguiente forma. Cada alumno escribe un texto justificando la nota que se ha puesto en el apartado de autoevaluación, pues entonces este valor se ha sacado usando la API llamada "Azure" [3] de la compañía Microsoft, esta API lo que nos proporciona es un valor referente al sentimiento, si el valor se acerca a 0 la opinión o sentimiento del texto introducido es muy negativo y cuanto más se acerca a 100 es lo contrario muy positivo.

Decir también que con el fin de proteger la privacidad de los/las alumnos/as y cumplir la Ley Orgánica de Protección de Datos todos los datos fueron desvinculados completamente de cualquier información que pudiera facilitar su identificación, tales como el nombre, apellidos, etcétera.

Como ya se comentó anteriormente para ejecutar los experimentos se ha utilizado el software de Minería de Datos Weka. Por lo tanto, nuestro Dataset debe de respetar su formato. Por lo tanto, en primer lugar, creamos un archivo con extensión .arff (Attribute-Relation File Format) donde, al comienzo, se añadió una cabecera con la información descriptiva tanto de los atributos de entrada como de la clase de salida. A continuación, se añadió un listado en la que cada fila contenía, separados por comas, los valores asociados a los atributos de entrada y a la clase de salida, de cada uno de los diferentes alumnos/as, en el mismo orden en el que estaban declarados en la cabecera.

La siguiente ilustración muestra el contenido del Dataset con los valores numéricos originales recopilados por el profesor de la asignatura, este Dataset será el utilizado para la parte de regresión:

```
@relation TFG

@attribute Num_PartT numeric
@attribute Num_EvalT numeric
@attribute Taller1 numeric
@attribute Taller2 numeric
@attribute Taller3 numeric
@attribute Taller4 numeric
@attribute Taller5 numeric
@attribute Taller6 numeric
@attribute Taller7 numeric
@attribute Taller8 numeric
@attribute Taller9 numeric
@attribute Media_Recibida numeric
@attribute Eval1 numeric
@attribute Eval2 numeric
@attribute Eval3 numeric
@attribute Eval4 numeric
@attribute Eval5 numeric
@attribute Eval6 numeric
@attribute Eval7 numeric
@attribute Eval8 numeric
@attribute Eval9 numeric
@attribute media_eval numeric
@attribute media_taller numeric
@attribute Test numeric
```

```
@attribute 2.a.Bayer numeric
@attribute 2.b.Bayer numeric
@attribute 2.c.Bayer numeric
@attribute 3.a.Logica numeric
@attribute 3.b.Logica numeric
@attribute 3.c.Logica numeric
@attribute Podaalfa numeric
@attribute Schank numeric
@attribute Sowa numeric
@attribute complejidad numeric
@attribute Marcos numeric
@attribute A* numeric
@attribute Bidireccional numeric
@attribute Nota_EX numeric
@attribute Hetero numeric
@attribute Auto numeric
@attribute Des_EX numeric
@attribute Azure numeric
```

@data

0,0,?,?,?,?,?,?,?,?,0,?,?,?,?,?,?,?,?,?,0,0,0,0.85,0.5,0.5,0.85,0.5,0,0,1,0.5,0,0.15,0.33,0.35,5.0,10.00,5.00,5.00,44

2,1,?,?,?,65.56,?,?,97.33,?,?,18.10,?,?,?,?,?,?,73,?,?,8.11,51.04,0.2,1,0,1,0.15,1,0.75,1,0.35,0.5,0,0.75,0.15,0,5.05,5.56,7.00,4.25,48

0,0,?,?,?,?,?,?,?,?,0,?,?,?,?,?,?,?,?,?,0,0,0.6,0.5,0.85,0,0.5,0.15,0,0,1,0.85,0,1,0.85,1,5,7.78,5,4,?

5,4,?,?,?,?,78.1,51.43,84.67,58.89,70.93,38.22,?,?,?,?,82.19,86.07,34.11,62.22,?,29.40,54.98,0.1,0,0.35,0.5,1,1,0.85,0,0.5,0.85,0,0.85,1, 0.5,6,4.44,6,4.75,49

5,5,?,90,79.17,73.57,62.14,?,51.56,?,?,39.60,?,43.6,73.46,88.41,100,?,50.97,?,?,39.61,58.62,0.2,1,1,0,1,0.5,0.35,0,1,0.85,0.85,0.85,0.35,1,8,5.56,8,6.80,41

4,3,87.38,69.17,?,69.76,59.05,?,?,?,?,31.71,?,99,71,?,27.96,78.24,?,?,?,?,22.88,53.86,0.6,1,0,1,1,1,1,0.15,0.5,1,1,1,1,8,1.11,8,7.75,74

2,1,?,?,?,?,?,?,?,58.44,82.47,?,15.66,?,?,?,?,?,?,?,76.43,?,?,8.49,52.12,0.5,0.5,0,1,0.85,0.5,0,0,1,0.15,0,0.15,0,1,5,0,5,3.75,53

1,1,?,?,?,79.37,?,?,?,?,?,8.82,?,?,?,70.71,?,?,?,?,?,7.86,45.95,0.2,1,0,1,0.85,0.85,1,0.15,1,0.85,0,1,1,0.5,5,1.11,5,3.75,43

2,1,42.14,38.75,?,?,?,?,?,?,?,8.99,?,98.13,?,?,?,?,?,?,?,10.90,62.47,0.1,0,0,0.5,1,1,0.35,0,0.85,0.5,0.85,0.85,0,0.35,5.62,1.11,6.50,4.25,?

1,0,77.38,?,?,?,?,?,?,?,?,8.60,?,?,?,?,?,?,?,?,?,?,0,0,0.8,0,0,0,0,0,0,0,0,0,0,0,0,1,1.11,?,2.50,?

1,0,53.33,?,?,?,?,?,?,?,?,5.93,?,?,?,?,?,?,?,?,?,?,0,0,0.3,0,0,0,0,0,0,0,1,0.35,0.5,0.15,0.15,0.35,2.73,7.78,?,4.25,?

4,1,43.97,59.17,?,55.95,?,52.54,?,?,?,23.51,?,?,?,97.4,?,?,?,?,?,10.82,67.85,0,0,0,1,0.15,0.15,0,0,0.5,0.5,0,0.85,1,0.5,5,4.44,5,5,29

2,1,87.86,?,42.5,?,?,?,?,?,?,14.48,100,?,?,?,?,?,?,?,?,11.11,65.79,0.2,0,1,0,0.35,1,0,1,1,0.5,0,0.15,1,1,5.33,3.33,7,6.75,48

7,6,75.71,66.25,92.5,54.29,51.9,77.14,93.67,?,?,56.83,100,100,75.39,72.96,59.97,78.85,?,?,?,54.13,71.45,0.8,1,1,0,1,1,1,1,1,0.85,0.85,1,1,1,8,2.22,8,8.50,58

7,7,0,85,88.75,74.92,?,39.29,83.33,55.06,?,47.37,66.67,81.25,80.58,87.72,?,100,75,98.21,?,65.49,69.47,0.4,1,1,1,1,0.5,1,1,1,1,0.35,1,1,5,4.44,5,6.5,53

7,6,?,78.75,60,74.44,?,72.38,89.56,62.72,73.33,56.80,?,100,71.25,100,?,33.33,79.58,54.83,?,48.78,66.62,0.1,0,1,0,1,0.5,0.5,1,0.5,0.85,1,0.5,0.15,0.85,6.50,0,6.50,5,67

3,3,68.41,50,60,?,?,?,?,?,?,19.82,70.1,100,100,?,?,?,?,?,?,30.01,61.95,0.6,0,0,0,1,1,0.85,1,1,1,0,0.85,1,1,8,4.44,8,6.50,71

1,1,?,?,?,?,?,?,84,?,?,9.33,?,?,?,?,?,?,87.77,?,?,9.75,56.40,0.5,0.5,1,1,1,1,1,0,1,0.85,0,0.95,0.15,0.5,6.99,0,9,7,57

0,0,?,?,?,?,?,?,?,?,?,0,?,?,?,?,?,?,?,?,?,?,0,0,0.4,1,1,1,0.85,1,0.85,1,0.85,0.85,0.85,0.35,1,1,8.11,1.11,8.50,7.75,43

1,1,?,?,?,?,?,?,86.89,?,?,9.65,?,?,?,?,?,?,92.57,?,?,10.29,59.40,0.2,1,0.85,1,0.5,0.5,0.5,0,1,0.5,0,0.85,0.35,0.5,5,1.11,5,3,7

Lo que hemos podido ver anteriormente es una pequeña muestra de cómo están definidas las instancias en el fichero de datos, entonces como tenemos 92 instancias ponerlas todas sería rellenar este documento con demasiadas hojas que serían prácticamente casi iguales, por lo tanto vamos a poner los estadísticos y como se distribuyen cada una de las 42 variables para intentar dejar lo más claro posible como son los valores de nuestras variables.

- **Num_Part**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **9** |
| **Media** | **3.185** |
| **Desviación** | **2.874** |
| **Valores perdidos** | **0%** |

*Tabla 1:Estadísticos de Num_Part.*

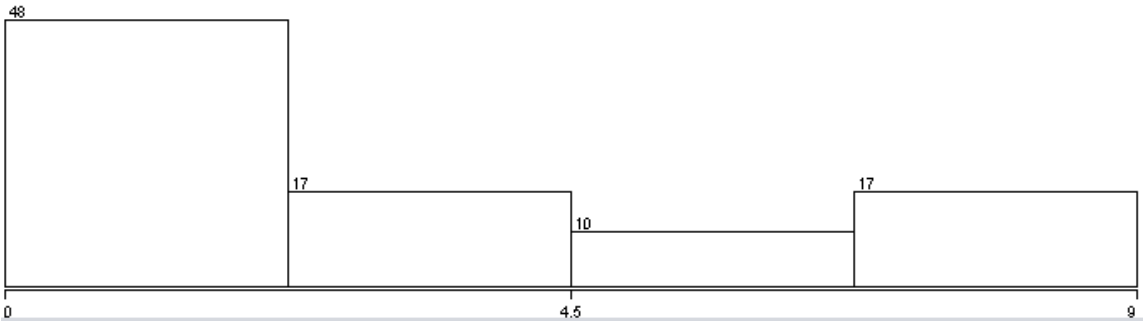



*Ilustración 1:Distribución de Num_Part.*

- **Num_Eval**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **9** |
| **Media** | **2.598** |
| **Desviación** | **2.762** |
| **Valores perdidos** | **0%** |

*Tabla 2:Estadísticos de Num_Eval.*

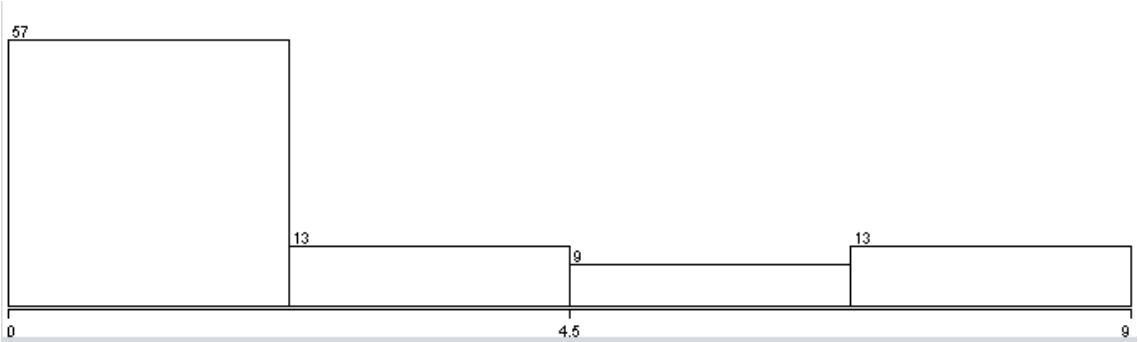



*Ilustración 2:Distribución de Num_Eval.*

- **Taller1**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **97.14** |
| **Media** | **61.717** |
| **Desviación** | **24.544** |
| **Valores perdidos** | **52%** |

*Tabla 3:Estadísticos de Taller1.*

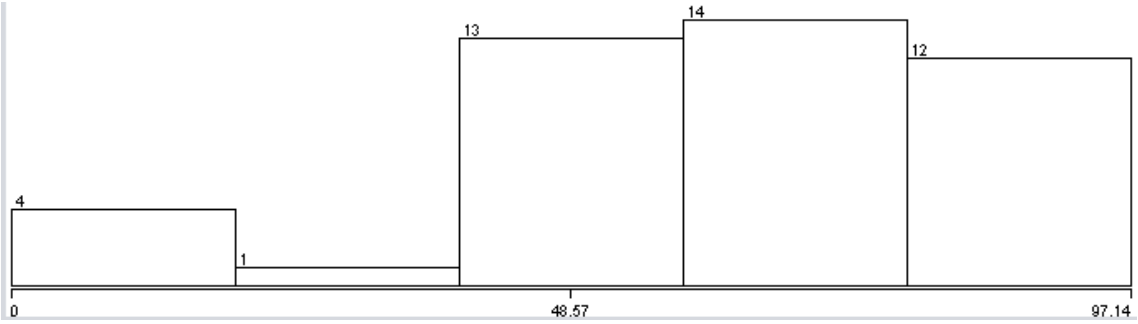



*Ilustración 3:Distribución de Taller1.*

- **Taller2**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **100** |
| **Media** | **67.572** |
| **Desviación** | **18.526** |
| **Valores perdidos** | **43%** |

*Tabla 4:Estadísticos de Taller2.*

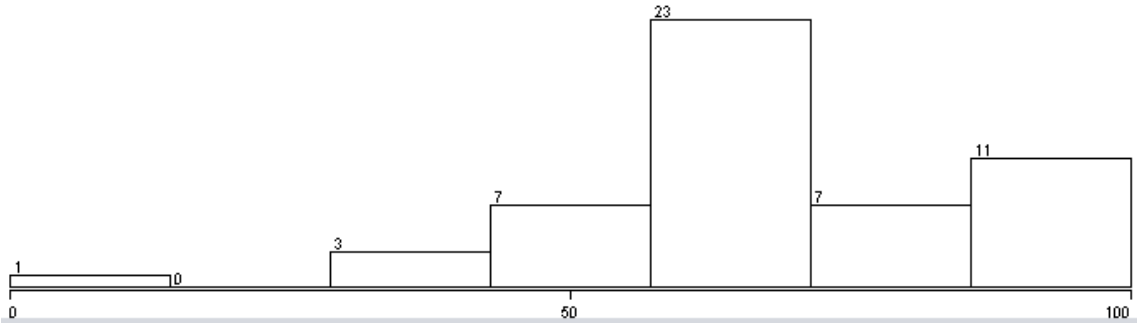



*Ilustración 4:Distribución de Taller2.*

- **Taller3**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **92.5** |
| **Media** | **70.327** |
| **Desviación** | **19.039** |
| **Valores perdidos** | **58%** |

*Tabla 5:Estadísticos de Taller3.*

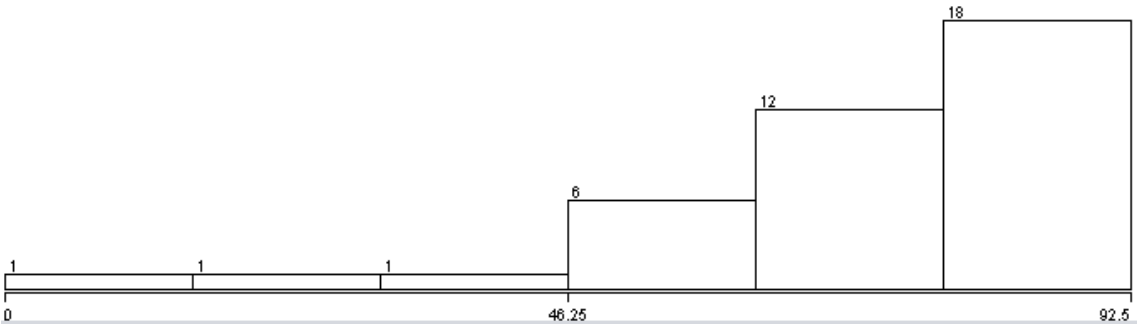



*Ilustración 5:Distribución de Taller3.*

- **Taller4**

| | |
|---|---|
| **Mínimo** | **29.37** |
| **Máximo** | **97.14** |
| **Media** | **73.002** |
| **Desviación** | **16.010** |
| **Valores perdidos** | **63%** |

*Tabla 6:Estadísticos de Taller4*

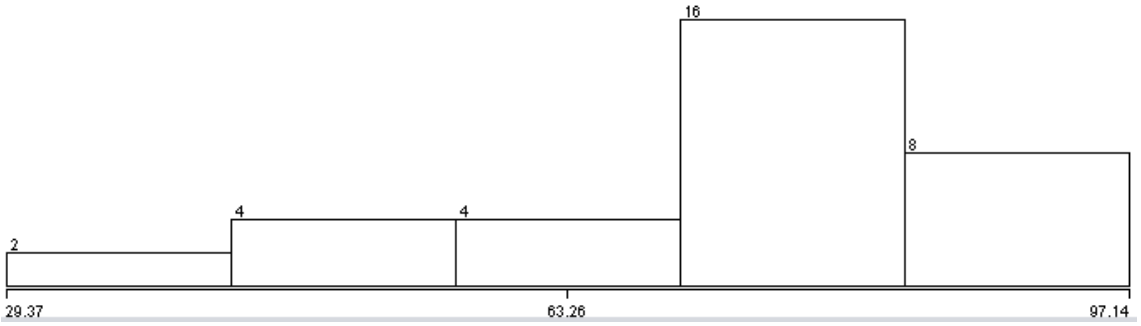



*Ilustración 6:Distribución de Taller4.*

- **Taller5**

| | |
|---|---|
| **Mínimo** | **2.38** |
| **Máximo** | **96.43** |
| **Media** | **66.09** |
| **Desviación** | **19.47** |
| **Valores perdidos** | **75%** |

*Tabla 7:Estadísticos de Taller5.*

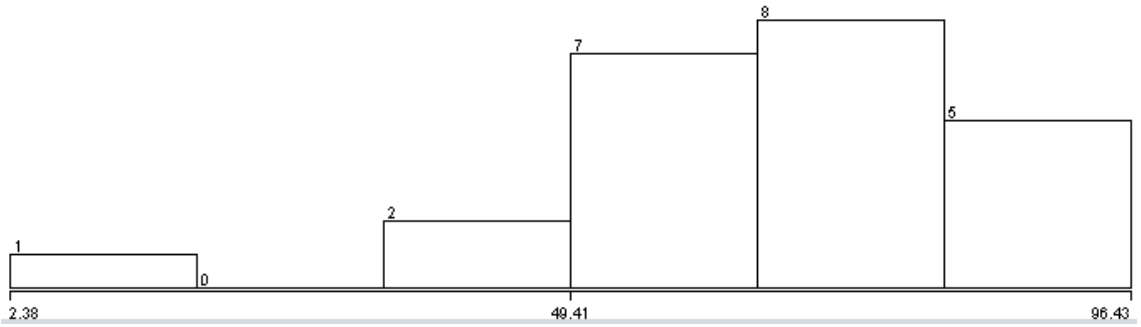



*Ilustración 7:Distribución de Taller5.*

- **Taller6**

| | |
|---|---|
| **Mínimo** | **38.33** |
| **Máximo** | **95.71** |
| **Media** | **61.57** |
| **Desviación** | **15.32** |
| **Valores perdidos** | **64%** |

*Tabla 8:Estadísticos de Taller6*

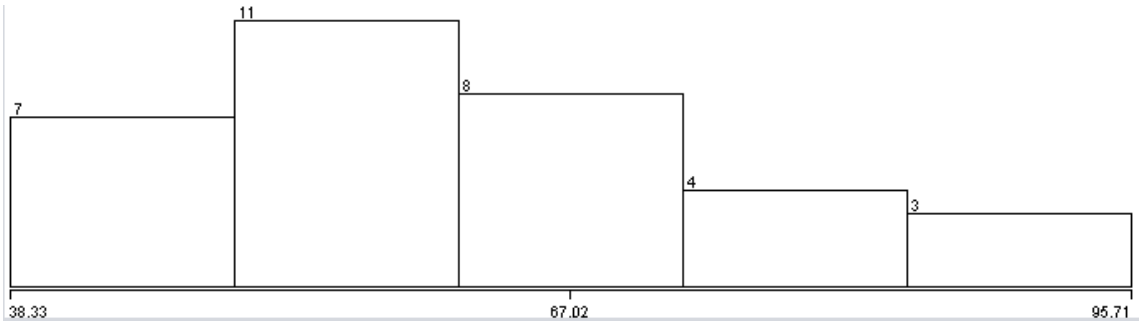



*Ilustración 8:Distribución de Taller6.*

- **Taller7**

| | |
|---|---|
| **Mínimo** | **29.67** |
| **Máximo** | **100** |
| **Media** | **79.16** |
| **Desviación** | **17.005** |
| **Valores perdidos** | **65%** |

*Tabla 9:Estadísticos de Taller7.*

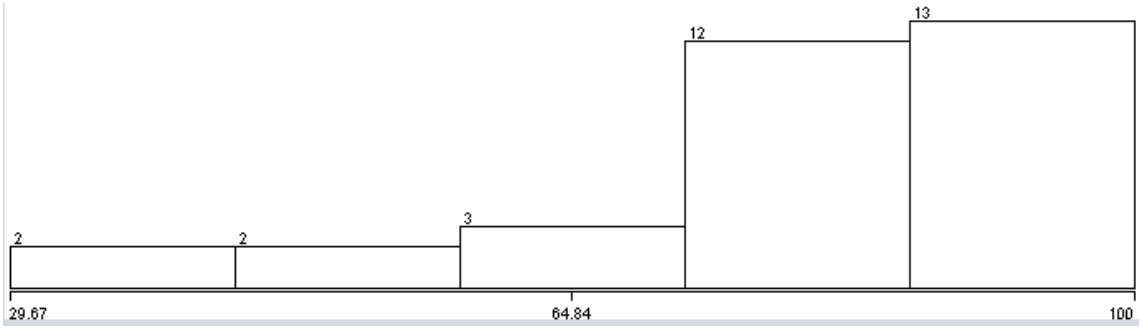



*Ilustración 9:Distribución de Taller7.*

- **Taller8**

| | |
|---|---|
| **Mínimo** | **30.93** |
| **Máximo** | **98.33** |
| **Media** | **71.14** |
| **Desviación** | **19.51** |
| **Valores perdidos** | **80%** |

*Tabla 10:Estadísticos de Taller8.*

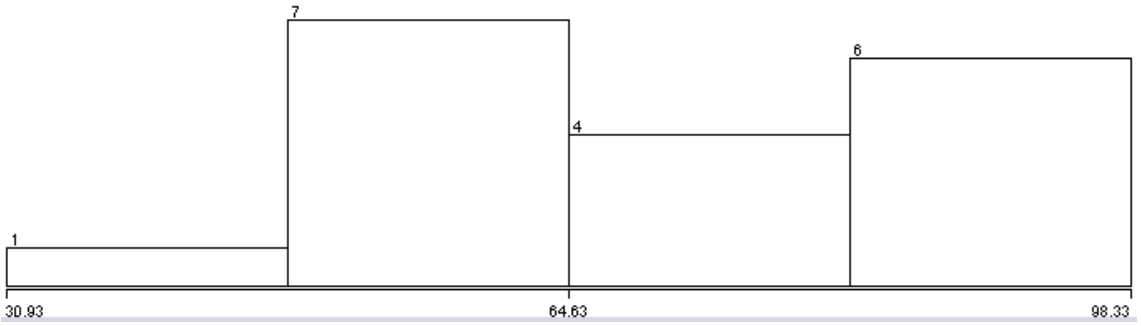



*Ilustración 10:Distribución de Taller8.*

- **Taller9**

| | |
|---|---|
| **Mínimo** | **0.55** |
| **Máximo** | **99.44** |
| **Media** | **61.28** |
| **Desviación** | **24.48** |
| **Valores perdidos** | **80%** |

*Tabla 11:Estadísticos de Taller9.*

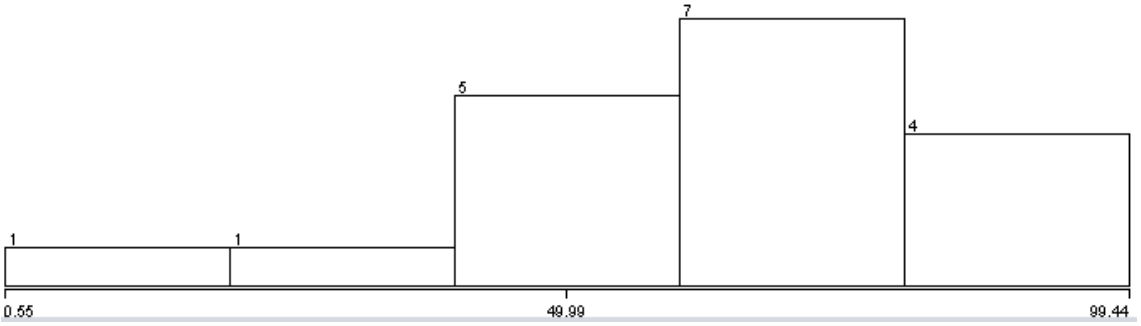



*Ilustración 11:Distribución de Taller9.*

- **Media_Recibida**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **91.10** |
| **Media** | **24.22** |
| **Desviación** | **24.18** |
| **Valores perdidos** | **0%** |

*Tabla 12:Estadísticos de Media_Recibida.*

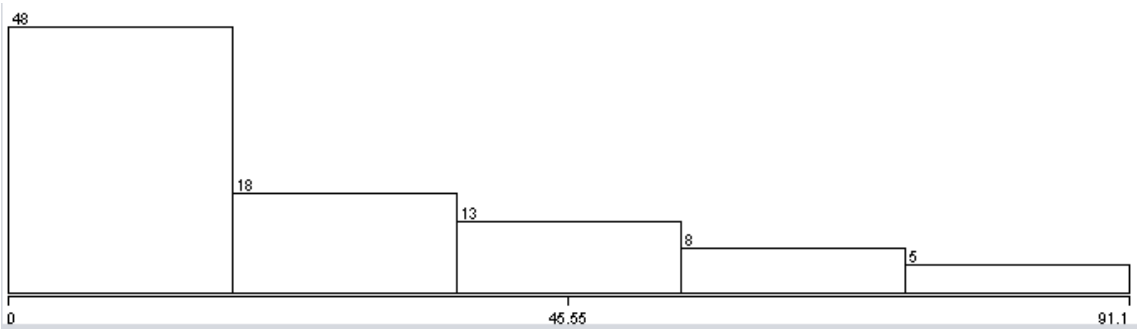



*Ilustración 12:Distribución de Media_Recibida.*

- **Eval1**

| | |
|---|---|
| **Mínimo** | **22.87** |
| **Máximo** | **100** |
| **Media** | **79.12** |
| **Desviación** | **16.32** |
| **Valores perdidos** | **63%** |

*Tabla 13::Estadísticos de Eval1.*

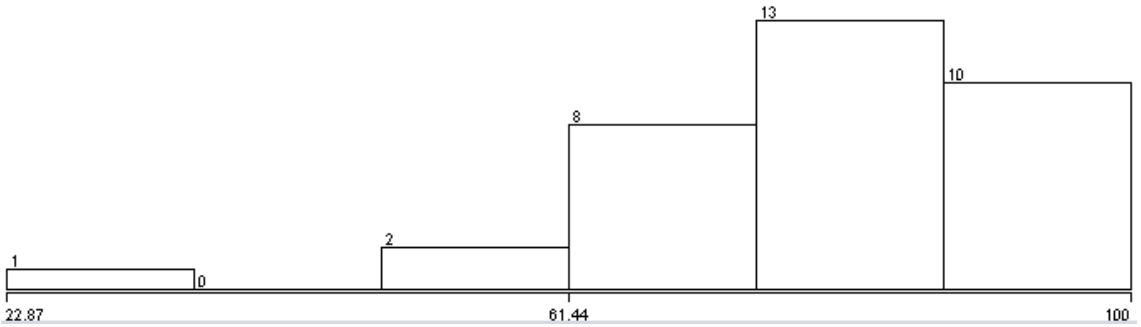



*Ilustración 13:Distribución de Eval1.*

- **Eval2**

| | |
|---|---|
| **Mínimo** | **33.33** |
| **Máximo** | **100** |
| **Media** | **85.072** |
| **Desviación** | **19.713** |
| **Valores perdidos** | **60%** |

*Tabla 14:Estadísticos de Eval2.*

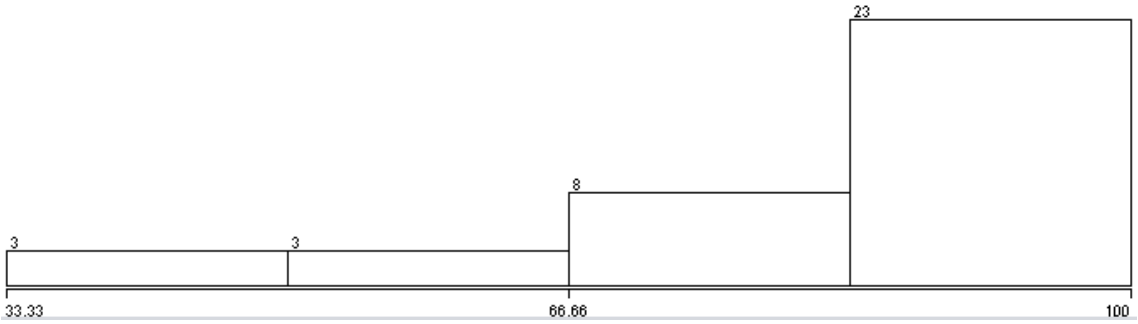



*Ilustración 14:Distribución de Eval2.*

- **Eval3**

| | |
|---|---|
| **Mínimo** | **33.81** |
| **Máximo** | **100** |
| **Media** | **75.569** |
| **Desviación** | **19.011** |
| **Valores perdidos** | **61%** |

*Tabla 15:Estadísticos de Eval3.*

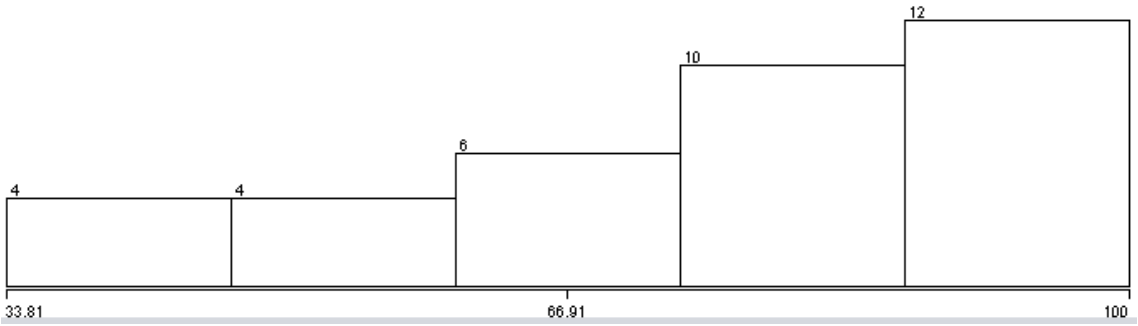



*Ilustración 15:Distribución de Eval3.*

- **Eval4**

| | |
|---|---|
| **Mínimo** | **43.46** |
| **Máximo** | **100** |
| **Media** | **79.663** |
| **Desviación** | **15.848** |
| **Valores perdidos** | **71%** |

*Tabla 16:Estadísticos de Eval4.*

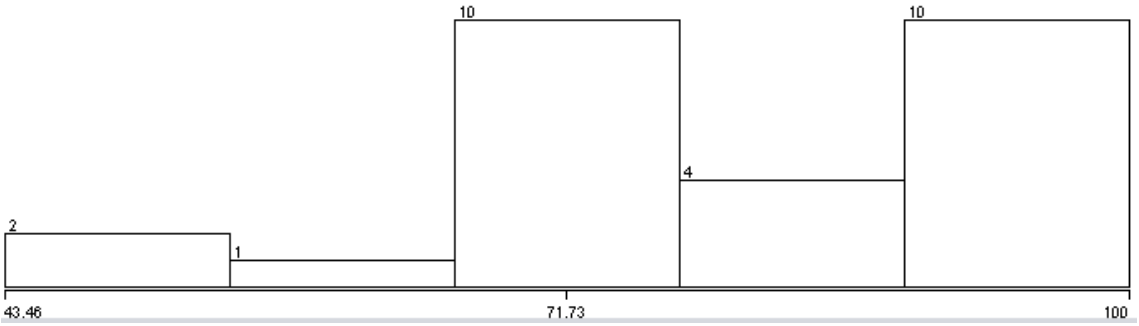



*Ilustración 16:Distribución de Eval4.*

- **Eval5**

| | |
|---|---|
| **Mínimo** | **27.96** |
| **Máximo** | **100** |
| **Media** | **72.78** |
| **Desviación** | **22.22** |
| **Valores perdidos** | **78%** |

*Tabla 17:Estadísticos de Eval5.*

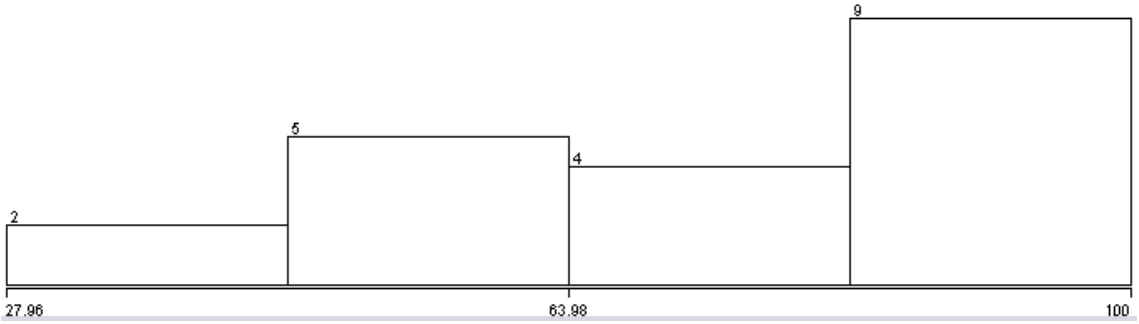



*Ilustración 17:Distribución de Eval5.*

- **Eval6**

| | |
|---|---|
| **Mínimo** | **29.32** |
| **Máximo** | **100** |
| **Media** | **75.503** |
| **Desviación** | **21.25** |
| **Valores perdidos** | **66%** |

*Tabla 18:Estadísticos de Eval6.*

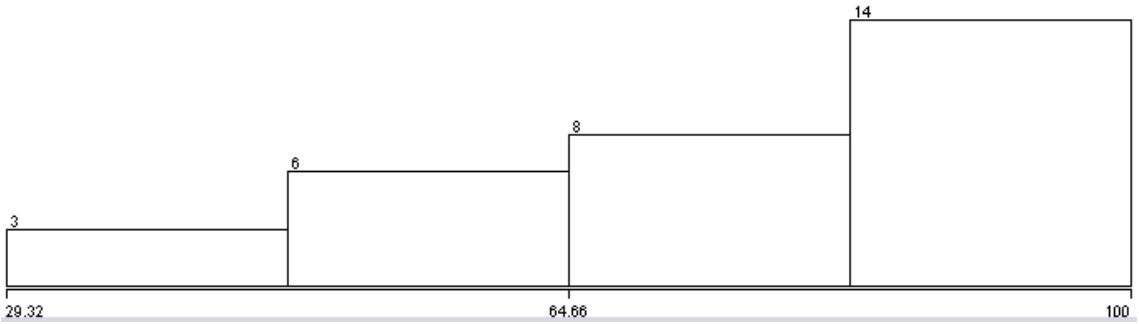



*Ilustración 18:Distribución de Eval6.*

- **Eval7**

| | |
|---|---|
| **Mínimo** | **34.11** |
| **Máximo** | **100** |
| **Media** | **78.214** |
| **Desviación** | **19.369** |
| **Valores perdidos** | **67%** |

*Tabla 19:Estadísticos de Eval7.*

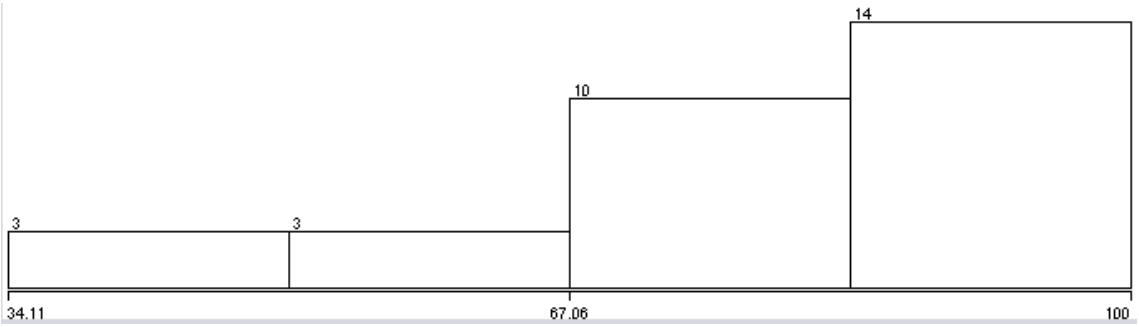



*Ilustración 19:Distribución de Eval7.*

- **Eval8**

| | |
|---|---|
| **Mínimo** | **37.22** |
| **Máximo** | **100** |
| **Media** | **82.187** |
| **Desviación** | **21.026** |
| **Valores perdidos** | **84%** |

*Tabla 20:Estadísticos de Eval8.*

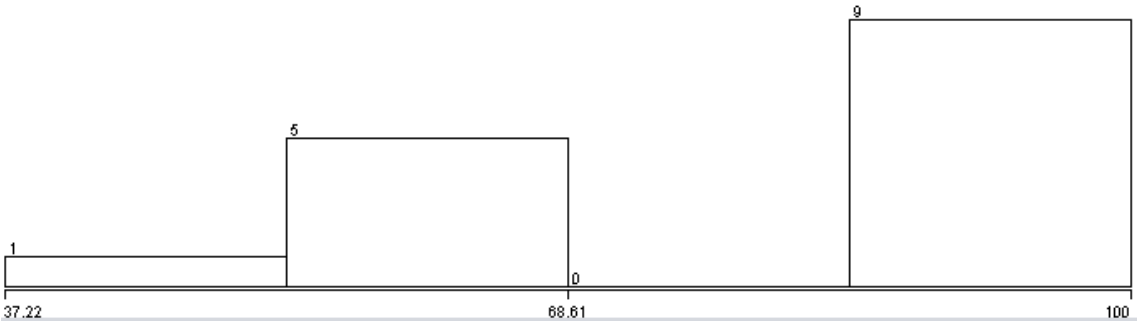



*Ilustración 20:Distribución de Eval8.*

- **Eval9**

| | |
|---|---|
| **Mínimo** | **66.67** |
| **Máximo** | **100** |
| **Media** | **92.695** |
| **Desviación** | **11.242** |
| **Valores perdidos** | **89%** |

*Tabla 21:Estadísticos de Eval9.*

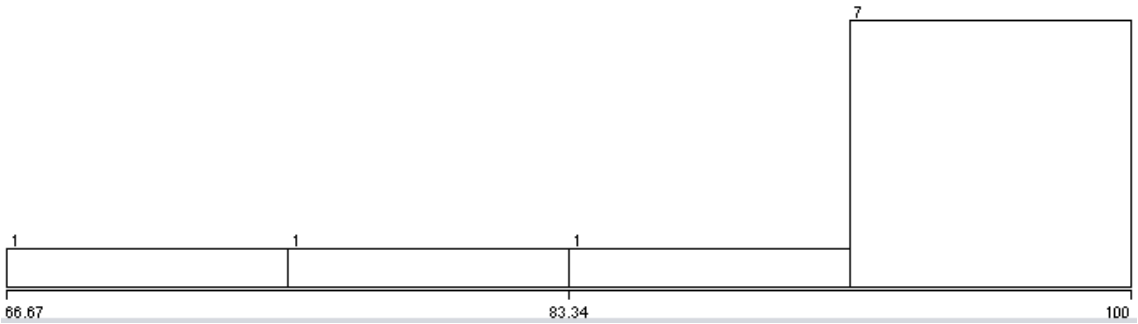



*Ilustración 21:Distribución de Eval9.*

- **Media_eval**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **86.91** |
| **Media** | **22.876** |
| **Desviación** | **24.958** |
| **Valores perdidos** | **1%** |

*Tabla 22:Estadísticos de Media_eval.*

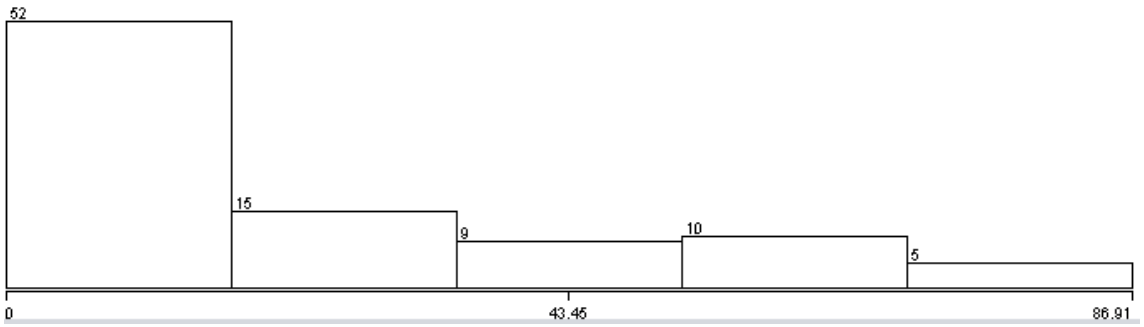



*Ilustración 22:Distribución de Media_eval.*

- **Media_taller**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **83.61** |
| **Media** | **42.633** |
| **Desviación** | **29.763** |
| **Valores perdidos** | **0%** |

*Tabla 23:Estadísticos de Media_taller.*

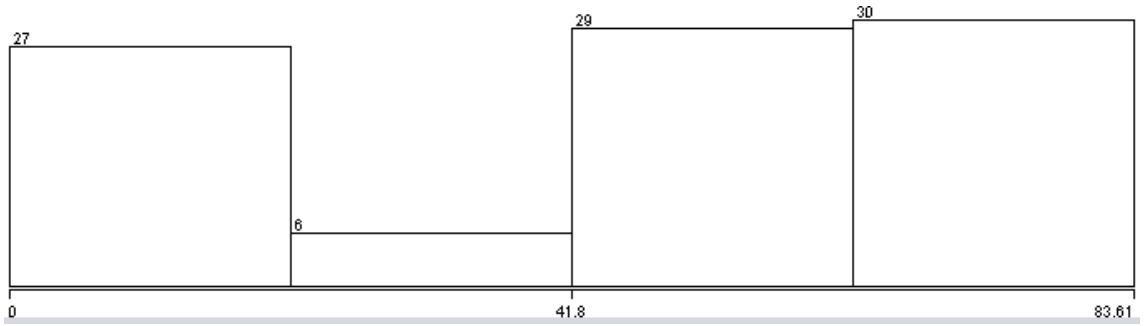



*Ilustración 23:Distribución de Media_taller.*

- **Test**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **1** |
| **Media** | **0.33** |
| **Desviación** | **0.251** |
| **Valores perdidos** | **0%** |

*Tabla 24:Estadísticos de Test.*

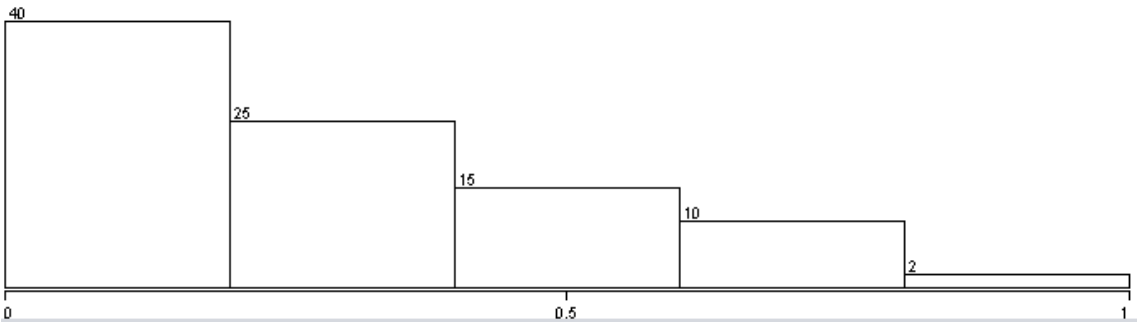



*Ilustración 24:Distribución de Test.*

- **2.a.Bayer**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **1** |
| **Media** | **0.537** |
| **Desviación** | **0.454** |
| **Valores perdidos** | **0%** |

*Tabla 25:Estadísticos de 2.a.Bayer.*

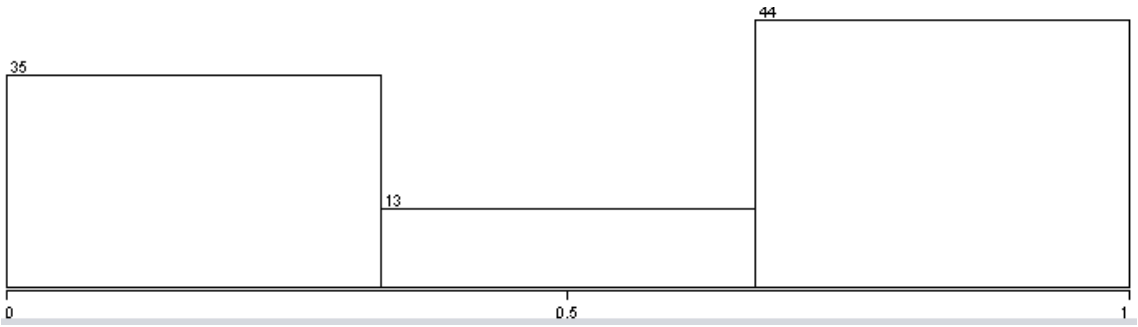



*Ilustración 25:Distribución de 2.a.Bayer.*

- **2.b.Bayer**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **1** |
| **Media** | **0.487** |
| **Desviación** | **0.439** |
| **Valores perdidos** | **0%** |

*Tabla 26:Estadísticos de 2.b.Bayer.*

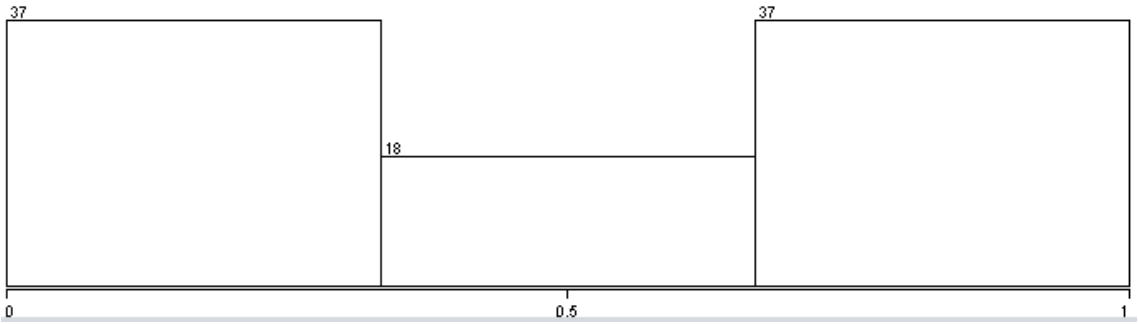



*Ilustración 26:Distribución de 2.b.Bayer.*

- **2.c.Bayer**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **1** |
| **Media** | **0.422** |
| **Desviación** | **0.474** |
| **Valores perdidos** | **0%** |

*Tabla 27:Estadísticos de 2.c.Bayer.*

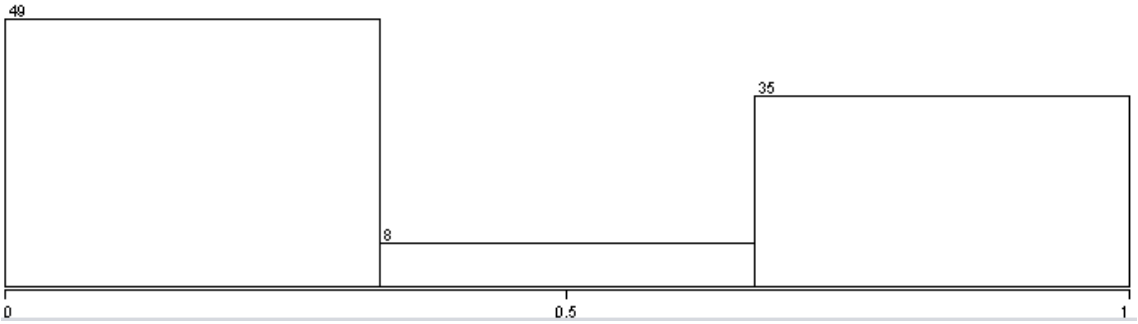



*Ilustración 27:Distribución de 2.c.Bayer.*

- **3.a.Logica**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **1** |
| **Media** | **0.645** |
| **Desviación** | **0.404** |
| **Valores perdidos** | **0%** |

*Tabla 28:Estadísticos de 3.a.Logica.*

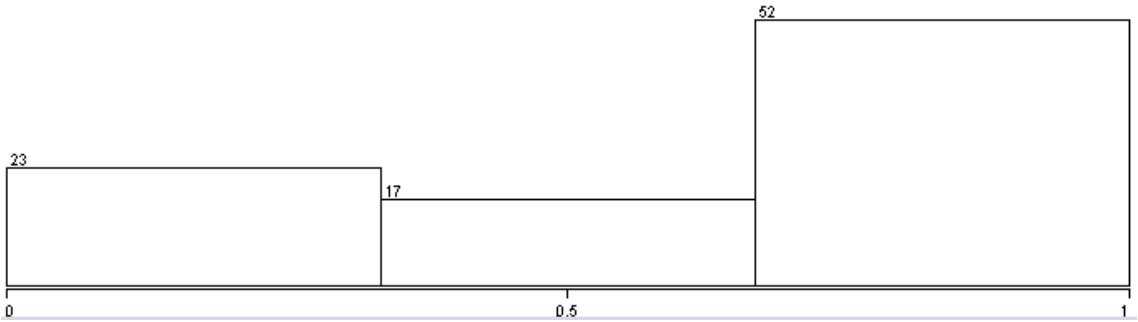



*Ilustración 28:Distribución de 3.a.Logica.*

- **3.b.Logica**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **1** |
| **Media** | **0.683** |
| **Desviación** | **0.393** |
| **Valores perdidos** | **0%** |

*Tabla 29:Estadísticos de 3.b.Logica.*

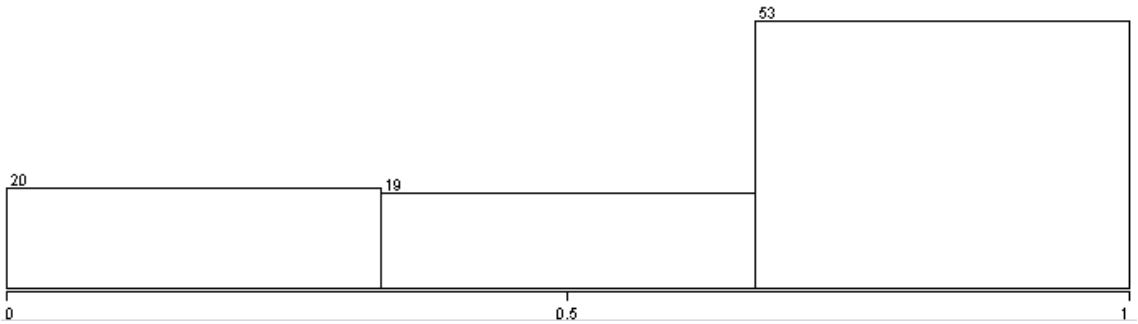



*Ilustración 29:Distribución de 3.b.Logica.*

- **3.c.Logica**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **1** |
| **Media** | **0.413** |
| **Desviación** | **0.425** |
| **Valores perdidos** | **0%** |

*Tabla 30:Estadísticos de 3.c.Logica.*

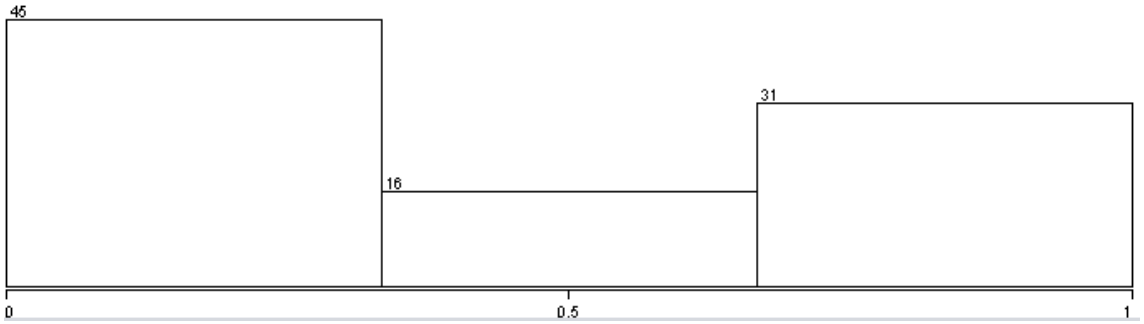



*Ilustración 30:Distribución de 3.c.Logica.*

- **Podaalfa**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **1** |
| **Media** | **0.305** |
| **Desviación** | **0.452** |
| **Valores perdidos** | **0%** |

*Tabla 31:Estadísticos de Podaalfa.*

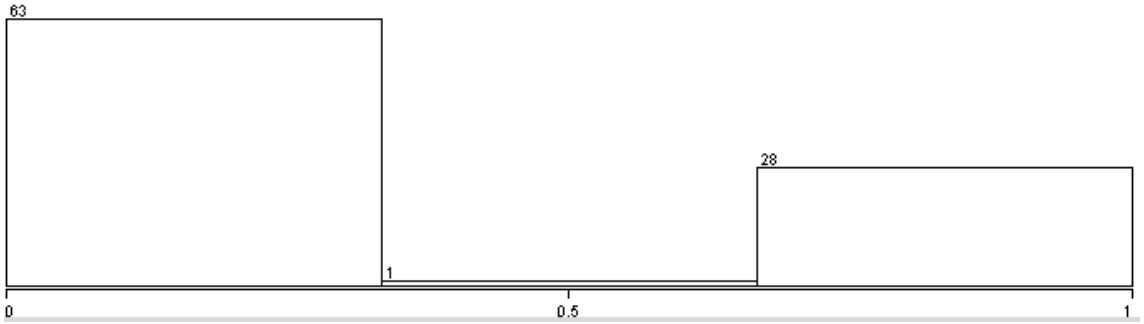



*Ilustración 31:Distribución de Podaalfa.*

- **Schank**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **1** |
| **Media** | **0.712** |
| **Desviación** | **0.344** |
| **Valores perdidos** | **0%** |

*Tabla 32:Estadísticos de schank.*

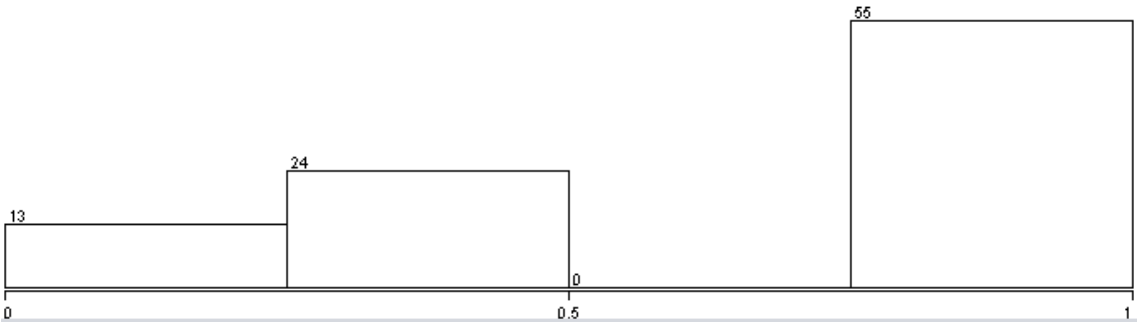



*Ilustración 32:Distribución de schank.*

- **Sowa**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **1** |
| **Media** | **0.657** |
| **Desviación** | **0.322** |
| **Valores perdidos** | **0%** |

*Tabla 33:Estadísticos de Sowa.*

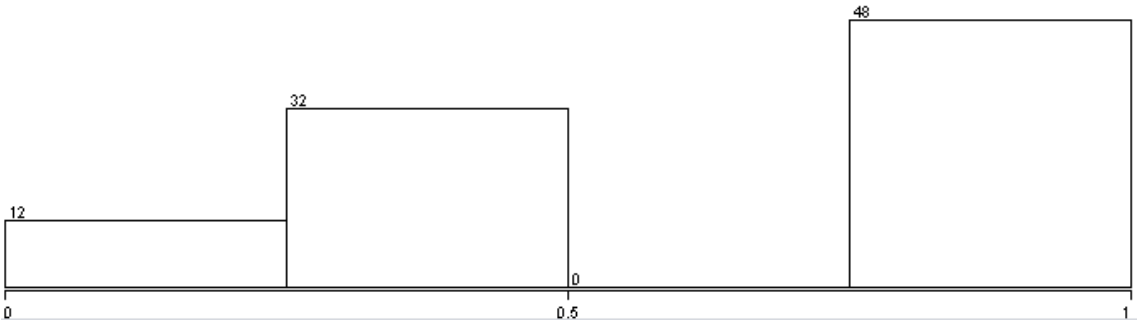



*Ilustración 33:Distribución de Sowa.*

- **Complejidad**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **1** |
| **Media** | **0.358** |
| **Desviación** | **0.448** |
| **Valores perdidos** | **0%** |

*Tabla 34:Estadísticos de Complejidad.*

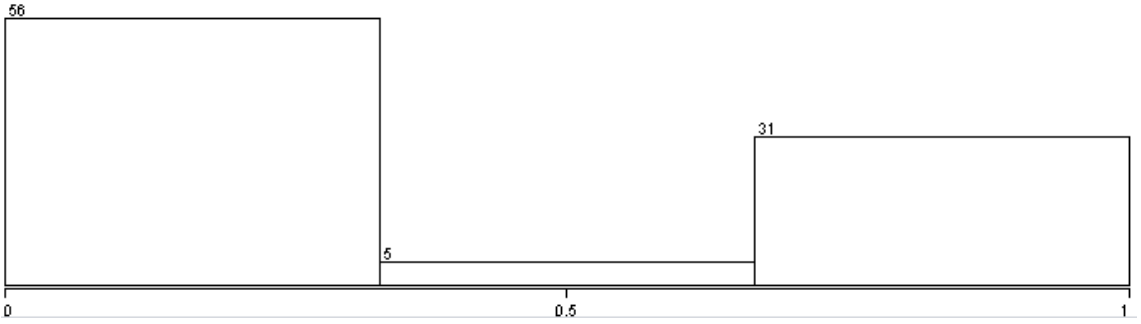



*Ilustración 34:Distribución de Complejidad.*

- **Marcos**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **1** |
| **Media** | **0.553** |
| **Desviación** | **0.342** |
| **Valores perdidos** | **0%** |

*Tabla 35:Estadísticos de Marcos*

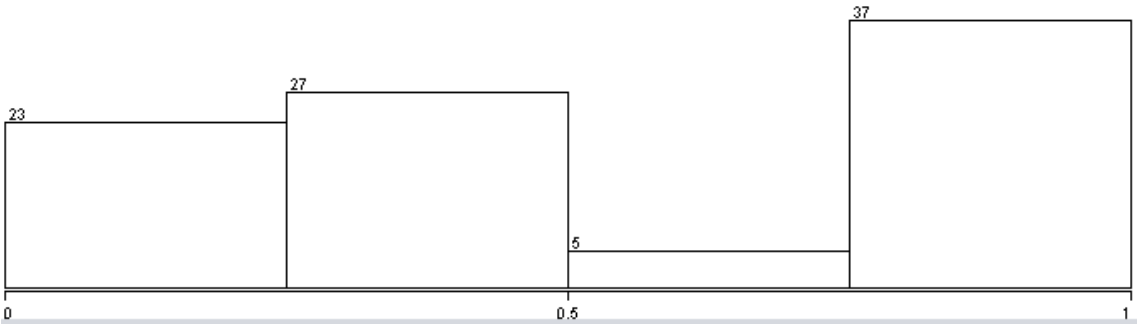



*Ilustración 35:Distribución de Marcos.*

- **A***

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **1** |
| **Media** | **0.488** |
| **Desviación** | **0.411** |
| **Valores perdidos** | **0%** |

*Tabla 36:Estadísticos de A*.*

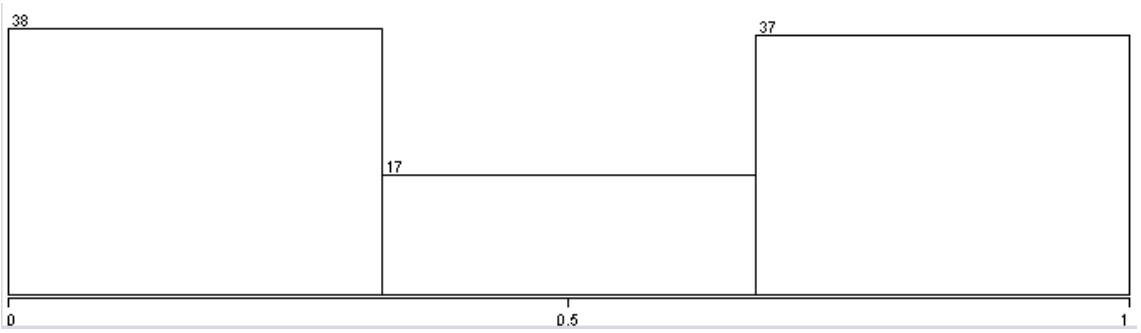



*Ilustración 36:Distribución de A*.*

- **Bidireccional**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **1** |
| **Media** | **0.592** |
| **Desviación** | **0.343** |
| **Valores perdidos** | **0%** |

*Tabla 37:Estadísticos de Bidireccional.*

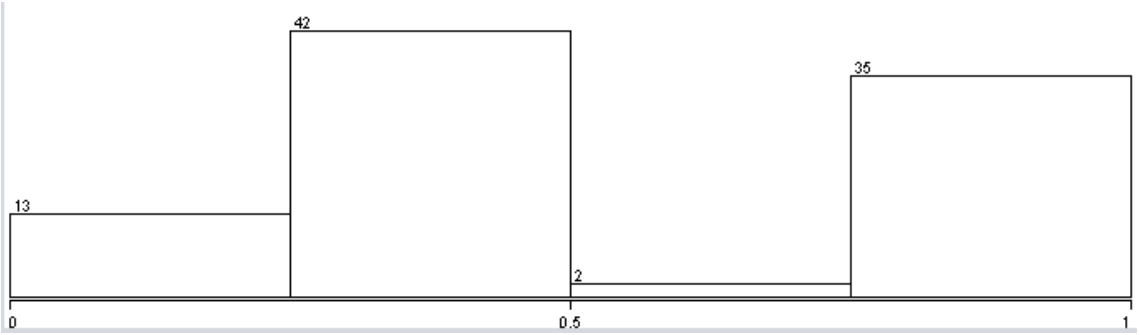



*Ilustración 37:Distribución de Bidireccional.*

- **Nota_Ex**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **10** |
| **Media** | **5.181** |
| **Desviación** | **2.043** |
| **Valores perdidos** | **0%** |

*Tabla 38:Estadísticos de Nota_Ex.*

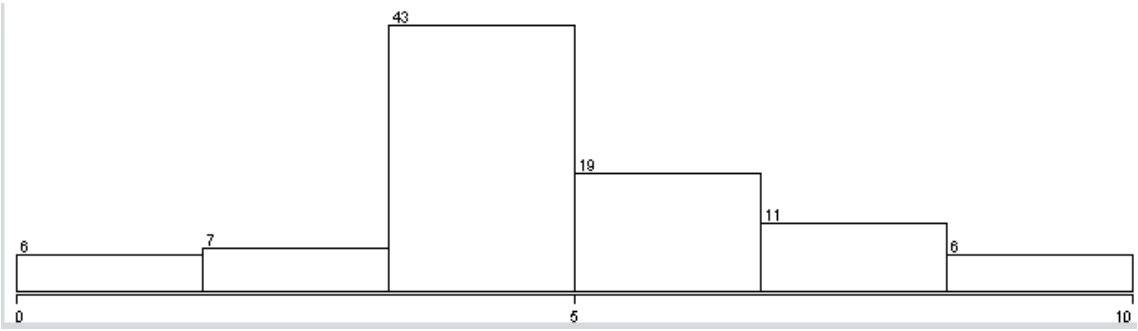



*Ilustración 38:Distribución de Nota_Ex.*

- **Hetero**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **10** |
| **Media** | **3.251** |
| **Desviación** | **3.362** |
| **Valores perdidos** | **0%** |

*Tabla 39:Estadísticos de Hetero.*

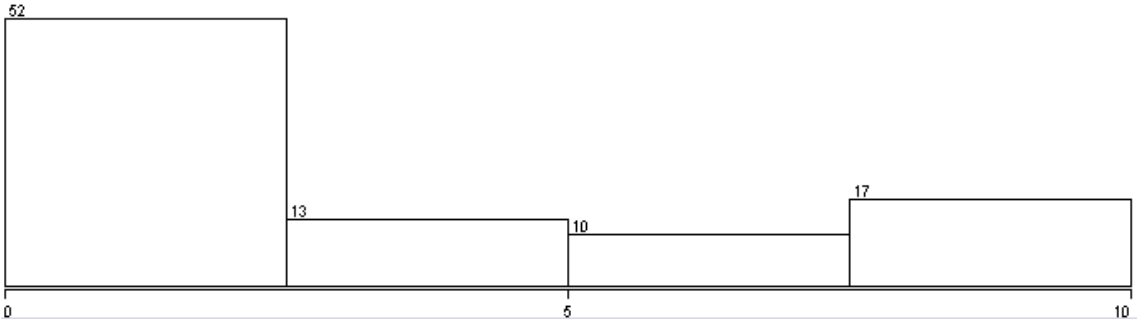



*Ilustración 39:Distribución de Hetero.*

- **Auto**

| | |
|---|---|
| **Mínimo** | **0** |
| **Máximo** | **10** |
| **Media** | **6.091** |
| **Desviación** | **1.763** |
| **Valores perdidos** | **11%** |

*Tabla 40:Estadísticos de Auto.*

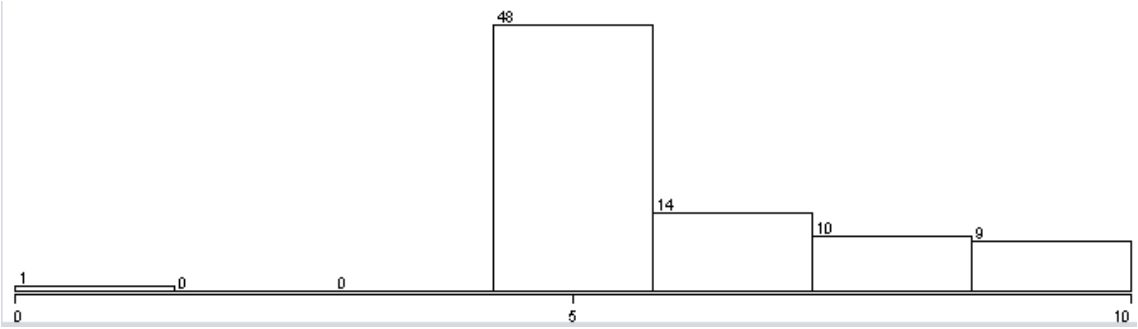



*Ilustración 40:Distribución de Auto.*

- **Des_EX**

| | |
|---|---|
| **Mínimo** | **2.5** |
| **Máximo** | **9** |
| **Media** | **5.239** |
| **Desviación** | **1.451** |
| **Valores perdidos** | **7%** |

*Tabla 41:Estadísticos de Des_Ex.*

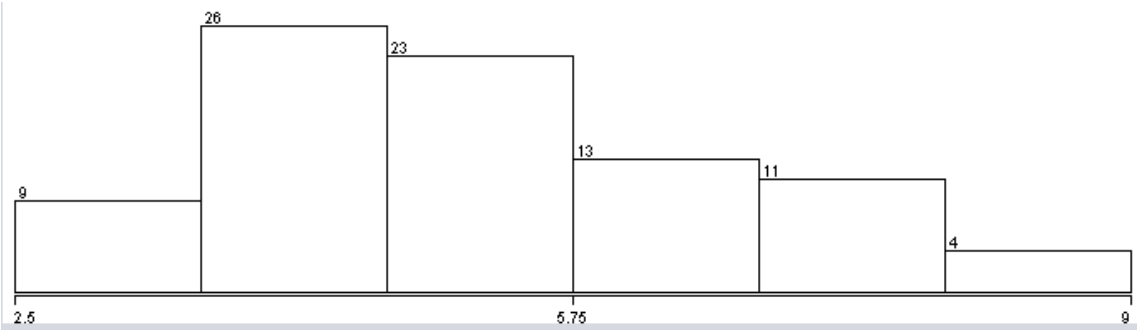



*Ilustración 41:Distribución de Des_Ex.*

- **Azure**

| | |
|---|---|
| **Mínimo** | **6.5** |
| **Máximo** | **74** |
| **Media** | **50.586** |
| **Desviación** | **12.069** |
| **Valores perdidos** | **17%** |

*Tabla 42:Estadísticos de Azure.*

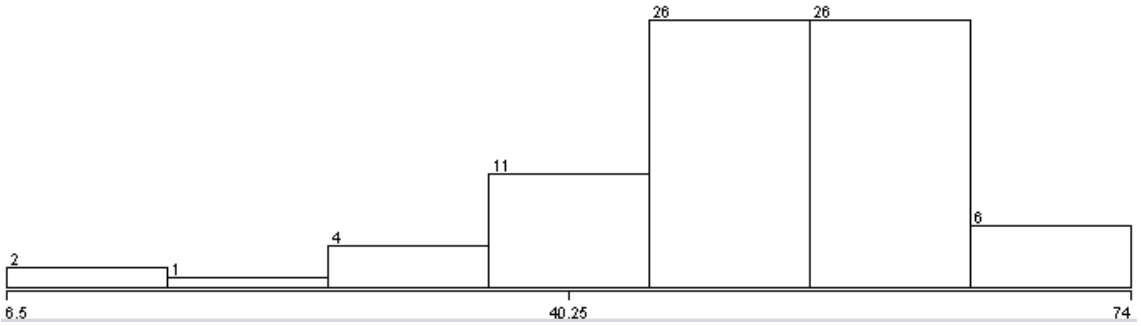



*Ilustración 42:Distribución de Azure.*

Para la parte de la investigación de clasificación lo que hemos hecho es añadir una nueva variable la cual solo podrá tomar valores de “SI” o “NO” quedando de la siguiente manera nuestro fichero:

```
@relation TFG

@attribute Num_PartT numeric
@attribute Num_EvalT numeric
@attribute Taller1 numeric
@attribute Taller2 numeric
@attribute Taller3 numeric
@attribute Taller4 numeric
@attribute Taller5 numeric
@attribute Taller6 numeric
@attribute Taller7 numeric
@attribute Taller8 numeric
@attribute Taller9 numeric
@attribute Media_Recibida numeric
@attribute Eval1 numeric
@attribute Eval2 numeric
@attribute Eval3 numeric
@attribute Eval4 numeric
@attribute Eval5 numeric
@attribute Eval6 numeric
@attribute Eval7 numeric
@attribute Eval8 numeric
@attribute Eval9 numeric
@attribute media_eval numeric
@attribute media_taller numeric
@attribute Test numeric
```

```
@attribute 2.a.Bayer numeric
@attribute 2.b.Bayer numeric
@attribute 2.c.Bayer numeric
@attribute 3.a.Logica numeric
@attribute 3.b.Logica numeric
@attribute 3.c.Logica numeric
@attribute Podaalfa numeric
@attribute Schank numeric
@attribute Sowa numeric
@attribute complejidad numeric
@attribute Marcos numeric
@attribute A* numeric
@attribute Bidireccional numeric
@attribute Nota_EX numeric
@attribute Hetero numeric
@attribute Auto numeric
@attribute Des_EX numeric
@attribute Azure numeric
@attribute Creer { SI, NO}
```

@data

0,0,?,?,?,?,?,?,?,?,?,0,?,?,?,?,?,?,?,?,?,?,0,0,0,0.85,0.5,0.5,0.85,0.5,0,0,1,0.5,0,0.15,0.33,0.35,5.0,10.00,5.00,5.00,44,NO

2,1,?,?,?,65.56,?,?,97.33,?,?,18.10,?,?,?,?,?,?,73,?,?,8.11,51.04,0.2,1,0,1,0.15,1,0.75,1,0.35,0.5,0,0.75,0.15,0,5.05,5.56,7.00,4.25,48,NO

0,0,?,?,?,?,?,?,?,?,?,0,?,?,?,?,?,?,?,?,?,?,0,0,0.6,0.5,0.85,0,0.5,0.15,0,0,1,0.85,0,1,0.85,1,5,7.78,5,4,?,NO

5,4,?,?,?,?,78.1,51.43,84.67,58.89,70.93,38.22,?,?,?,?,82.19,86.07,34.11,62.22,?,29.40,54.98,0.1,0,0.35,0.5,1,1,0.85,0,0.5,0.85,0,0.85,1, 0.5,6,4.44,6,4.75,49,SI

5,5,?,90,79.17,73.57,62.14,?,51.56,?,?,39.60,?,43.6,73.46,88.41,100,?,50.97,?,?,39.61,58.62,0.2,1,1,0,1,0.5,0.35,0,1,0.85,0.85,0.85,0.35,1,8,5.56,8,6.80,41,SI

4,3,87.38,69.17,?,69.76,59.05,?,?,?,?,31.71,?,99,71,?,27.96,78.24,?,?,?,?,22.88,53.86,0.6,1,0,1,1,1,1,0.15,0.5,1,1,1,1,8,1.11,8,7.75,74,SI

2,1,?,?,?,?,?,?,?,58.44,82.47,?,15.66,?,?,?,?,?,?,?,76.43,?,?,8.49,52.12,0.5,0.5,0,1,0.85,0.5,0,0,1,0.15,0,0.15,0,1,5,0,5,3.75,53,SI

1,1,?,?,?,79.37,?,?,?,?,?,?,8.82,?,?,?,70.71,?,?,?,?,?,7.86,45.95,0.2,1,0,1,0.85,0.85,1,0.15,1,0.85,0,1,1,0.5,5,1.11,5,3.75,43,NO

2,1,42.14,38.75,?,?,?,?,?,?,?,?,8.99,?,98.13,?,?,?,?,?,?,?,10.90,62.47,0.1,0,0,0.5,1,1,0.35,0,0.85,0.5,0.85,0.85,0,0.35,5.62,1.11,6.50,4.25,?,SI

1,0,77.38,?,?,?,?,?,?,?,?,?,8.60,?,?,?,?,?,?,?,?,?,?,0,0,0.8,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1.11,?,2.50,?,?

1,0,53.33,?,?,?,?,?,?,?,?,?,5.93,?,?,?,?,?,?,?,?,?,?,0,0,0.3,0,0,0,0,0,0,0,1,0.35,0.5,0.15,0.15,0.35,2.73,7.78,?,4.25,?,?

4,1,43.97,59.17,?,55.95,?,52.54,?,?,?,23.51,?,?,?,97.4,?,?,?,?,?,10.82,67.85,0,0,0,1,0.15,0.15,0,0,0.5,0.5,0,0.85,1,0.5,5,4.44,5,5,29,SI

2,1,87.86,?,42.5,?,?,?,?,?,?,14.48,100,?,?,?,?,?,?,?,?,11.11,65.79,0.2,0,1,0,0.35,1,0,1,1,0.5,0,0.15,1,1,5.33,3.33,7,6.75,48,NO

7,6,75.71,66.25,92.5,54.29,51.9,77.14,93.67,?,?,56.83,100,100,75.39,72.96,59.97,78.85,?,?,?,54.13,71.45,0.8,1,1,0,1,1,1,1,1,0.85,0.85,1,1,1,8,2.22,8,8.50,58,NO

7,7,0,85,88.75,74.92,?,39.29,83.33,55.06,?,47.37,66.67,81.25,80.58,87.72,?,100,75,98.21,?,65.49,69.47,0.4,1,1,1,1,1,0.5,1,1,1,1,0.35,1,1,5,4.44,5,6.5,53,NO

7,6,?,78.75,60,74.44,?,72.38,89.56,62.72,73.33,56.80,?,100,71.25,100,?,33.33,79.58,54.83,?,48.78,66.62,0.1,0,1,0,1,0.5,0.5,1,0.5,0.85,1,0.5,0.15,0.85,6.50,0,6.50,5,67,SI

3,3,68.41,50,60,?,?,?,?,?,?,19.82,70.1,100,100,?,?,?,?,?,?,30.01,61.95,0.6,0,0,0,1,1,0.85,1,1,1,0,0.85,1,1,8,4.44,8,6.50,71,SI

1,1,?,?,?,?,?,?,84,?,?,9.33,?,?,?,?,?,?,87.77,?,?,9.75,56.40,0.5,0.5,1,1,1,1,1,0,1,0.85,0,0.95,0.15,0.5,6.99,0,9,7,57,NO

0,0,?,?,?,?,?,?,?,?,?,0,?,?,?,?,?,?,?,?,?,?,0,0,0.4,1,1,1,0.85,1,0.85,1,0.85,0.85,0.85,0.35,1,1,8.11,1.11,8.50,7.75,43,SI

Lo expuesto anteriormente solo sería un fragmento de cómo se distribuye el fichero de datos para clasificación, como se puede observar sería igual que en regresión pero solo que le hemos añadido la variable de clasificación, ahora tendríamos que realizar lo mismo que antes es decir mostrar los estadísticos de todas las variables pero serían los mismo que los explicados anteriormente por lo tanto solo pondremos los de la variable nueva añadida como clase llamada “creer”.

- **Creer**

| **Valores perdidos** | **15%** |
|---|---|

| | **SI** | **NO** |
|---|---|---|
| **Count** | **30** | **30.0** |
| **Weight** | **48** | **48.0** |

*Tabla 43:Distribución de creer.*

# 3. Objetivos

En este punto vamos a explicar todo lo que intentamos conseguir que nuestro proyecto realice o que nos gustaría llegar obtener con él. Como ya se ha comentado anteriormente cuando a un docente se le presenta el tener que calificar a un grupo muy grande de alumnos se le crea un trabajo muy laborioso, este se puede intentar facilitar gracias a la realización de la autoevaluación, explicado en el punto anterior de este informe.

Entonces con nuestra investigación nos gustaría llegar al objetivo de poder obtener algún algoritmo o clasificador para hacer que la maquina sea la que pueda tomar una información, en nuestro caso por ejemplo una hoja de Excel, y pueda extraer resultados. Estos resultados serán obtenidos usando los datos de las notas que un alumno recibe durante el curso académico y esta serie de algoritmos. Con esto se conseguirá llegar a un objetivo final que será el poder ayudar al profesor a decidir si aceptar o no la nota final que ha propuesto el alumno o el poder adjudicar una nota y con esto también se podrá llegar a que al final el alumno no tenga que realizar ningún examen final para determinar su nota.

Tenemos como requisitos obligatorios el uso de la API de Weka [5].

Entonces podemos exponer que nuestra herramienta tendría los siguientes objetivos:

- **Recogida y preprocesado de la información:**

  Se podrá coger la información que se tiene en formato Excel, dicha información se recogerá y preprocesara adecuadamente para poder posteriormente hacer un análisis y minería de datos, dicha información se basa en un conjunto de notas obtenidas por los alumnos, distribuidas en dos fuentes de datos, esto ya ha sido explicado en apartado anteriores, así como también se ha hablado de los diferentes atributos que tenemos en nuestra base de dato.

- **Descubrir relaciones entre los atributos:**
  Queremos que nuestra investigación poder encontrar relaciones entre las diferentes variables descritas anteriormente en este punto, para ello vamos a utilizar pruebas estadísticas para determinar correlación (indicar la fuerza y la dirección de una relación lineal y proporcionalidad entre dos variables), por ejemplo, se usara estadísticos como “t de Student”, “test de Pearson”, entre otros.

- **Predecir notas y decisión:**

  Nuestra investigación vamos a intentar averiguar qué algoritmo sería el mejor para poder predecir cuál sería la nota final del alumno o si se acepta la nota que este propone, para ello vamos a utilizará algoritmos de predicción, clasificación y regresión. Todos estos algoritmos serán proporcionados por el programa *"Weka".* Algunos de estos algoritmos podrían ser *"J48", "MultilayerPerceptron", "NaiveBayes", "IB1"*, entre otros. Resultando de especial interés aquellos que proporcionan reglas de clasificación, ya que ayudaran a comprender el motivo por el que se realiza la predicción, facilitando un feedback [7] al profesor.

- **Informes finales:**

  Una vez se realice el análisis completo, con nuestra investigación vamos a poder mostrar un informe con los resultados que se ha obtenido, si los algoritmos son adecuados y también mostrar todo lo obtenidos, esta información se tendrá en formato PDF.

  Todo esto podría ser de ayuda para poder calificar al alumnado, siempre pensando en que se debería realizar un examen.

- **Creación de herramienta:**
  Una vez tengamos toda la investigación realizada tendríamos una serie de algoritmos como los mejores entonces realizaremos una aplicación en la cual se podría ejecutar los resultados de esta aplicación y la creación de PDF con los valores resultados que nos daría para una serie de datos.

# 4. Antecedentes

Como no hay alguna aplicación como tal que realiza lo mismo que la que se quiere realizar, entonces vamos a detallar en este apartado los softwares o técnicas utilizadas:

- **AUTOEVALUACIÓN:**

La autoevaluación es la evaluación que alguien hace de sí mismo o de algún aspecto o actividad propios.

Entonces la autoevaluación la vamos a ver como un proceso para permitir al alumnado conocer sus capacidades y limitaciones, con esto adoptar medidas para superarse en su proceso de aprendizaje. Esta técnica tiene muchas ventajas tanto para el alumnado como para el profesor.

Pero no todo son ventajas ya que, con la autoevaluación, podemos llegar a tener algunas dificultades como son la inflación de las calificaciones finales de los alumnados y la perdida de precisión de estas notas.

Actualmente los cuestionarios son la mejor forma de autoevaluación ya que tienes una serie de ventajas como son, el ser sencillos de corregir para la proporción de una respuesta rápida de los resultados lo cual es una ventaja para un número grande de alumnos y el fácil almacenamiento de los datos obtenidos para la creación de informes.

- **KDD**

Se puede definir **KDD (Knowledge Discovery in Database)** como un proceso no-trivial que permite la extracción de conocimiento, nuevo y significativo, oculto en una enorme cantidad de información almacenada en una o varias bases de datos.

KDD podemos decir que consta de una serie de etapas que se pueden enumerar de la manera que podemos ver en la ilustración 43.

1. **Selección de datos.** En esta etapa se determina las fuentes de información que se van a utilizar. Dichas fuentes de información pueden ser internas o externas a la organización.

2. **Preparación de los datos.** Por un lado, se eliminarán aquellos datos (atributos o tuplas) que resulten irrelevantes y también aquellos que sean inconsistentes o erróneos. Por otro lado, se definirán nuevos atributos mediante la agrupación o separación de otros ya existentes.

3. **Minería de datos.** En esta etapa se aplicarán sofisticados algoritmos a los datos preparados en el paso anterior con el fin de descubrir patrones ocultos, consistentes, comprensibles y potenciales útiles.

4. **Evaluación e interpretación.** Los expertos evaluarán e interpretarán los patrones obtenidos en el paso anterior para decidir lo que puede considerarse como conocimiento. Dependiendo de los resultados obtenidos, a veces se hace necesario regresar a uno de los pasos anteriores para una nueva iteración. La interpretación puede beneficiarse de procesos de visualización y sirve también para borrar patrones redundantes o irrelevantes.

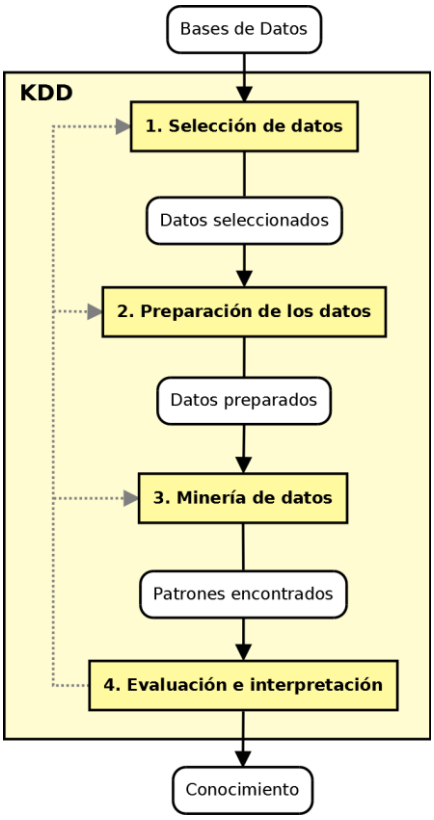



*Ilustración 43:Etapas de KDD*

Se puede deducir de los pasos anteriores que KDD es un proceso iterativo e interactivo. Exige un alto desempeño computacional y un alto grado de colaboración entre el experto y la computadora. De modo que la parte humana juega un rol clave. Por ese motivo, comúnmente se suele definir como un proceso semiautomático.
El conocimiento descubierto se incorporará al funcionamiento del sistema, o simplemente se documentará e informará a las partes interesadas.

- **MINERIA DE DATOS**

La Minería de Datos (Data Mining, DM) o exploración de datos es un campo de las ciencias de la computación referido al proceso que intenta descubrir patrones ocultos y útiles en grandes volúmenes de conjuntos de datos.

Los patrones de conocimiento descubiertos son evaluados e interpretados por expertos del dominio, en la siguiente etapa dentro del proceso KDD, para convertirse en conocimiento nuevo, que podrá explotarse en busca de algún beneficio para la organización. Se estima que la etapa de la Minería de Datos ocupa solo del 15% al 20% del esfuerzo total del proceso del KDD.

La Minería de Datos es la etapa de análisis dentro del proceso KDD. En ella, se puede utilizar métodos de la inteligencia artificial, aprendizaje automático, estadística y sistemas de bases de datos.

- **APRENDIZAJE AUTOMATICO**

El aprendizaje automático puede definirse como la disciplina de la inteligencia artificial, dedicada al diseño de algoritmos para identificar regularidades, patrones o reglas en un conjunto de datos disponibles y representar el conocimiento adquirido para modelar y explicar datos y para efectuar predicciones, diagnósticos, reconocimiento, controles, validaciones o simulaciones.

De forma más concreta, se trata de crear algoritmos que permitan generalizar comportamientos a partir de una información no estructurada suministrada en forma de nuestras. Por lo tanto, se trata de un proceso inducción del conocimiento.

En el corazón de los sistemas de aprendizaje automático se halla un algoritmo de inferencia, que indica al computador cómo llegar a conclusiones a partir de la colección de datos suministrados por el experto. El carácter generalista de estos algoritmos de inferencia es lo que confiere potencial a los sistemas de aprendizaje automático: El mismo algoritmo puede ser capaz de guiar un sistema que esté aprendiendo a reconocer objetos en imágenes digitales, a filtrar correo spam, a recomendar películas basándose en las alquiladas en ocasiones anteriores o a realizar cualquier otra cosa que se pueda pedir a un programa de inteligencia artificial.

Según el objetivo del análisis de los datos, los algoritmos utilizados se clasifican en Supervisados y No Supervisados.

  - Aprendizaje Supervisado: predice un dato desconocidos a priori, a partir de otros conocidos. Están dirigidas a la clasificación y a los sistemas de predicción.

- Aprendizaje No Supervisado: Se descubre patrones y tendencias en los datos.

Los únicos algoritmos relevantes para nuestra investigación son los de Aprendizaje Supervisado ya que el objetivo principal de este trabajo es el de llegar a predecir una nota final, por lo tanto, queremos llevar a cabo tareas de predicción.

- **MINERIA DE DATOS EDUCACIONALES**

Se puede denominar Minería de datos educacionales a un área emergente de investigación multidisciplinar que tiene como objetivo la aplicación de técnicas de minería de datos a información generada en entornos educativos para resolver determinados problemas de investigación educativa y entender el entorno en el que los alumnos aprenden.

Por lo tanto, se puede considerar EDM como una disciplina en la intersección de la minería de datos y la pedagogía. Por una parte, la pedagogía aporta el conocimiento intrínseco del proceso de aprendizaje y por otra parte, la minería de datos aporta las técnicas para descubrir patrones ocultos, consistentes, compresibles y potencialmente útiles.

El conocimiento que puede extraerse usando EDM es muy diverso ya que depende de a quién va dirigido (alumnos, profesores, autoridades académicas etcétera), del tipo de enseñanza (presencial, semipresencial, etcétera), del tipo de información, etcétera.

Como ya se ha dado a entender el campo de la minería de datos educacional ha seguido creciendo con el objetivo de traducir los datos en bruto en información significativa sobre el proceso de aprendizaje, por lo tanto podríamos decir que el EDM generalmente consta de cuatro fases.

La primera fase del proceso, sin contar el preprocesamiento de los datos, es el descubrimiento de las relaciones entre los datos. Esto implica la búsqueda a través de un repositorio de datos de un entorno educativo con el objetivo de encontrar relaciones consistentes entre variables. Se usan varios algoritmos para la identificación de este tipo de relaciones, incluyendo la clasificación estadística por correlación, como un análisis de regresión, el análisis de clasificación, el análisis de redes sociales, la aplicación de reglas de asociación y la extracción de secuencias. Las relaciones descubiertas deben ser entonces validadas con el fin de evitar errores, esta sería la segunda fase. Lo que se realiza es usar estas relaciones para hacer predicciones de eventos futuros en el ambiente de aprendizaje. Estas predicciones al final se usan para apoyar el proceso de la toma de decisiones.

Durante las fases 3 y 4, los datos se visualizan de manera diseminada para facilitar la emisión del juicio humano. Mucho se ha estudiado para llevar a cabo en las mejores visualizaciones de datos.

En este enfoque de la minería de datos podemos diferencias a cuatro usuarios o participantes que estarían implicados, estos son:

1. **Los estudiantes.** Nos interesa la comprensión de las necesidades de los alumnos y los métodos para mejorar la experiencia y el rendimiento del alumno. Por ejemplo, estos se podrían beneficiar de los conocimientos descubiertos para poder sugerir actividades y recursos que pueden utilizar en función de sus interacciones con las herramientas de aprendizaje en línea.

2. **Los educadores,** tratan de comprender el proceso de aprendizaje y los métodos que pueden utilizar para mejorar sus métodos de enseñanza. Pueden utilizar las aplicaciones del EDM para poder determinar cómo organizar y estructurar los planes de estudio, encontrar métodos para hacer llegar información del curso así como herramientas para poder motivar a sus estudiantes.

3. **Los investigadores,** estos se centran en el desarrollo y la evaluación de las técnicas de minería de datos para conseguir encontrar mejoras.

4. **Los administradores,** estos son los responsables de asignar los diferentes recursos para poder implementar las diferentes herramientas investigadas en las instituciones.

EDM tiene una inmensa cantidad de aplicaciones, entre las que podemos destacar las siguientes:

1. Permite la toma de decisiones basada en datos para mejorar la práctica educativa actual y el material de aprendizaje.
2. Mejora de la calidad y rentabilidad del sistema educativo.
3. Creación de modelos de predicción del rendimiento de los alumnos, obteniendo resultados prometedores que demuestra cómo determinadas características sociológicas, económicas y educativas de los alumnos pueden afectar en el rendimiento.

- **EVALUACIÓN POR PARES:**

La evaluación por pares [3] es el método usado para validar trabajos escritos con el fin de evaluar su calidad, originalidad, etc. Básicamente la idea es someter lo propuesto por un autor o estudiante al escrutinio de uno o más expertos en el tema. Esta evaluación suele responder con una evaluación del trabajo, que comúnmente incluye sugerencias acerca de cómo mejorarlo, la cual se le vuelve a enviar al autor del trabajo evaluado. Luego se procede a una revisión de las evaluaciones por el autor, todo cambio que ha sido propuesto no es vinculante, es decir obligatorios, solo son meramente orientativos para una mejora del trabajo.

La evaluación por pares puede ser dividido en 5 aspectos muy importantes estos son [4]:

1. Definición, esta es la etapa donde se define todo lo referente sobre los autores y revisores y el tema que se va a tratar.
2. Formas, sería la forma en la que se puede realizar la evaluación, existe la siguientes:
    - **Simple – Ciego:** El revisor conoce el autor, pero el autor no al revisor.
    - **Abierta:** Se conoce la identidad de autores y editores.
    - **Doble – Ciego:** Anónimos tanto autores como revisores.
3. Criterios de elección del revisor, especificación de los criterios que debe cumplir los revisores.
4. Base del procedimiento de revisión por pares.
5. Normas básicas de redacción, estándares de cómo se debe redactar el trabajo para que sea entendido por las dos partes.

- **WEKA:**

Weka (*Waikato Environment for Knowledge Analysis*) [5] es una plataforma de software para el Aprendizaje Automático y la Minaría de Datos escrito en Java y desarrollado por un grupo de investigadores del *Machine Learning Laboratory* en la Universidad de Waikato (Nueva Zelanda).


*Ilustración 44:Logo de Weka*

Las ventajas que tienes este software y por qué ha sido elegido como uno de los softwares que usaremos en nuestro proyecto son:

1. Es un *software* libre, distribuido bajo la licencia **GNU-GLP,** lo que significa que es de libre distribución y difusión.
2. Es un software implementado en **Java**, gracias a esto tiene la ventaja de ser multiplataforma pudiendo ejecutarse en cualquier máquina.
3. Contiene una **API** que se puede emplear en cualquier aplicación implementada en Java, permitiendo a dicha aplicación hacer uso de los diferentes algoritmos y funcionalidades que nos ofrece desde su interfaz gráfico o líneas de comandos.

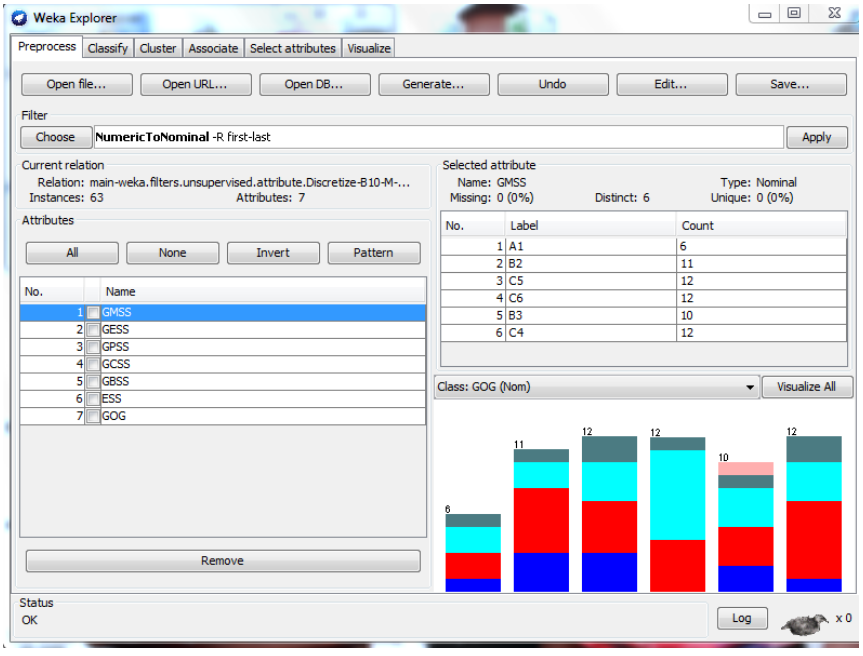



*Ilustración 45:Interfaz de Weka*

# 5. Restricciones

En primer lugar, hay que decir que se detallan los factores dato como aquellas restricciones que vienen impuestas por la propia naturaleza del problema y los factores estratégicos, se describen aquellas que han sido seleccionadas con el fin de obtener la solución óptima para resolver el problema.

## 5.1. FACTORES DATO

- La investigación debería cumplir todos los objetivos que no hemos propuesto.
- Deberíamos poder tratar la resolución de problemas en los diferentes enfoques que hemos tomado, correlación, regresión y clasificación.
- Se utilizará la herramienta “Weka” como apoyo de toda nuestra investigación.
- Sera obligatorio el uso de los datos proporcionados por el profesor Carlos García Martínez, profesor titular del Departamento de Informática y Análisis Numérico de la Escuela Politécnica Superior de la Universidad de Córdoba.

## 5.2. FACTORES ESTRATÉGICOS

- Para el análisis de los textos se ha utilizado la API “Azure” de Microsoft, la razón es porque está muy optimizado ya que tiene por detrás una gran compañía trabajando en ello y además nos da los datos de la forma que queremos para nuestra investigación y además hemos observado que tiene un porcentaje alto de acierto.
- Usaremos el lenguaje denominado “Java” para la creación de la aplicación y para ello utilizaremos el programa NetBeans.

# 6. Recursos

Una de las partes fundamentales a la hora de abordar un proyecto, es definir claramente los recursos de que se disponen y aquellos que serán necesarios para su total desarrollo. A continuación, se detallarán los medios tanto humanos como técnicos (recursos hardware y software) que serán utilizados para el desarrollo del proyecto que aquí se detalla.

- **Recursos humanos:**

  La dirección, coordinación y supervisión del proyecto será correspondido a:

  - Prof. Dr. Cristóbal Romero Morales, profesor titular del Departamento de Informática y Análisis Numérico de la Escuela Politécnica de la Universidad de Córdoba. Su labor consistirá en guiar los pasos a seguir por parte del autor del proyecto, revisar periódicamente el trabajo que se ira realizando y proporcionar los datos necesarios para la correcta realización del proyecto que nos ocupa.

  - Prof. Dr. Alberto Cano Rojas, profesor ayudante del Departamento de Informática de la Virginia Commonwealth University, Richmond, Virginia, EE. UU.. Su labor consistirá en guiar los pasos a seguir por parte del autor del proyecto, revisar periódicamente el trabajo que se ira realizando y proporcionar los datos necesarios para la correcta realización del proyecto que nos ocupa.

  El estudio del problema, análisis de requisitos, diseño, codificación, generar pruebas para encontrar fallos, documentación y desarrollo del proyecto será correspondido a D. Antonio Rafael Carmona Mengual, alumno del Grado en Ingeniería Informática de la Escuela Politécnica Superior de Córdoba.

- **Recursos hardware:**

Para la implementación de toda la investigación se ha precisado de los siguientes recursos hardware:

- **Ordenador personal:**
  - Marca: Hp 15-d002ss.
  - Procesador: Intel Core i3 2.4GHz.
  - 500 GB de memoria.
  - 8 GB de RAM.

- **Recursos software:**

    - Para el desarrollo de todo el software se hará uso de los siguientes recursos software bajo un S.O. de Windows 8.1.
    - **Weka** *"Waikato Environment for Knowledge Analysis"* [5]: Es un entorno para el análisis del conocimiento creado por la Universidad de Waikato.
    - **Azure:** Es una API de Microsoft la cual se ha utilizado para el análisis de sentimientos para el conjunto de textos escritos por los alumnos.
    - **Mozilla Firefox:** Navegador web utilizado a lo largo del proyecto para acceder a las distintas fuentes de información.
    - **Microsoft Excel 2016:** Para la realización de algunos de los cálculos necesarios mientras realizamos nuestra investigación.
    - **NetBeans:** Para la realización de toda la herramienta final.

Para el desarrollo de la documentación del proyecto se usarán las siguientes aplicaciones:

- **Microsoft Word 2016:** Editor de texto que se utilizará en la redacción de los manuales.
- **Microsoft PowerPoint 2016:** Programa para la realización de la presentación que se utilizará para la exposición del proyecto.

# 7. Experimentos y análisis

En este punto vamos a ir explicando los diferentes puntos que hemos ido tomado para el análisis de nuestros datos y los resultados que hemos obtenido de ello:

## 7.1. CORRELACIÓN:

En probabilidad y estadística entendemos como correlación [8] a un indicador el cual nos da la fuerza y dirección de una relación lineal y proporcionalidad entre dos variables estadísticas. Se considera que dos variables cuantitativas están correlacionadas cuando los valores de una de ellas varían sistemáticamente con respecto a los valores homónimos de la otra, entonces, si tenemos dos variables (A y B) existe correlación entre ellas si al disminuir los valores de A lo hacen también los de B y viceversa. La correlación entre dos variables no implica, por si misma, ninguna relación de causalidad.

Existen diversos coeficientes que miden el grado de correlación, adaptados a la naturaleza de los datos. El más conocido es el **coeficiente de correlación de Pearson**, el cual vamos a utilizar en todo el estudio de nuestro proyecto, pero también hay otros como:

- Coeficiente de correlación de Spearman.
- Correlación de Kendall.
- Correlación canónica.
- Coeficiente de correlación intraclase.
- Correlación Poliserial.
- Correlación de Kendall
- Correlación de Fecher.
- Etcétera.

Ahora se va a hablar un poco sobres el coeficiente de correlación de Pearson [9], que como se ha comentado anteriormente es el utilizado por nosotros. Como ya se ha comentado es una medida que nos da la relación entre dos variables y a diferencia de la covarianza, la correlación de Pearson es independiente de la escala de medida en la que se encuentre las variables. Entonces podemos definir este coeficiente con la siguiente expresión:

$$\rho_{X,Y} = \frac{\sigma_{X,Y}}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

Donde podemos decir que:

- $\sigma_{X,Y}$: Es la covarianza de las variables (X, Y).
- $\sigma_X$: Es la desviación típica de la variable X.
- $\sigma_Y$: Es la desviación típica de la variable X.

El valor de este coeficiente puede variar entre los valores [-1,1], por lo tanto, podemos decir que:

- Si nos da un valor de 1, entonces existe una correlación positiva perfecta, es decir nos indica una dependencia total entre las dos variables denominada relación directa, por lo tanto, cuando una de ellas aumenta, la otra también lo hace en la misma cantidad.
- Si nos da un valor que este en el rango (0,1), podemos decir que existe una correlación positiva.
- Si nos da un valor de 0, se puede afirmar que no existe una relación lineal entre las dos variables, pero esto no quiere decir que sean independientes ya que puede existir todavía relaciones no lineales.
- Si nos da un valor que este en el rango (-1,0), tenemos una correlación negativa.
- Si nos da un valor de -1 entonces tenemos una correlación negativa perfecta, por lo tanto, se puede decir que cuando una de las variables aumenta la otra disminuye en la misma proporción.

Entonces ahora vamos a hablar sobre cómo hemos planteado esto en nuestros datos y que resultados nos han salido. Lo primero que hemos realizado es decir cuáles son las variables más importantes para nuestro caso quedándonos con, ***"Notas de Examen", "Notas de heteroevaluación", "Notas de Autoevaluación", "Nota que se pone después de realizar el examen" y "Puntación de Azure".***

Por lo tanto, con estas nombradas lo que hemos realizado para empezar es ver que correlación ahí entre ellas y hemos obtenido los resultados que podemos ver en la siguiente tabla:

| VARIABLES RELACIONADAS | INDICE DE CORRELACIÓN |
|---|---|
| **Nota Examen – Nota Heteroevaluación** | 0.00831701 |
| **Nota Examen – Nota Autoevaluación** | 0.74291741 |
| **Nota Examen – Nota después de examen** | 0.72488594 |
| **Nota Heteroevaluación – Nota Autoevaluación** | -0.10092153 |
| **Nota Heteroevaluación – Nota después de examen** | 0.03157024 |
| **Nota Autoevaluación – Nota después de examen** | 0.59819886 |
| **Azure – Nota Examen** | 0.20310498 |

| | |
|---|---|
| **Azure – Nota Heteroevaluación** | -0.04668649 |
| **Azure – Nota Autoevaluación** | 0.20740816 |
| **Azure – Nota después de examen** | 0.25034037 |

*Tabla 44:Correlación variables importantes.*

Entonces se puede ver que las mejores variables corraladas serían, la nota del examen con la nota que los alumnos se pone en autoevaluación y la nota del examen con la nota que estos también se ponen después de realizar dicho examen. Entonces el que tengamos estar relaciones es muy bueno para la finalidad de este trabajo ya que podemos ver que las notas que se pone el alumnado antes de realizar el examen, sabiendo lo realizado durante el curso y la calificación que esperan obtener después de ver lo que han realizado en el examen están muy relacionadas con la nota que al final sacan en dicho examen. Por lo tanto, podemos seguir trabajando ya que si no tuviéramos nada de relación entre la nota que saca un alumno en el examen y la nota de autoevaluación entonces no tendría sentido seguir con la realización de este trabajo.

Esto último que se ha dicho está muy bien pero luego pensamos que estas variables para obtener la nota del examen la obtenemos ya muy avanzado el curso o ya casi cuando este está acabando, por lo tanto pensamos que tendríamos que buscar otra manera de obtener dicho valor y entonces pasamos a la siguiente fase y en esta lo que hicimos es buscar si tenemos una relación de estas variables antes nombradas con la anteriores que obtenemos durante el trascurso del curso, entonces realizamos los cálculos y obtenemos los resultados que podemos ver en la siguientes tablas:

- Nota Examen

| VARIABLES RELACIONADAS | INDICE DE CORRELACIÓN | VARIABLES RELACIONADAS | INDICE DE CORRELACIÓN |
|---|---|---|---|
| **Envió 2** | -0.02368043 | **Media de envíos** | 0,40096795 |
| **Envió 3** | 0.31203299 | **Media de evaluación** | 0,38635411 |
| **Envió 4** | 0.31203299 | **Media en talleres** | 0,39435015 |
| **Envió 5** | 0.07513077 | **Test** | 0,36667063 |
| **Envió 6** | 0.37096497 | **2a.Bayer** | 0,35914235 |
| **Envió 7** | 0.32687728 | **2b.Bayer** | 0,45391097 |
| **Envió 8** | 0.25704235 | **2c.Bayer** | 0,19824231 |
| **Envió 9** | 0.28375551 | **3a.Logica** | 0,50415622 |
| **Envió 10** | 0.33505564 | **3b.Logica** | 0,53544049 |
| **Evaluación 2** | -0.04911362 | **3c.Logica** | 0,48156501 |
| **Evaluación 3** | -0.10167418 | **Poda** | 0,41978688 |
| **Evaluación 4** | -0.02707551 | **Schank** | 0,59576817 |
| **Evaluación 5** | -0.46292087 | **Sowa** | 0,59876932 |
| **Evaluación 6** | 0.20859093 | **Complejidad** | 0,47554332 |
| **Evaluación 7** | -0,05477158 | **Marcos** | 0,60352983 |
| **Evaluación 8** | -0,12470852 | **A*** | 0,48347931 |

| | | | |
|---|---|---|---|
| **Evaluación 9** | 0,04465067 | **Bidireccional** | 0,60521321 |
| **Evaluación 10** | -0,3413579 | | |

*Tabla 45:Correlación de nota del examen con el resto.*

- Nota Heteroevaluación

| VARIABLES RELACIONADAS | INDICE DE CORRELACIÓN | VARIABLES RELACIONADAS | INDICE DE CORRELACIÓN |
|---|---|---|---|
| **Envió 2** | -0,30289672 | **Media de envíos** | -0,04625751 |
| **Envió 3** | 0,03338492 | **Media de evaluación** | -0,02960369 |
| **Envió 4** | -0,15379034 | **Media en talleres** | -0,0444465 |
| **Envió 5** | 0,05163964 | **Test** | -0,05030557 |
| **Envió 6** | -0,1848236 | **2a.Bayer** | 0,01765171 |
| **Envió 7** | -0,07568104 | **2b.Bayer** | 0,01119279 |
| **Envió 8** | -0,39260991 | **2c.Bayer** | -0,11026919 |
| **Envió 9** | -0,03522333 | **3a.Logica** | -0,02285985 |
| **Envió 10** | -0,39759114 | **3b.Logica** | 0,05363801 |
| **Evaluación 2** | -0,23544727 | **3c.Logica** | 0,06709643 |
| **Evaluación 3** | -0,33383207 | **Poda** | -0,19119449 |
| **Evaluación 4** | 0,15600359 | **Schank** | 0,14841889 |
| **Evaluación 5** | 0,20550578 | **Sowa** | 0,15784565 |
| **Evaluación 6** | -0,15232398 | **Complejidad** | -0,08761621 |
| **Evaluación 7** | -0,10280649 | **Marcos** | 0,03757566 |
| **Evaluación 8** | -0,07245147 | **A*** | 0,02210507 |
| **Evaluación 9** | 0,2865258 | **Bidireccional** | -0,0218625 |
| **Evaluación 10** | 0,36997085 | | |

*Tabla 46:Correlación de nota heteroevaluación con el resto*

- Nota Autoevaluación

| VARIABLES RELACIONADAS | INDICE DE CORRELACIÓN | VARIABLES RELACIONADAS | INDICE DE CORRELACIÓN |
|---|---|---|---|
| **Envió 2** | -0,08351821 | **Media de envíos** | 0,29611717 |
| **Envió 3** | 0,36537667 | **Media de evaluación** | 0,28807475 |
| **Envió 4** | 0,29716141 | **Media en talleres** | 0,14563511 |
| **Envió 5** | 0,23493209 | **Test** | 0,30625221 |
| **Envió 6** | 0,41910792 | **2a.Bayer** | 0,1623102 |
| **Envió 7** | 0,35721103 | **2b.Bayer** | 0,28320431 |
| **Envió 8** | 0,14849052 | **2c.Bayer** | 0,11772825 |
| **Envió 9** | 0,4064021 | **3a.Logica** | 0,20227696 |
| **Envió 10** | 0,31321676 | **3b.Logica** | 0,26703734 |
| **Evaluación 2** | 0,02795041 | **3c.Logica** | 0,35529106 |
| **Evaluación 3** | 0,01456772 | **Poda** | 0,27696307 |
| **Evaluación 4** | 0,09192905 | **Schank** | 0,20064827 |

| | | | |
|---|---|---|---|
| **Evaluación 5** | -0,47502894 | **Sowa** | 0,11102327 |
| **Evaluación 6** | 0,23261785 | **Complejidad** | 0,41241982 |
| **Evaluación 7** | -0,09220109 | **Marcos** | 0,28522326 |
| **Evaluación 8** | -0,1129066 | **A*** | 0,02826103 |
| **Evaluación 9** | 0,04693246 | **Bidireccional** | 0,31556356 |
| **Evaluación 10** | -0,53673919 | | |

*Tabla 47:Correlación de nota autoevaluación con el resto*

- Nota después del examen

| VARIABLES RELACIONADAS | INDICE DE CORRELACIÓN | VARIABLES RELACIONADAS | INDICE DE CORRELACIÓN |
|---|---|---|---|
| **Envió 2** | -0,03903902 | **Media de envíos** | 0,42709989 |
| **Envió 3** | 0,28133044 | **Media de evaluación** | 0,44851058 |
| **Envió 4** | 0,10316656 | **Media en talleres** | 0,29857535 |
| **Envió 5** | 0,17992524 | **Test** | 0,39679456 |
| **Envió 6** | 0,09356224 | **2a.Bayer** | 0,12533158 |
| **Envió 7** | 0,408163 | **2b.Bayer** | 0,4509408 |
| **Envió 8** | 0,04479147 | **2c.Bayer** | -0,07063308 |
| **Envió 9** | -0,04341136 | **3a.Logica** | 0,41105936 |
| **Envió 10** | 0,17197156 | **3b.Logica** | 0,48673772 |
| **Evaluación 2** | 0,02699676 | **3c.Logica** | 0,37678495 |
| **Evaluación 3** | -0,12048564 | **Poda** | 0,39301078 |
| **Evaluación 4** | 0,12983194 | **Schank** | 0,39894055 |
| **Evaluación 5** | -0,38751791 | **Sowa** | 0,31958072 |
| **Evaluación 6** | -0,08338749 | **Complejidad** | 0,46649591 |
| **Evaluación 7** | -0,15951086 | **Marcos** | 0,28151874 |
| **Evaluación 8** | 0,13010053 | **A*** | 0,38825191 |
| **Evaluación 9** | 0,09943974 | **Bidireccional** | 0,42329658 |
| **Evaluación 10** | -0,28472861 | | |

*Tabla 48:Correlación de nota después del examen con el resto*

- Azure

| VARIABLES RELACIONADAS | INDICE DE CORRELACIÓN | VARIABLES RELACIONADAS | INDICE DE CORRELACIÓN |
|---|---|---|---|
| **Envió 2** | -0,21444191 | **Media de envíos** | 0,19827408 |
| **Envió 3** | 0,00862855 | **Media de evaluación** | 0,2360795 |
| **Envió 4** | 0,01705887 | **Media en talleres** | 0,12403362 |
| **Envió 5** | 0,34177739 | **Test** | 0,24749998 |
| **Envió 6** | 0,20587222 | **2a.Bayer** | -0,20816884 |
| **Envió 7** | 0,43998812 | **2b.Bayer** | -0,00110972 |
| **Envió 8** | 0,02684165 | **2c.Bayer** | -0,21750961 |
| **Envió 9** | -0,1476109 | **3a.Logica** | 0,18173765 |

| | | | |
|---|---|---|---|
| **Envió 10** | 0,16637078 | **3b.Logica** | 0,10119488 |
| **Evaluación 2** | -0,19350031 | **3c.Logica** | 0,08033638 |
| **Evaluación 3** | 0,15539767 | **Poda** | 0,20510774 |
| **Evaluación 4** | 0,18958734 | **Schank** | -0,03291611 |
| **Evaluación 5** | -0,2967306 | **Sowa** | 0,10991854 |
| **Evaluación 6** | -0,18791171 | **Complejidad** | -0,03948755 |
| **Evaluación 7** | -0,01952156 | **Marcos** | -0,08313521 |
| **Evaluación 8** | -0,12361699 | **A*** | 0,02398155 |
| **Evaluación 9** | 0,22843238 | **Bidireccional** | 0,04802455 |
| **Evaluación 10** | 0,10776077 | | |

*Tabla 49:Correlación de Azure con el resto*

Entonces en estas tablas podemos llegar a ver que variables están más correladas con respecto a las principales que hemos elegido, entonces podemos decir que por ejemplo para la "Nota de examen" tenemos como mejor correladas variables como, ***"Bidireccional", "Marcos", "Sowa", "Schank",*** **entre otras.** Para la "Nota de Heteroevaluación" tendríamos, ***"Envió 10", "Envió 8", "Envió 2", entre otros.*** Para la variable "Nota de Autoevaluación" tenemos las variables, ***"Evaluación 10", "Evaluación 5", "Envió 9", "Envió 6".*** Para la nota después del examen cogeríamos en orden de más correlado a menos algunas variables como, ***"3.b.Logica", "Complejidad", "Bidireccional", "2.b.Bayes", entre otras*** y para acabar para la variable "Azure" podemos elegir, por ejemplo, ***"Envió 7", "Envió 5",*** **entre otros.**

Algo curioso que veremos en puntos posteriores es que en los algoritmos para la obtención de una nota aproximada del alumno ahí una alta probabilidad que de estos elijan estas variables para la obtención de esta.

Entonces ahora para "rizar más el rizo" en la última fase de la parte de correlación lo que hicimos es ver qué valor de correlación obtendríamos de todas las variables con todas las demás en vez de solo ver con las variables principales elegidas. Entonces podemos obtener una serie de valores.

Como sería una tabla de un tamaño demasiado grande, como de unas 300 filas aproximadamente para no llenar este trabajo con una cantidad de hojas en las que estuviera solo esta tabla se ha puesto el siguiente enlace en el cual se puede ver todos los valores resultantes de dicha experimentación.

https://ucordoba-my.sharepoint.com/:x:/g/personal/i42camea_uco_es/Eb-vlKqLgvhHt6Tzh_BqNrMB8NzRuauqyvfO1RlNidHjLQ?e=yEisxg

Con esto terminaríamos la primera fase de investigación de este proyecto.

## 7.2. REGRESIÓN:

La siguiente fase que vamos a tomar para nuestra investigación va a ser aplicar una serie de algoritmos para ver como de buenos sería aplicar regresión en nuestro conjunto de datos, lo que vamos a intentar es conseguir algún buen modelo para poder obtener una aproximación de la nota que el alumno sacaría en el examen teniendo en cuenta el resto de nuestras variables.

Lo primero de lo que vamos a hablar es sobre la regresión [10] en general, en estadística podemos entender la regresión como un proceso para estimar las relaciones entre variables. Entonces podemos decir que tenemos una variable dependiente y una o más variables independientes por lo tanto este método lo que intenta es entender como el valor de la variable dependiente varía al cambiar el valor de una o más de las variables independiente.

El análisis de regresión es ampliamente utilizado para la predicción y previsión, donde su uso tiene superposición sustancial en el campo del aprendizaje automático.

Entonces podemos clasificar los tipos de regresión [11] según numerosos criterios:

1. En función del número de variables independiente:
    - **Regresión simple:** Cuando la variable Y depende únicamente de una sola variable X.
    - **Regresión múltiple:** Cuando la variable Y depende de varias variables X.

2. En función del tipo de función f(x):
    - **Regresión lineal:** Cuando f(x) es una función lineal.
    - **Regresión no lineal:** Cuando f(x) no es una función lineal.

3. En función de la naturaleza de la relación que existe entre las variables:
    - La variable X puede ser causa del valor de la variable Y.
    - Cuando las dos variables tienen simplemente una relación entre ellas.

Entonces lo primero que se ha realizado es lo siguiente, para tener más variedad y ver cómo afecta la nota del examen según qué variables tenemos hemos dividido nuestro Dataset en varios diferentes, hemos sacado 5 Dataset los cuales tiene diferente número de variables estos son los siguientes:

- **TodosAtributos:** En este hemos dejado todas las variables que teníamos desde el inicio.

- **SoloProfe:** En esta hemos dejado las variables las cuales son puestas por el profesor a lo largo del curso, como por ejemplo las notas de cada parte del examen o las notas de los diferentes talleres.

- **SoloAlumnos:** En este Dataset hemos dejado las variables las cuales son de los alumnos como son la Autoevaluación, nota después de hacer el examen etc.

- **AtributosSelecionados:** En este nos hemos quedado con los atributos que nos ha dado "Weka" como los mejores. Estas variables se obtienen de la sección que tiene dicho programa que se llama "Select atributes", las variables que hemos obtenido son "media de taller", "autoevaluación", "marcos", entre otras.

- **MejoresCorrelados:** En este Dataset nos hemos quedado con los atributos que mejor están correlados con la nota del examen, esto lo hemos obtenido de la parte anterior que ya hemos explicado.

Entonces para aplicar regresión en nuestros conjuntos de datos lo que hemos utilizado como ya hemos mencionada a lo largo de este trabajo es una herramienta llamada "Weka", entonces lo que hemos realizado es aplicar todos los algoritmos posibles para regresión estos han sido los siguiente: *"GaussianProcesses", "LinealRegression", "MultilayerPerceptron", "RBFNetwork", "RBFRegressor", "SimpleLinearRegression", "SMOreg", "IBK", "Kstar", "LWL", "AdditiveRegression", "Bagging", CVParameterSelection", "RandomCommittee", "RandomSubSpace", "RegressionByDiscretization", "Stacking", "InputMappedClassifier", "DecisionTable", "M5Rules", "ZeroR", "DecisionStump", "M5P", "RandomForest", "RandomTree" y "REPTree".*

Cada uno de estos algoritmos nos proporciona una serie de resultados y nosotros nos hemos quedado con las siguientes variables para ver que algoritmo es mejor, estas variables son:

- **Correlation coefficient:** Se usa en estadística para medir qué tan fuerte es una relación entre dos variables. Existen varios tipos, pero el más importante es el coeficiente de correlación.

- **Mean Absolute error** [12]**:** Mide la magnitud promedio de los errores en un conjunto de predicciones, sin considerar su dirección. Es el promedio sobre la muestra de prueba de las diferencias absolutas entre la predicción y la observación real donde todas las diferencias individuales tienen el mismo peso.

- **Root mean squared error** [13]**:** Es la desviación estándar de los residuos (errores de predicción). Los residuos son una medida de cuanto lejos están los puntos de los datos de la línea de regresión, es decir es una medida de dispersión de estos residuos.

- **Relative Absolute error** [14]**:** Es muy similar al relative squared error en el sentido de que también es relativo a un predictor simple, que es solo el promedio de los valores reales. En este caso, sin embargo, el error es solo el error absoluto total en lugar del error cuadrado total. Por lo tanto, el relative Absolute error toma el error absoluto total y lo normaliza dividiendo por el error absoluto total del predictor simple.

- **Root relative squared error** [15]**:** Es relativo a lo que hubiera sido si se hubiera utilizado un predictor simple. Más específicamente, este predictor simple es solo el promedio de los valores reales. Por lo tanto, el root relative squared error toma el error cuadrado total y lo normaliza dividiendo por el error cuadrado total del predictor simple. Al tomar la raíz se reduce el error a las mismas dimensiones que la cantidad que se predice.

Entonces hemos realizado todas las pruebas y hemos obtenido los resultados que podemos ver en las siguientes tablas, una por cada Dataset que hemos dicho anteriormente que tenemos:

- **Todos los atributos:**

| | **Correlation coefficient** | **Mean Absolute error** | **Root mean squared error** | **Relative Absolute error** | **Root relative squared error** |
|---|---|---|---|---|---|
| **GaussianProcesses** | 0,8217 | 0,9122 | 1,1685 | 68,07% | 56,90% |
| **LinealRegression** | 0,7975 | 0,9717 | 1,2882 | 72,51% | 62,81% |
| **MultilayerPerceptron** | 0,8226 | 0,8700 | 1,1900 | 65,37% | 58,49% |
| **RBFNetwork** | 0,3313 | 1,5301 | 1,9351 | 100,19% | 94,35% |
| **RBFRegressor** | 0,7880 | 0,9120 | 1,2800 | 68,11% | 62,41% |
| **SimpleLinearRegression** | 0,6406 | 1,1317 | 1,5904 | 84,45% | 77,54% |
| **SMOreg** | 0,7765 | 1,0280 | 1,3516 | 76,77% | 65,91% |
| **IBK** | 0,6756 | 1,4432 | 2,0600 | 100,70% | 100,50% |
| **Kstar** | 0,3526 | 4,5342 | 5,0246 | 338,00% | 245,00% |
| **LWL** | 0,7013 | 0,9848 | 1,4502 | 73,40% | 70,71% |
| **AdditiveRegression** | 0,7335 | 1,0452 | 1,4616 | 78,00% | 71,27% |
| **Bagging** | 0,8503 | 0,7554 | 1,1430 | 56,37% | 55,75% |
| **CVParameterSelection** | -0,2779 | 1,3400 | 2,0500 | 100,00% | 100,00% |

| | | | | | |
|---|---|---|---|---|---|
| **RandomCommittee** | 0,8395 | 0,7695 | 1,1050 | 57,42% | 53,88% |
| **RandomSubSpace** | 0,7762 | 0,9137 | 1,3316 | 68,19% | 64,93% |
| **RegressionByDiscretization** | 0,7340 | 0,9560 | 1,3980 | 71,40% | 68,18% |
| **Stacking** | -0,2779 | 1,3400 | 2,0500 | 100,00% | 100,00% |
| **InputMappedClassifier** | -0,2779 | 1,3400 | 2,0500 | 100,00% | 100,00% |
| **DecisionTable** | 0,7021 | 0,9770 | 1,4560 | 72,91% | 70,99% |
| **M5Rules** | 0,8279 | 0,8460 | 1,1700 | 63,13% | 57,17% |
| **ZeroR** | -0,2779 | 1,3400 | 2,0500 | 100,00% | 100,00% |
| **DecisionStump** | 0,6145 | 1,0490 | 1,6300 | 78,29% | 79,51% |
| **M5P** | 0,8160 | 0,8844 | 1,1867 | 66,00% | 57,86% |
| **RandomForest** | 0,8549 | 0,7653 | 1,0879 | 57,11% | 53,05% |
| **RandomTree** | 0,7161 | 1,0311 | 1,6362 | 76,95% | 79,78% |
| **REPTree** | 0,6687 | 1,0218 | 1,5418 | 76,25% | 75,18% |

*Tabla 50:Regresion para TodosAtributos.*

Para poder ver que algoritmo es el mejor en este Dataset y poder tomar una decisión se ha realizado primero un diagrama box plot para decidir que variable sería la más representativa en este Dataset. El resultado que hemos podido obtener es el siguiente:

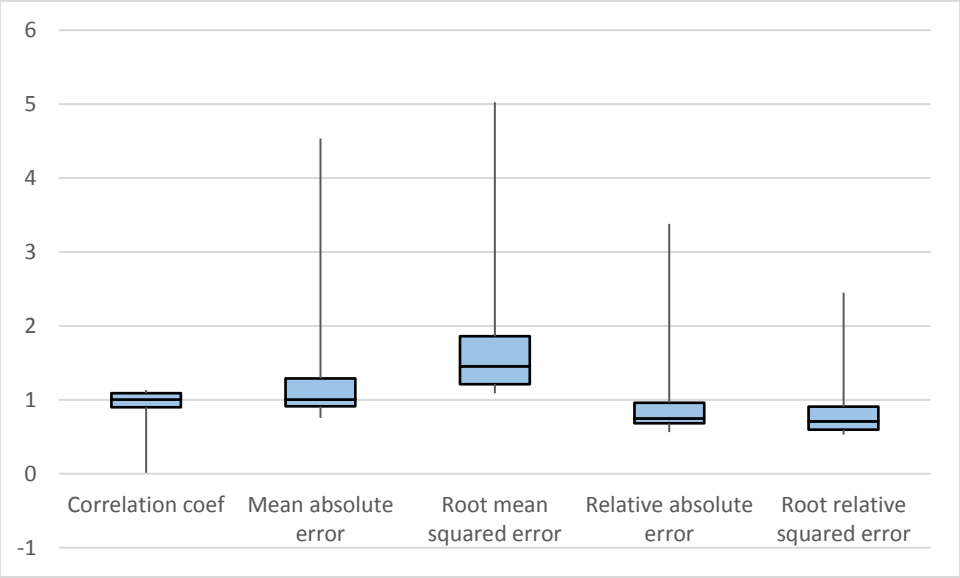



*Ilustración 46:Diagrama para TodosAtributos, regresión.*

Entonces en este diagrama se puede observar que las variables *"Mean Absolute error"* y *"Root mean squared error",* tiene máximos que se pueden considerar ruido, es decir no son comunes a la media esto lo podemos observar en que uno de los "whisker" ("bigotes", en español) se va muy lejos de donde está la caja o rango intercuartil y por lo tanto de los valores medios. Entonces las variables con las que nos vamos a quedar para ver cuál es el mejor algoritmo en este caso podría ser *"Correlation coefficient" o "Root relative squared error",* esto es porque podemos observar que estas dos variables son las que menos dispersión y más equilibradas están con respecto a las demás.

Entonces habiendo decidido que variables vamos a utilizar y con los datos que tenemos en la tabla 7 podemos decidir que para este caso nos quedaríamos con los algoritmos siguientes, *"RandomCommittee"* con un error del 53.8% y un índice de correlación del 0.8395 o también podríamos elegir el algoritmo *"RandomForest"* con un error del 53% y un índice de correlación del 0.8549.

Como estos dos van a ser los algoritmos escogidos en esta situación vamos ahora a comentar un poco sobre ellos:

- **RandomCommittee:** Este algoritmo lo que realiza es la construcción de un conjunto de clasificadores o arboles de forma aleatoria. Cada clasificador de base es construido usando una semilla diferente (pero basado en los mismos datos). La predicción que te da al final es un promedio directo de las predicciones generadas por los clasificadores básicos individuales.

- **RandomForest:** Es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector formado de forma aleatoria, probado independientemente y con la misma distribución para cada uno de estos. Este algoritmo es una modificación sustancial de "bagging" que construye una larga colección de árboles no correlacionados y luego los promedia.

- **Solo Profe:**

| | **Correlation coefficient** | **Mean Absolute error** | **Root mean squared error** | **Relative Absolute error** | **Root relative squared error** |
|---|---|---|---|---|---|
| **GaussianProcesses** | 0,7103 | 1,0598 | 1,4835 | 79,09% | 72,34% |
| **LinealRegression** | 0,5668 | 1,2030 | 2,0040 | 89,78% | 97,54% |
| **MultilayerPerceptron** | 0,5585 | 1,4577 | 2,2099 | 108,00% | 107,00% |
| **RBFNetwork** | 0,3107 | 1,4883 | 1,6432 | 111,00% | 94,75% |
| **RBFRegressor** | 0,6155 | 1,1901 | 1,7196 | 88,82% | 83,85% |
| **SimpleLinearRegression** | 0,3844 | 1,4791 | 1,9124 | 110,00% | 93,25% |
| **SMOreg** | 0,5809 | 1,2894 | 2,0151 | 96,23% | 98,26% |
| **IBK** | 0,6324 | 1,6995 | 2,3223 | 126,00% | 113,00% |
| **Kstar** | 0,3093 | 4,5321 | 5,0270 | 338,00% | 245,00% |
| **LWL** | 0,5440 | 1,3303 | 1,7516 | 99,28% | 85,41% |
| **AdditiveRegression** | 0,5319 | 1,4124 | 1,9472 | 105,00% | 94,94% |
| **Bagging** | 0,7513 | 0,9991 | 1,4026 | 74,56% | 68,39% |
| **CVParameterSelection** | -0,2779 | 1,3400 | 2,0508 | 100,00% | 100,00% |
| **RandomCommittee** | 0,7206 | 0,9780 | 1,4090 | 72,99% | 68,70% |

| | | | | | |
|---|---|---|---|---|---|
| **RandomSubSpace** | 0,7356 | 1,0431 | 1,4450 | 77,85% | 70,44% |
| **RegressionByDiscretization** | 0,6233 | 1,2356 | 1,7119 | 92,21% | 83,47% |
| **Stacking** | -0,2779 | 1,3400 | 2,0508 | 100,00% | 100,00% |
| **InputMappedClassifier** | -0,2779 | 1,3400 | 2,0508 | 100,00% | 100,00% |
| **DecisionTable** | 0,6230 | 1,1005 | 1,6216 | 82,13% | 79,07% |
| **M5Rules** | 0,0038 | 1,6732 | 6,2105 | 124,00% | 302,00% |
| **ZeroR** | -0,2779 | 1,3400 | 2,0508 | 100,00% | 100,00% |
| **DecisionStump** | 0,5429 | 1,2786 | 1,7148 | 95,42% | 83,62% |
| **M5P** | 0,6364 | 1,1055 | 1,6649 | 82,50% | 81,18% |
| **RandomForest** | 0,7866 | 0,9157 | 1,2710 | 68,34% | 52,05% |
| **RandomTree** | 0,5869 | 1,2719 | 1,8466 | 94,92% | 90,04% |
| **REPTree** | 0,5117 | 1,3503 | 1,7968 | 100,00% | 87,61% |

*Tabla 51:Regresión para soloProfe.*

Para poder ver que algoritmo es el mejor en este Dataset y poder tomar una decisión se ha realizado un diagrama box plot como ya se ha comentado anteriormente con esto se va a decidir que variable sería la más representativa en este conjunto de datos. El resultado que hemos podido obtener es el siguiente:

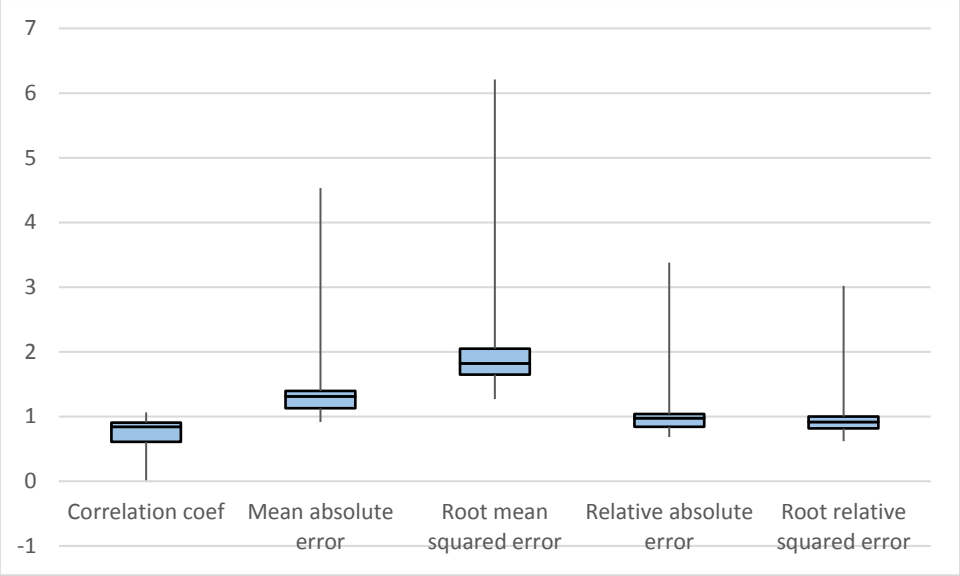



*Ilustración 47:Diagrama para soloprofe en regresión.*

Podemos ver qué pasa algo parecido al caso anterior y esto es que varias variables como por ejemplo *"Root mean squared error"* tiene valores máximos o mínimos que se pueden considerar anormales. Entonces las variables que menos dispersión tiene son *"Relative Absolute error" y "Root relative squared error"* y de estas dos nos vamos a quedar con la segunda y la razón de esta elección es porque la segunda variable es más simétrica, es decir el 1º y 3º cuartil están a una distancia de la media muy parecida cosa que no pasa con la primera variable.

Entonces como ya se ha elegido la variable que vamos a tener en cuenta vamos a ver que algoritmo es el mejor para este Dataset, lo mejor con lo que nos podríamos quedar son *"RandomForest"* con un error del 52% y *"Bagging"* con un error del 68%. *"RandomForest"* ya ha sido explicado por lo tanto no se va a volver a hacer, pero *"Bagging"* no por eso se va a proceder a comentar brevemente algo sobre él.

- **Bagging**: Es un meta-algoritmo del conjunto de aprendizaje automático diseñado para mejorar la estabilidad y la precisión de los algoritmos de aprendizaje automático utilizados en la clasificación y regresión estadística. También reduce la varianza y ayuda a evitar el sobreajuste. Aunque por lo general se aplica a los métodos de árboles de decisión, también se puede usar con cualquier tipo de método. El embolsado es un caso especial del enfoque de promediado modelo.

- **Solo Alumno:**

| | **Correlation coefficient** | **Mean Absolute error** | **Root mean squared error** | **Relative Absolute error** | **Root relative squared error** |
|---|---|---|---|---|---|
| **GaussianProcesses** | 0,4627 | 1,2609 | 1,8248 | 94,10% | 88,98% |
| **LinealRegression** | 0,6064 | 1,1851 | 1,6184 | 88,45% | 78,91% |
| **MultilayerPerceptron** | 0,5354 | 1,3136 | 1,7464 | 98,03% | 85,15% |
| **RBFNetwork** | 0,0210 | 1,3662 | 2,0534 | 101,00% | 100,00% |
| **RBFRegressor** | 0,5795 | 1,1520 | 1,6750 | 85,97% | 81,67% |
| **SimpleLinearRegression** | 0,6406 | 1,1317 | 1,5904 | 84,46% | 77,55% |
| **SMOreg** | 0,5775 | 1,0744 | 1,7779 | 80,18% | 86,69% |
| **IBK** | 0,6288 | 0,9957 | 1,6416 | 74,30% | 80,04% |
| **Kstar** | 0,5277 | 1,2448 | 1,7362 | 92,89% | 84,66% |
| **LWL** | 0,4840 | 1,1524 | 1,8337 | 86,00% | 89,41% |
| **AdditiveRegression** | 0,7245 | 1,0028 | 1,4378 | 74,84% | 70,11% |
| **Bagging** | 0,5549 | 1,1544 | 1,6949 | 86,15% | 82,64% |
| **CVParameterSelection** | -0,2779 | 1,3400 | 2,0508 | 100,00% | 100,00% |
| **RandomCommittee** | 0,4564 | 1,4330 | 1,9813 | 106,00% | 96,61% |
| **RandomSubSpace** | 0,4827 | 1,2114 | 1,7966 | 90,41% | 87,60% |
| **RegressionByDiscretization** | 0,5507 | 1,1911 | 1,6968 | 88,89% | 82,74% |
| **Stacking** | -0,2779 | 1,3400 | 2,0508 | 100,00% | 100,00% |
| **InputMappedClassifier** | -0,2779 | 1,3400 | 2,0508 | 100,00% | 100,00% |
| **DecisionTable** | 0,6774 | 1,0608 | 1,5208 | 79,17% | 74,16% |
| **M5Rules** | 0,7407 | 0,9590 | 1,4148 | 71,57% | 68,99% |
| **ZeroR** | -0,2779 | 1,3400 | 2,0508 | 100,00% | 100,00% |
| **DecisionStump** | 0,6145 | 1,0491 | 1,6307 | 78,29% | 79,51% |
| **M5P** | 0,7117 | 0,9942 | 1,4282 | 74,20% | 69,64% |

| | | | | | |
|---|---|---|---|---|---|
| **RandomForest** | 0,4668 | 1,4454 | 1,8876 | 107,00% | 92,04% |
| **RandomTree** | 0,3249 | 1,4528 | 2,3139 | 108,00% | 112,00% |
| **REPTree** | 0,5299 | 1,1770 | 1,7336 | 87,84% | 84,53% |

*Tabla 52:Regresión para SoloAlumno.*

Como ya se comentó anteriormente para poder hacer una elección sobre que algoritmo elegir se va a crear un diagrama box plot, para este caso la solución es el que podemos ver a continuación:

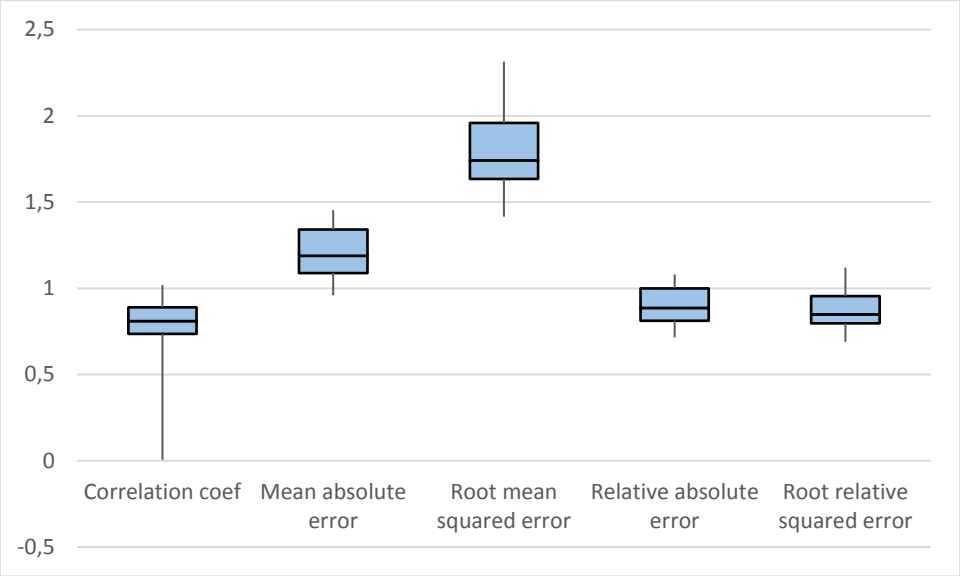



*Ilustración 48: Diagrama para SoloAlumnos en regresión.*

Como se puede observar vemos que las variables con menos dispersión serían *"Correlation coefficient", "Relative Absolute error" y "Root relative squared error",* pero la última dicha tiene una asimetría por lo tanto esta va a ser descartada en la elección. Entonces, de las dos que nos queda va a ser elegida *"Relative Absolute error"* y esto es porque esta no tiene algún máximo o mínimo que se aleje de lo normal cosa que si le pasa a la variable *"Correlation coefficient".*

Entonces con esto se va a tomar la decisión de elegir que algoritmo sería el mejor para el caso de este conjunto de datos por lo tanto hemos elegido los algoritmos, M5Rules con una variable de valor 71.5% y también el algoritmo M5P con un valor de 74.2%. Estos dos algoritmos se podrían definir de la siguiente forma:

- **M5Rules:** Genera una lista de decisiones para problemas de regresión usando la metodología de "Separa y Conquista". En cada iteración construye un árbol usando el algoritmo M5 y convierte la "mejor" hoja de este árbol en una regla solución del algoritmo.

- **M5P:** Implementa una serie de rutinas para la generación de árboles y reglas de forma más básica del algoritmo M5.

- **Atributo Seleccionados:**

| | Correlation coefficient | Mean Absolute error | Root mean squared error | Relative Absolute error | Root relative squared error |
|---|---|---|---|---|---|
| **GaussianProcesses** | 0,6346 | 1,1487 | 1,5911 | 85,72% | 77,58% |
| **LinealRegression** | 0,8582 | 0,8293 | 1,0524 | 61,89% | 51,32% |
| **MultilayerPerceptron** | 0,8074 | 0,9396 | 1,3146 | 70,12% | 64,10% |
| **RBFNetwork** | 0,5756 | 1,3528 | 1,6635 | 100,00% | 81,11% |
| **RBFRegressor** | 0,9039 | 0,7022 | 0,8722 | 52,41% | 42,53% |
| **SimpleLinearRegression** | 0,6406 | 1,1317 | 1,5904 | 84,46% | 77,55% |
| **SMOreg** | 0,8761 | 0,7836 | 0,9806 | 58,48% | 47,81% |
| **IBK** | 0,7735 | 0,9263 | 1,3424 | 69,13% | 65,46% |
| **Kstar** | 0,8617 | 0,6966 | 1,0352 | 51,99% | 50,48% |
| **LWL** | 0,7534 | 0,9073 | 1,3394 | 67,71% | 65,31% |
| **AdditiveRegression** | 0,7783 | 0,8954 | 1,2839 | 66,82% | 62,60% |
| **Bagging** | 0,8500 | 0,7348 | 1,1423 | 54,84% | 55,70% |
| **CVParameterSelection** | -0,2779 | 1,3400 | 2,0508 | 100,00% | 100,00% |
| **RandomCommittee** | 0,8886 | 0,6152 | 0,9381 | 45,91% | 45,74% |
| **RandomSubSpace** | 0,8306 | 0,8288 | 1,2069 | 61,85% | 58,85% |
| **RegressionByDiscretization** | 0,7953 | 0,8427 | 1,2357 | 62,89% | 60,25% |
| **Stacking** | -0,2779 | 1,3400 | 2,0508 | 100,00% | 100,00% |
| **InputMappedClassifier** | -0,2779 | 1,3400 | 2,0508 | 100,00% | 100,00% |
| **DecisionTable** | 0,6697 | 1,0288 | 1,5151 | 76,78% | 73,88% |
| **M5Rules** | 0,8259 | 0,7801 | 1,1561 | 58,22% | 56,37% |
| **ZeroR** | -0,2779 | 1,3400 | 2,0508 | 100,00% | 100,00% |
| **DecisionStump** | 0,6145 | 1,0491 | 1,6307 | 78,29% | 79,51% |
| **M5P** | 0,8722 | 0,7472 | 0,9964 | 55,76% | 48,58% |
| **RandomForest** | 0,8873 | 0,6447 | 0,9566 | 48,11% | 46,64% |
| **RandomTree** | 0,7460 | 0,8987 | 1,4294 | 67,07% | 69,70% |
| **REPTree** | 0,6890 | 0,9699 | 1,5085 | 72,39% | 73,55% |

*Tabla 53:Regresión para Atributos Seleccionados.*

Para poder ver que algoritmo es el mejor en este Dataset y poder tomar una decisión se ha realizado primero un diagrama box plot para decidir que variable sería la más representativa en este Dataset. El resultado que hemos podido obtener es el siguiente:

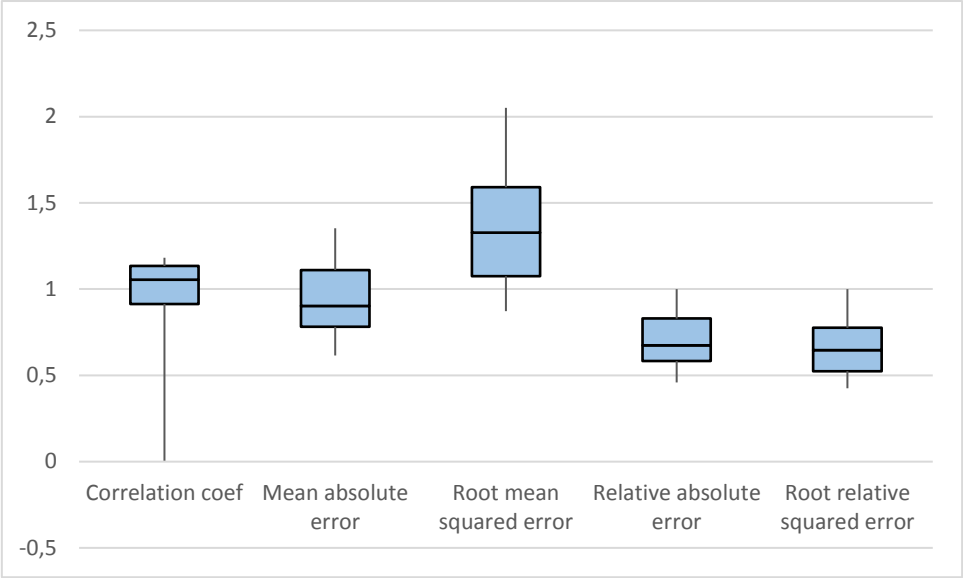



*Ilustración 49:Diagrama para Atributos Seleccionados en regresión.*

Se puede descartar directamente como variables significativas para nosotros de este Dataset, la variable *"Correlation coefficient"* por ese mínimo anormal que se puede ver ya que uno de sus "whisker" ("bigotes", en español) se alega demasiado de los valores medios y otra variable a descartar podría ser *"Root mean squared error"* ya que podemos ver un gran tamaño de la caja o rango intercuartil lo que nos daría la información de una gran dispersión de los datos.

Entonces de las otras tres que nos queda podemos descartar dos que son *"Mean Absolute error" y "Relative Absolute error"* y la razón sería por tener un desequilibrio, es decir que el 1º cuartil está mucho más alegado de la media que el 3º cuartil, por lo tanto, para este Dataset nos quedaremos con la variable *"Root relative squared error".*

Con esto decidido podemos ahora decidir con que algoritmo o algoritmos nos podríamos quedar para obtener los resultados, estos serían, "RBFRegressor" con un valor de la variable elegida del 42,5% y "RandomCommittee" con un valor de 45,7%. Como "RandomCommittee" ya ha sido comentado antes solo se va a decir de que trata el algoritmo "RBFRegressor".

- **RBFRegressor:** Algoritmo que implementa redes de función de base radial para la regresión, entrenadas de manera totalmente supervisada utilizando la clase de optimización de Weka minimizando el error al cuadrado con el método BFGS.

- **Mejores Correlados:**

| | Correlation coefficient | Mean Absolute error | Root mean squared error | Relative Absolute error | Root relative squared error |
|---|---|---|---|---|---|
| **GaussianProcesses** | 0,6946 | 1,1353 | 1,4850 | 84,73% | 72,41% |
| **LinealRegression** | 0,8647 | 0,7911 | 1,0251 | 59,04% | 49,98% |
| **MultilayerPerceptron** | 0,8239 | 0,9323 | 1,2885 | 69,57% | 62,83% |
| **RBFNetwork** | 0,5665 | 1,3796 | 1,6765 | 102,00% | 81,75% |
| **RBFRegressor** | 0,8849 | 0,7319 | 0,9513 | 54,62% | 46,38% |
| **SimpleLinearRegression** | 0,6406 | 1,1317 | 1,5904 | 84,46% | 77,55% |
| **SMOreg** | 0,8342 | 0,8837 | 1,1380 | 65,95% | 55,49% |
| **IBK** | 0,7159 | 1,1250 | 1,5523 | 83,96% | 75,69% |
| **Kstar** | 0,8405 | 0,6858 | 1,1297 | 51,18% | 55,08% |
| **LWL** | 0,7870 | 0,8318 | 1,2672 | 62,07% | 61,79% |
| **AdditiveRegression** | 0,7908 | 0,9169 | 1,2711 | 68,42% | 61,98% |
| **Bagging** | 0,8454 | 0,7398 | 1,1520 | 55,21% | 56,17% |
| **CVParameterSelection** | -0,2779 | 1,3400 | 2,0508 | 100,00% | 100,00% |
| **RandomCommittee** | 0,8292 | 0,7659 | 1,1384 | 57,16% | 55,51% |
| **RandomSubSpace** | 0,8230 | 0,8911 | 1,2826 | 66,50% | 62,54% |
| **RegressionByDiscretization** | 0,7213 | 0,9584 | 1,4483 | 71,52% | 70,62% |
| **Stacking** | -0,2779 | 1,3400 | 2,0508 | 100,00% | 100,00% |
| **InputMappedClassifier** | -0,2779 | 1,3400 | 2,0508 | 100,00% | 100,00% |
| **DecisionTable** | 0,7162 | 0,9546 | 1,4314 | 71,24% | 69,80% |
| **M5Rules** | 0,8748 | 0,6243 | 0,9933 | 46,59% | 48,44% |
| **ZeroR** | -0,2779 | 1,3400 | 2,0508 | 100,00% | 100,00% |
| **DecisionStump** | 0,6145 | 1,0491 | 1,6307 | 78,29% | 79,51% |
| **M5P** | 0,9090 | 0,5971 | 0,8474 | 44,58% | 41,32% |
| **RandomForest** | 0,8800 | 0,7311 | 0,9775 | 54,56% | 47,66% |
| **RandomTree** | 0,6309 | 1,0905 | 1,7670 | 81,38% | 86,16% |
| **REPTree** | 0,6749 | 1,0151 | 1,5490 | 75,75% | 75,53% |

*Tabla 54: Regresión para Mejores correlados.*

Como ya se ha comentado varias veces vamos a generar un diagrama box plot para ver que variable tiene menos dispersión y menos valores extraños para elegir esta como ayuda para la elección del mejor algoritmo, el diagrama resultado para este Dataset sería el siguiente:

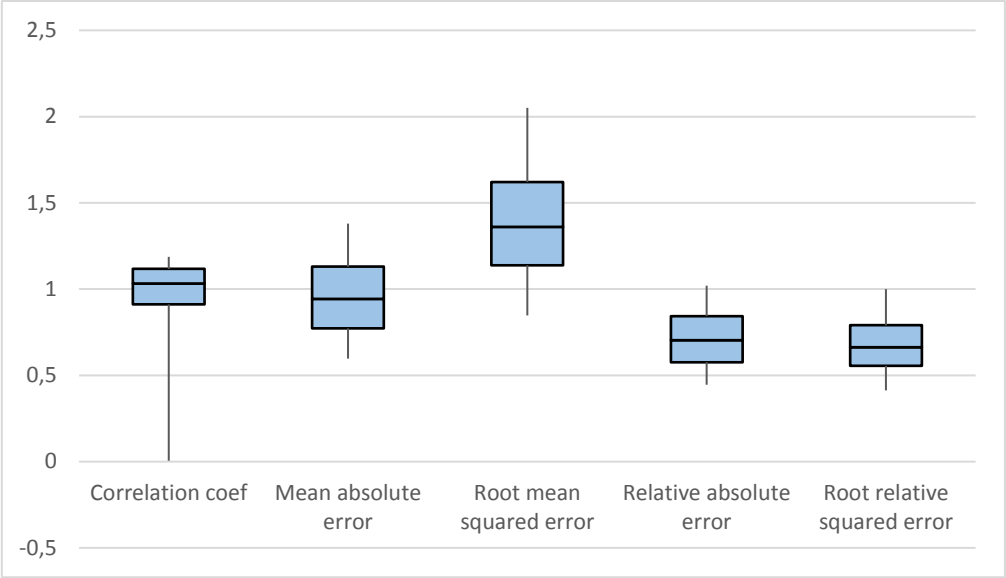



*Ilustración 50:Diagrama para Mejores correlados en regresión.*

Como pasaba en el caso anterior se puede eliminar de la lista dos variables, estas son *"Correlation coefficient" y "Root mean squared error"* y por la mismas razones que se comentamos en el punto anterior, de las tres que nos quedan la variable *"Mean Absolute error"* se podría quitar también por tener más dispersión que las demás y de las dos que nos queda ya es más difícil de elegir por que la diferencia en la dispersión es casi mínima por eso se va a intentar encontrar un algoritmo con un equilibrio de estas dos variables. Entonces nos podríamos quedar con los siguientes algoritmos, "M5P" con un valor de la variable *"Relative Absolute error"* de 44,5% y un valor de la variable *"Root relative squared error"* de 41,3% y el otro algoritmo puede ser *"M5Rules"* con un valor en la primera variable de 46,5% y en la segunda variable un valor de 48,4%. Como estos dos algoritmos ya han sido explicados no se van a comentar nada más de ellos.

## 7.3. CLASIFICACIÓN:

En la siguiente y última parte de esta investigación lo que hemos realizado es clasificación, para ver si nos creemos la nota que nos ha dicho el alumno o no.

Podríamos decir que la clasificación [22] es el problema de identificar a cuál de un conjunto de categorías (subpoblaciones) pertenece una observación, todo esto sobre un conjunto de datos que contiene una serie de instancia cuya categoría de que miembro pertenece es conocida. Entonces podemos decir que la clasificación es un ejemplo del problema de reconocimiento de patrones.

Para este problema también hemos realizados la división de los Dataset que hemos explicado en el punto anterior por lo cual tenemos los cuatro Dataset con el nombre, *"TodosAtributos", "SoloProfes", "SoloAlumno", "AtributoSelecionados" y "MejoresCorrelados".*

Por lo tanto, lo primero que hemos tenido que hacer es clasificar los patrones que ya tenemos, esto lo hemos hecho de la siguiente forma, si la diferencia entre la nota de autoevaluación y la que el alumno ha sacado en el examen es menos a una frontera.

Entonces de primeras se escogió como frontera el valor de un punto de diferencia, pero se pensó que en algunas situaciones podría ser demasiado una diferencia de un punto por eso también se realizó toda la experimentación con una diferencia de medio punto.

Con todo esto establecido lo que hicimos fue entonces algo parecido a lo realizado en el punto anterior pero esta vez aplicamos todos los algoritmos posibles para clasificación que nos proporciona la aplicación "Weka". Estos son: *"BayesNet", "NaiveBayes", "Logistic", "MultilayerPerception", "RBFClassifier", "RBFNetwork", "SGD", "SGDText", "SimpleLogistic", "SMO", "VotedPerceptron", "IB1", "IBK", "Kstar", "LWL", "AdaBoostM1", "AtributeSelectedClassifier", "Bagging", "ClassificationViaRegression", "CVParameterSelection", "IterativeClassifierOptimizer", "LogitBoost", "MultiBoostAB", "MultiClassClassifier", "MultiScheme", "RandomCommittee", "RandomSubSpace", "Stacking", "Vote", "InputMappedClassifier", "DecisionTable", "Jrip", "OneR", "Part", "ZeroR", "DecisionStump", "HoeffdingTree", "J48", "LMT", "RandomForest", "RandomTree" y "REPTree".*

Todos estos algoritmos nos proporcionan una serie de variables como resultado, nosotros nos hemos quedado para decidir que algoritmo es mejor las siguientes variables:

- **Correctly Classified Instances:** Es el porcentaje de instancias de test que ha clasificado correctamente el algoritmo.

- **Precision:** [23] (también llamado “positive predictive value”), es el porcentaje de patrones positivos predichos como positivos, frente al total de patrones predichos como positivos.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

- **Recall:** [23] (también conocido como “sensitivity”), es el porcentaje de patrones positivos predichos como positivos.

$$Recall = \frac{True\ Positive}{True\ Positive + \ False\ Negative}$$

- **F-Measure:** [24] Es una medida de la precisión de una prueba y se puede definir como la media armónica ponderada de Precision y Recall.

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall}$$
$$= \frac{2 * True\ Positive}{2 * True\ Positive + False\ Positive + False\ Negative}$$

- **ROC Area:** [25] Este índice se puede interpretar como la probabilidad de que un clasificador ordenara o puntuara una instancia positiva elegida aleatoriamente más alta que una negativa.

Con todo esto podríamos obtener los siguientes resultados:

- **Frontera de medio punto**
    - **Todos Atributos**

| | **Correctly Classified Instances** | **Precision** | **Recall** | **F-Measure** | **ROC Area** |
|---|---|---|---|---|---|
| **BayesNet** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **NaiveBayes** | 60,26% | 0,628 | 0,603 | 0,615 | 0,415 |
| **Logistic** | 65,38% | 0,690 | 0,654 | 0,670 | 0,502 |
| **MultilayerPerception** | 70,51% | 0,683 | 0,706 | 0,693 | 0,384 |
| **RBFClassifier** | 70,51% | 0,628 | 0,769 | 0,691 | 0,474 |
| **RBFNetwork** | 67,95% | 0,678 | 0,679 | 0,683 | 0,555 |
| **SGD** | 61,54% | 0,596 | 0,615 | 0,606 | 0,403 |
| **SGDText** | 79,49% | 0,632 | 0,795 | 0,704 | 0,500 |
| **SimpleLogistic** | 74,36% | 0,623 | 0,744 | 0,678 | 0,434 |
| **SMO** | 70,51% | 0,616 | 0,705 | 0,657 | 0,446 |
| **VotedPerceptron** | 79,49% | 0,632 | 0,795 | 0,704 | 0,526 |
| **IB1** | 64,10% | 0,641 | 0,641 | 0,641 | 0,454 |

| | | | | | |
|---|---|---|---|---|---|
| **IBK** | 64,10% | 0,641 | 0,641 | 0,641 | 0,493 |
| **Kstar** | 28,21% | 0,739 | 0,282 | 0,231 | 0,513 |
| **LWL** | 73,08% | 0,621 | 0,731 | 0,671 | 0,411 |
| **AdaBoostM1** | 66,67% | 0,608 | 0,667 | 0,636 | 0,289 |
| **AtributeSelectedClassifier** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **Bagging** | 78,21% | 0,630 | 0,782 | 0,698 | 0,418 |
| **ClassificationViaRegression** | 74,36% | 0,623 | 0,744 | 0,678 | 0,396 |
| **CVParameterSelection** | 79,49% | 0,632 | 0,795 | 0,704 | 0,418 |
| **IterativeClassifierOptimizer** | 74,36% | 0,623 | 0,744 | 0,678 | 0,364 |
| **LogitBoost** | 67,95% | 0,636 | 0,679 | 0,656 | 0,301 |
| **MultiBoostAB** | 74,36% | 0,623 | 0,744 | 0,678 | 0,353 |
| **MultiClassClassifier** | 65,38% | 0,690 | 0,654 | 0,670 | 0,502 |
| **MultiScheme** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **RandomCommittee** | 74,36% | 0,663 | 0,744 | 0,659 | 0,472 |
| **RandomSubSpace** | 79,49% | 0,632 | 0,795 | 0,704 | 0,349 |
| **Stacking** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **Vote** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **InputMappedClassifier** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **DecisionTable** | 79,49% | 0,632 | 0,795 | 0,704 | 0,483 |
| **Jrip** | 74,36% | 0,623 | 0,744 | 0,678 | 0,441 |
| **OneR** | 66,67% | 0,608 | 0,667 | 0,636 | 0,434 |
| **Part** | 70,51% | 0,666 | 0,705 | 0,684 | 0,422 |
| **ZeroR** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **DecisionStump** | 79,64% | 0,625 | 0,756 | 0,685 | 0,337 |
| **HoeffdingTree** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **J48** | 78,21% | 0,704 | 0,782 | 0,718 | 0,458 |
| **LMT** | 74,36% | 0,623 | 0,744 | 0,678 | 0,434 |
| **RandomForest** | 78,21% | 0,630 | 0,782 | 0,698 | 0,426 |
| **RandomTree** | 69,23% | 0,706 | 0,692 | 0,699 | 0,528 |
| **REPTree** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |

*Tabla 55:Clasificación para TodosAtributos con frontera de medio punto.*

En clasificación se va a hacer lo mismo que se viene haciendo en el apartado de regresión, entonces hemos creado un diagrama box plot y vamos a elegir la variable con menos dispersión y valores extraños para que sea la representativa de este Dataset, pues el diagrama que resulta sería el siguiente:

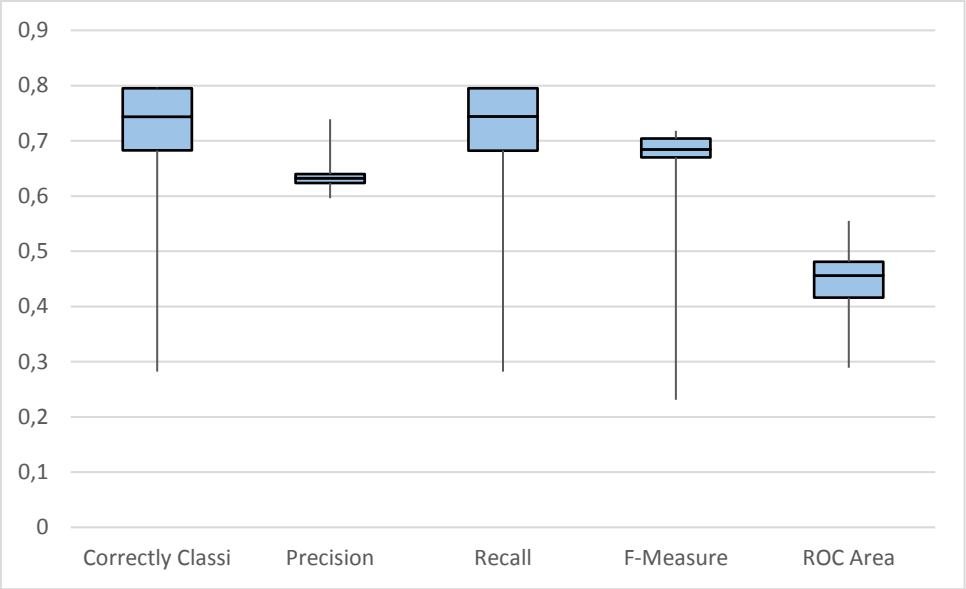



*Ilustración 51: Diagrama para TodosAtributos en Clasificación con frontera de medio punto.*

En este diagrama se ve rápidamente que variable sería la mejor para elegir, esta sería *"Precision"* ya que es la que menos dispersión tiene y los valores máximos y mínimos no están tan alegados de la media como los demás.

Entonces nos podríamos quedar con los siguientes algoritmos, "Kstar" con un valor en la variable de 0,739 y también "RandomTree" con un valor de 0,706, estos algoritmos se podrían definir de la siguiente manera:

- **"Kstar":** [26] Es un clasificador basado en instancias, es decir, la clase de una instancia de prueba se basa en la clase de las instancias similares a ella, según lo determinado por alguna función de similitud. Se diferencia de otros algoritmos basados en instancias en que este utiliza una función de distancia basado en entropía.

- **"RandomTree":** [27] Es un clasificador para construir un árbol de forma aleatoria, este considera K atributos elegidos al azar en cada nodo, no realiza ninguna poda en ningún momento.

- **Solo Profes:**

| | Correctly Classified Instances | Precision | Recall | F-Measure | ROC Area |
|---|---|---|---|---|---|
| **BayesNet** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **NaiveBayes** | 56,41% | 0,615 | 0,564 | 0,587 | 0,431 |
| **Logistic** | 58,97% | 0,666 | 0,590 | 0,621 | 0,478 |
| **MultilayerPerception** | 64,10% | 0,641 | 0,641 | 0,641 | 0,403 |
| **RBFClassifier** | 76,92% | 0,628 | 0,769 | 0,691 | 0,467 |
| **RBFNetwork** | 69,23% | 0,677 | 0,692 | 0,684 | 0,471 |
| **SGD** | 69,23% | 0,640 | 0,692 | 0,664 | 0,450 |
| **SGDText** | 79,49% | 0,632 | 0,795 | 0,704 | 0,500 |
| **SimpleLogistic** | 74,36% | 0,623 | 0,744 | 0,678 | 0,443 |
| **SMO** | 75,64% | 0,625 | 0,756 | 0,685 | 0,477 |
| **VotedPerceptron** | 79,49% | 0,632 | 0,795 | 0,704 | 0,504 |
| **IB1** | 62,82% | 0,637 | 0,628 | 0,632 | 0,447 |
| **IBK** | 62,82% | 0,637 | 0,628 | 0,632 | 0,486 |
| **Kstar** | 28,21% | 0,739 | 0,282 | 0,231 | 0,512 |
| **LWL** | 75,64% | 0,625 | 0,756 | 0,685 | 0,385 |
| **AdaBoostM1** | 67,95% | 0,636 | 0,679 | 0,656 | 0,328 |
| **AtributeSelectedClassifier** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **Bagging** | 78,21% | 0,630 | 0,782 | 0,698 | 0,578 |
| **ClassificationViaRegression** | 79,49% | 0,741 | 0,795 | 0,726 | 0,452 |
| **CVParameterSelection** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **IterativeClassifierOptimizer** | 74,36% | 0,623 | 0,744 | 0,678 | 0,364 |
| **LogitBoost** | 62,82% | 0,599 | 0,628 | 0,613 | 0,338 |
| **MultiBoostAB** | 73,08% | 0,621 | 0,731 | 0,671 | 0,355 |
| **MultiClassClassifier** | 58,97% | 0,666 | 0,590 | 0,621 | 0,478 |
| **MultiScheme** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **RandomCommittee** | 71,79% | 0,650 | 0,718 | 0,679 | 0,446 |
| **RandomSubSpace** | 79,49% | 0,632 | 0,795 | 0,704 | 0,496 |
| **Stacking** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **Vote** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **InputMappedClassifier** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **DecisionTable** | 76,92% | 0,628 | 0,769 | 0,691 | 0,431 |
| **Jrip** | 78,21% | 0,630 | 0,782 | 0,698 | 0,482 |
| **OneR** | 70,51% | 0,645 | 0,705 | 0,672 | 0,469 |
| **Part** | 71,79% | 0,672 | 0,718 | 0,692 | 0,388 |
| **ZeroR** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **DecisionStump** | 75,64% | 0,625 | 0,756 | 0,685 | 0,337 |
| **HoeffdingTree** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **J48** | 78,21% | 0,704 | 0,572 | 0,718 | 0,440 |
| **LMT** | 74,36% | 0,623 | 0,744 | 0,678 | 0,443 |
| **RandomForest** | 78,21% | 0,630 | 0,782 | 0,698 | 0,414 |
| **RandomTree** | 70,51% | 0,683 | 0,705 | 0,693 | 0,519 |

| | | | | | |
|---|---|---|---|---|---|
| **REPTree** | 78,21% | 0,630 | 0,782 | 0,698 | 0,502 |

*Tabla 56: Clasificación para SoloProfes con frontera de medio punto.*

Como en apartados anteriores se ha generado un diagrama para ver la distribución de las diferentes variables en este Dataset, el resultado sería el siguiente:

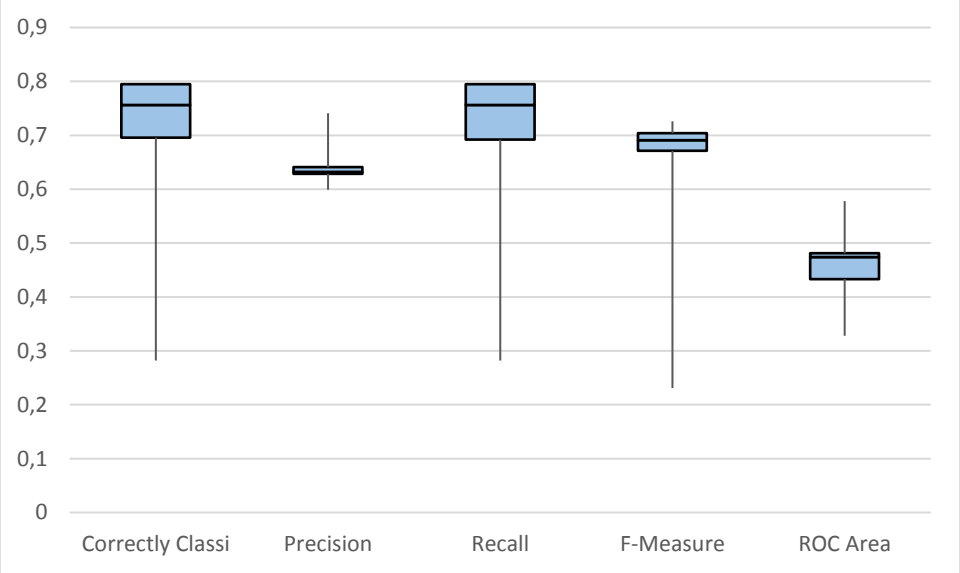



*Ilustración 52: Diagrama para SoloProfes en clasificación con frontera de medio punto.*

En este caso pasa como en el Dataset anterior la variable que se podría elegir con las condiciones que hemos puesto sería *"Precision",* y por lo tanto los algoritmos que podríamos elegir podrían ser lo siguiente, "Kstar" con un valor en la variable de 0,739 y "ClassificationViaRegression" con un valor de 0,741. Solo se explicará "ClassificationViaRegression" ya que es el único del cual no hemos hablado todavía.

- **ClassificationViaRegression**: [28] Este algoritmo lo que realiza es una clasificación usando algún método de regresión. Lo primero que realiza es binarizar las clases y entonces la salida ya son números y consiste en averiguar qué salida binaria nos da y luego a pasarla otra vez a la clase que más se asemeja.

- **Solo Alumnos:**

| | Correctly Classified Instances | Precision | Recall | F-Measure | ROC Area |
|---|---|---|---|---|---|
| **BayesNet** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **NaiveBayes** | 74,36% | 0,623 | 0,744 | 0,678 | 0,375 |
| **Logistic** | 79,49% | 0,741 | 0,795 | 0,736 | 0,475 |
| **MultilayerPerception** | 75,64% | 0,673 | 0,756 | 0,703 | 0,454 |
| **RBFClassifier** | 79,49% | 0,632 | 0,795 | 0,704 | 0,399 |
| **RBFNetwork** | 75,64% | 0,250 | 0,756 | 0,685 | 0,304 |
| **SGD** | 79,49% | 0,632 | 0,795 | 0,704 | 0,500 |
| **SGDText** | 79,49% | 0,632 | 0,795 | 0,704 | 0,500 |
| **SimpleLogistic** | 79,49% | 0,632 | 0,795 | 0,704 | 0,435 |
| **SMO** | 79,49% | 0,632 | 0,795 | 0,704 | 0,500 |
| **VotedPerceptron** | 79,49% | 0,632 | 0,795 | 0,704 | 0,492 |
| **IB1** | 61,54% | 0,596 | 0,615 | 0,606 | 0,415 |
| **IBK** | 61,54% | 0,596 | 0,615 | 0,606 | 0,415 |
| **Kstar** | 65,38% | 0,605 | 0,654 | 0,629 | 0,397 |
| **LWL** | 75,64% | 0,625 | 0,756 | 0,685 | 0,258 |
| **AdaBoostM1** | 71,79% | 0,618 | 0,718 | 0,664 | 0,251 |
| **AtributeSelectedClassifier** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **Bagging** | 79,49% | 0,632 | 0,795 | 0,704 | 0,466 |
| **ClassificationViaRegression** | 79,49% | 0,632 | 0,795 | 0,704 | 0,432 |
| **CVParameterSelection** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **IterativeClassifierOptimizer** | 75,64% | 0,625 | 0,756 | 0,685 | 0,322 |
| **LogitBoost** | 70,51% | 0,645 | 0,705 | 0,672 | 0,265 |
| **MultiBoostAB** | 76,92% | 0,628 | 0,769 | 0,691 | 0,332 |
| **MultiClassClassifier** | 65,38% | 0,605 | 0,654 | 0,629 | 0,487 |
| **MultiScheme** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **RandomCommittee** | 62,82% | 0,619 | 0,628 | 0,624 | 0,422 |
| **RandomSubSpace** | 79,49% | 0,632 | 0,795 | 0,704 | 0,505 |
| **Stacking** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **Vote** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **InputMappedClassifier** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **DecisionTable** | 79,49% | 0,632 | 0,795 | 0,704 | 0,463 |
| **Jrip** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **OneR** | 70,51% | 0,616 | 0,705 | 0,657 | 0,458 |
| **Part** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **ZeroR** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **DecisionStump** | 75,64% | 0,625 | 0,756 | 0,685 | 0,304 |
| **HoeffdingTree** | 79,49% | 0,632 | 0,795 | 0,704 | 0,495 |
| **J48** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **LMT** | 79,49% | 0,632 | 0,795 | 0,704 | 0,435 |
| **RandomForest** | 70,51% | 0,616 | 0,705 | 0,657 | 0,354 |
| **RandomTree** | 79,49% | 0,741 | 0,795 | 0,726 | 0,475 |

| | | | | | |
|---|---|---|---|---|---|
| **REPTree** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |

*Tabla 57:Clasificación para SoloAlumnos con frontera de medio punto.*

A continuación, se puede observar el diagrama resultante para este Dataset:

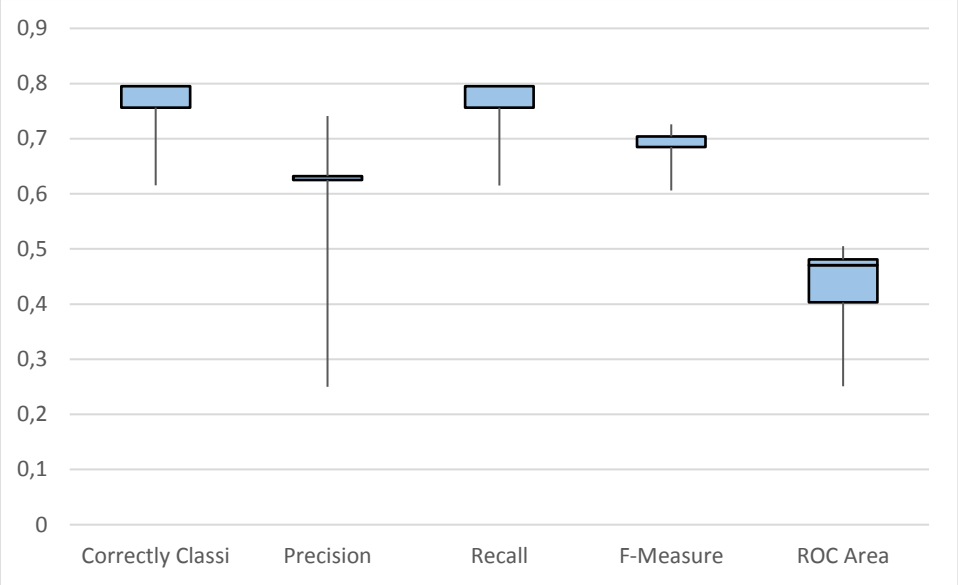



*Ilustración 53: Diagrama para SoloAlumnos en clasificación con frontera de medio punto.*

Se puede observar rápidamente que la variable *"Precision"* es la que menos dispersión tiene como sucedía en los casos anteriores, pero como vemos tiene algún valor extraño por la gran lejanía de su máximo y mínimo, por esta razón se va a descartar su elección, por lo tanto, para este Dataset nos quedaríamos con la siguiente con menos dispersión y esta seria *"F-Measure"*.

Con esto elegido podemos ver que los mejores algoritmos para este Dataset podrían ser, "Logistic" con un valor de la variable *"F-Measure"* de 0,736 y "RandomTree" con un valor de 0,726. Este algoritmo se podría definir de la siguiente forma:

- **Logistic:** [29] Este algoritmo lo que intenta hacer es construir y usar un modelo de regresión logística multinomial con un estimador de cresta. Entonces si tenemos k clases para n instancias con m atributos, la matriz de parámetros B que se calculará será de tamaño m*(k-1), este algoritmo utiliza el método Cuasi-Newton.

- **Atributos Seleccionados:**

| | Correctly Classified Instances | Precision | Recall | F-Measure | ROC Area |
|---|---|---|---|---|---|
| **BayesNet** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **NaiveBayes** | 66,67% | 0,608 | 0,667 | 0,636 | 0,335 |
| **Logistic** | 69,23% | 0,677 | 0,692 | 0,684 | 0,515 |
| **MultilayerPerception** | 67,95% | 0,672 | 0,679 | 0,676 | 0,438 |
| **RBFClassifier** | 79,49% | 0,632 | 0,795 | 0,704 | 0,459 |
| **RBFNetwork** | 70,51% | 0,666 | 0,705 | 0,684 | 0,513 |
| **SGD** | 78,21% | 0,704 | 0,782 | 0,718 | 0,516 |
| **SGDText** | 79,49% | 0,632 | 0,795 | 0,704 | 0,500 |
| **SimpleLogistic** | 76,92% | 0,685 | 0,769 | 0,710 | 0,398 |
| **SMO** | 79,49% | 0,632 | 0,795 | 0,704 | 0,500 |
| **VotedPerceptron** | 78,21% | 0,630 | 0,782 | 0,698 | 0,500 |
| **IB1** | 73,08% | 0,679 | 0,731 | 0,700 | 0,497 |
| **IBK** | 73,08% | 0,679 | 0,731 | 0,700 | 0,525 |
| **Kstar** | 70,51% | 0,698 | 0,705 | 0,702 | 0,543 |
| **LWL** | 75,64% | 0,625 | 0,756 | 0,685 | 0,436 |
| **AdaBoostM1** | 75,64% | 0,673 | 0,756 | 0,703 | 0,360 |
| **AtributeSelectedClassifier** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **Bagging** | 79,49% | 0,632 | 0,795 | 0,704 | 0,521 |
| **ClassificationViaRegression** | 76,92% | 0,628 | 0,769 | 0,691 | 0,356 |
| **CVParameterSelection** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **IterativeClassifierOptimizer** | 76,92% | 0,628 | 0,769 | 0,691 | 0,377 |
| **LogitBoost** | 67,95% | 0,611 | 0,679 | 0,643 | 0,485 |
| **MultiBoostAB** | 75,64% | 0,625 | 0,756 | 0,685 | 0,388 |
| **MultiClassClassifier** | 69,23% | 0,677 | 0,692 | 0,684 | 0,515 |
| **MultiScheme** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **RandomCommittee** | 74,36% | 0,704 | 0,744 | 0,720 | 0,534 |
| **RandomSubSpace** | 79,49% | 0,632 | 0,495 | 0,704 | 0,466 |
| **Stacking** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **Vote** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **InputMappedClassifier** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **DecisionTable** | 79,49% | 0,632 | 0,795 | 0,704 | 0,483 |
| **Jrip** | 80,77% | 0,845 | 0,808 | 0,733 | 0,513 |
| **OneR** | 71,79% | 0,672 | 0,718 | 0,692 | 0,477 |
| **Part** | 74,36% | 0,719 | 0,744 | 0,729 | 0,432 |
| **ZeroR** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **DecisionStump** | 79,49% | 0,632 | 0,795 | 0,704 | 0,347 |
| **HoeffdingTree** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **J48** | 67,95% | 0,636 | 0,679 | 0,656 | 0,370 |
| **LMT** | 76,92% | 0,685 | 0,769 | 0,710 | 0,398 |
| **RandomForest** | 74,36% | 0,623 | 0,744 | 0,678 | 0,506 |
| **RandomTree** | 67,95% | 0,672 | 0,679 | 0,676 | 0,523 |

| | | | | | |
|---|---|---|---|---|---|
| **REPTree** | 80,77% | 0,845 | 0,808 | 0,733 | 0,502 |

*Tabla 58:Clasificación para Atributos Seleccionados con frontera de medio punto.*

El diagrama resultado para este conjunto de datos podría ser el que vemos a continuación:

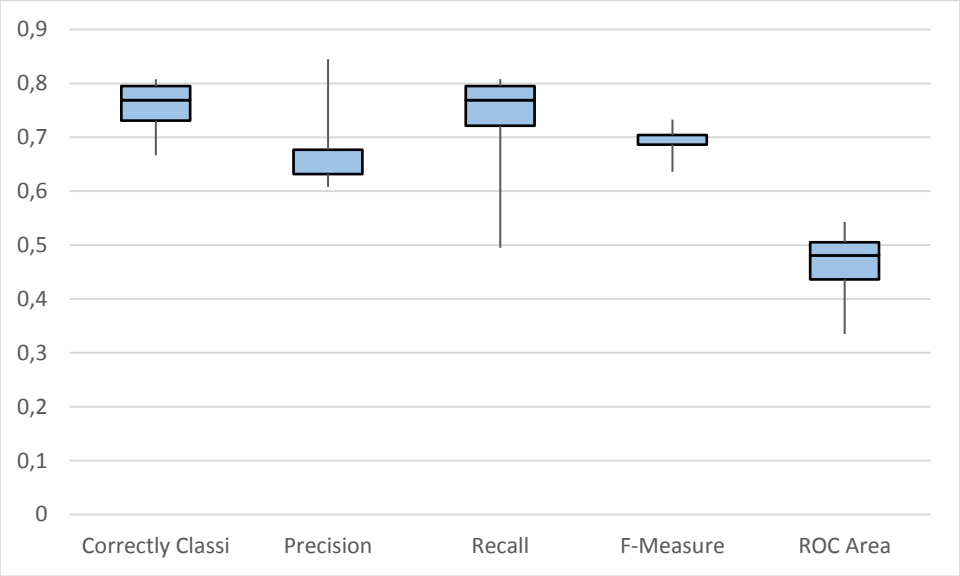



*Ilustración 54: Diagrama para Atributos Seleccionados en clasificación con frontera de medio punto.*

Fácilmente se pude observar que las variables con menos dispersión en este caso son, *"Precision" y "F-Measuere"* pero tenemos la misma situación del caso anterior y es que *"Precision"* tiene un "whisker " ("bigote", en español) con una variable extraña ya que su máximo está más alegado de lo normal de la media, entonces por esta razón nos vamos a quedar con la variable *"F-Measure"*, con esto elegido vamos ahora a decidir que algoritmos serían los mejores para este conjunto de datos.

Los algoritmos escogidos son "Jrip" y "REPTree" los dos con un valor de la variable *"F-Measure"* de 0,733, estos clasificadores se podrían definir de la manera que podemos ver a continuación.

- **Jrip:** [30] Este algoritmo lo que hace es implementar un conjunto de reglas de aprendizaje para poder averiguar a la clase que pertenece cada instancia, utiliza poda para reducir los errores cometidos, este algoritmo siempre realiza las mismas etapas y estas son, inicialización y construcción, crecimiento en la que se va construyendo las reglas, fase de poda y por último optimización de lo que quede.

- **REPTree:** [31] Este algoritmo construye un árbol de decisión/regresión usando ganancia/varianza de información y la poda es usada para reducir los errores. Sole ordena valores para atributos numéricos una vez y los valores faltantes se tratan dividiendo las instancias correspondientes en partes.

- **Mejores correlados:**

| | Correctly Classified Instances | Precision | Recall | F-Measure | ROC Area |
|---|---|---|---|---|---|
| **BayesNet** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **NaiveBayes** | 71,79% | 0,650 | 0,718 | 0,679 | 0,248 |
| **Logistic** | 70,51% | 0,683 | 0,705 | 0,693 | 0,528 |
| **MultilayerPerception** | 65,38% | 0,662 | 0,654 | 0,658 | 0,468 |
| **RBFClassifier** | 79,49% | 0,632 | 0,795 | 0,704 | 0,427 |
| **RBFNetwork** | 73,08% | 0,656 | 0,731 | 0,687 | 0,382 |
| **SGD** | 78,21% | 0,630 | 0,782 | 0,698 | 0,504 |
| **SGDText** | 79,49% | 0,632 | 0,795 | 0,704 | 0,500 |
| **SimpleLogistic** | 79,49% | 0,632 | 0,795 | 0,704 | 0,307 |
| **SMO** | 79,49% | 0,632 | 0,795 | 0,704 | 0,500 |
| **VotedPerceptron** | 79,49% | 0,632 | 0,795 | 0,704 | 0,500 |
| **IB1** | 71,79% | 0,690 | 0,718 | 0,702 | 0,537 |
| **IBK** | 71,79% | 0,690 | 0,718 | 0,702 | 0,550 |
| **Kstar** | 69,23% | 0,677 | 0,692 | 0,684 | 0,509 |
| **LWL** | 76,92% | 0,685 | 0,769 | 0,710 | 0,357 |
| **AdaBoostM1** | 75,64% | 0,625 | 0,756 | 0,685 | 0,432 |
| **AtributeSelectedClassifier** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **Bagging** | 79,49% | 0,632 | 0,795 | 0,704 | 0,500 |
| **ClassificationViaRegression** | 78,21% | 0,704 | 0,782 | 0,718 | 0,370 |
| **CVParameterSelection** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **IterativeClassifierOptimizer** | 76,92% | 0,685 | 0,769 | 0,730 | 0,447 |
| **LogitBoost** | 73,08% | 0,656 | 0,731 | 0,687 | 0,488 |
| **MultiBoostAB** | 76,92% | 0,628 | 0,769 | 0,691 | 0,410 |
| **MultiClassClassifier** | 70,51% | 0,683 | 0,705 | 0,693 | 0,528 |
| **MultiScheme** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **RandomCommittee** | 73,08% | 0,711 | 0,731 | 0,720 | 0,550 |
| **RandomSubSpace** | 78,49% | 0,632 | 0,795 | 0,704 | 0,475 |
| **Stacking** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **Vote** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **InputMappedClassifier** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **DecisionTable** | 79,49% | 0,632 | 0,795 | 0,704 | 0,483 |
| **Jrip** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **OneR** | 70,51% | 0,616 | 0,705 | 0,657 | 0,458 |
| **Part** | 71,79% | 0,650 | 0,718 | 0,679 | 0,447 |

| | | | | | |
|---|---|---|---|---|---|
| **ZeroR** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **DecisionStump** | 76,92% | 0,628 | 0,769 | 0,691 | 0,368 |
| **HoeffdingTree** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |
| **J48** | 70,51% | 0,616 | 0,705 | 0,657 | 0,425 |
| **LMT** | 78,21% | 0,630 | 0,782 | 0,698 | 0,335 |
| **RandomForest** | 78,21% | 0,630 | 0,782 | 0,698 | 0,500 |
| **RandomTree** | 60,26% | 0,643 | 0,603 | 0,621 | 0,447 |
| **REPTree** | 79,49% | 0,632 | 0,795 | 0,704 | 0,481 |

*Tabla 59:Clasificación para Mejore Correlados con frontera de medio punto.*

El resultado de construir un diagrama box plot con los datos que nos ha salido en este Dataset sería el que se puede observar a continuación:

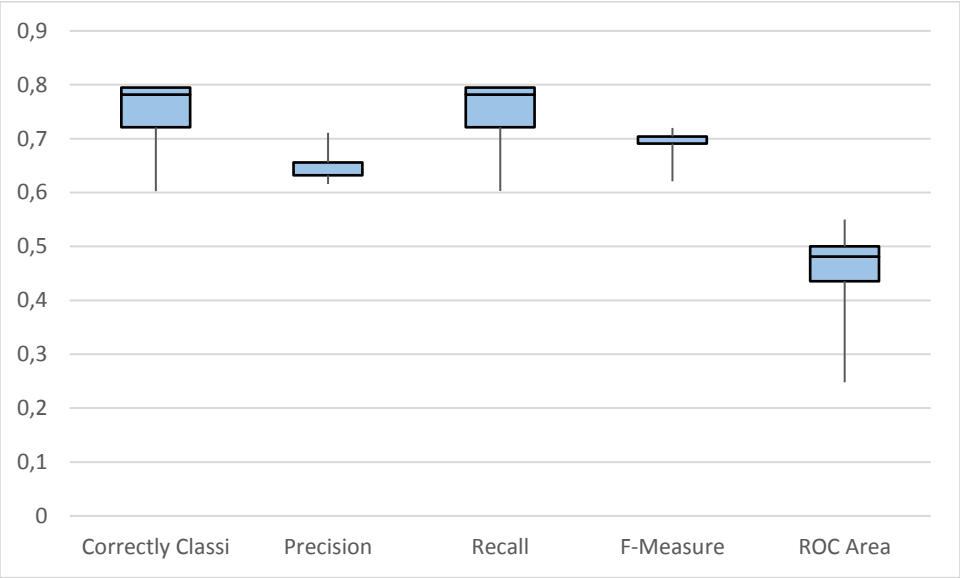



*Ilustración 55:Diagrama para Mejore Correlados en clasificación con frontera de medio punto.*

Se puede observar rápidamente que las variables *"Precision" y "F-Measure"* son las dos que menos dispersión o rango intercuartil tienen, con la diferencia es muy poca y luego una tiene un máximo alejado de la media y la otra tiene un mínimo, entonces vamos a tomar las dos variables para obtener la decisión de que algoritmo es el mejor.

Entonces podemos decir que los algoritmos que tiene mejor valor en estas variables podrían ser los siguientes, "ClassificationViaRegresion" con los valores de 0.709 en la variable *"Precision"* y de 0.718 en la variable de *"F-Measuere",* luego también se podría elegir el clasificador "IterateClassifierOptimizer" con los valores de 0.685 y de 0.73 respetivamente, en este caso solo se va a definir el algoritmo "IterateClassifierOptimizer" ya que del otro se ha hablado anteriormente.

- **IterateClassifierOptimizer:** [32] Elige el número óptimo de iteraciones para luego aplicarlo en un clasificador iterativo como por ejemplo “LogitBoost”, utiliza la validación cruzada o una evaluación de porcentaje de división.

- **Frontera de un punto:**
  - **Todos los atributos:**

| | Correctly Classified Instances | Precision | Recall | F-Measure | ROC Area |
|---|---|---|---|---|---|
| **BayesNet** | 61,54% | 0,379 | 0,615 | 0,469 | 0,510 |
| **NaiveBayes** | 62,82% | 0,607 | 0,628 | 0,599 | 0,468 |
| **NaiveBayesSimple** | 57,14% | 0,440 | 0,571 | 0,472 | 0,455 |
| **Logistic** | 58,97% | 0,575 | 0,590 | 0,579 | 0,545 |
| **MultilayerPerception** | 60,26% | 0,600 | 0,603 | 0,601 | 0,533 |
| **RBFClassifier** | 57,69% | 0,542 | 0,577 | 0,544 | 0,479 |
| **RBFNetwork** | 55,13% | 0,509 | 0,551 | 0,516 | 0,455 |
| **SGD** | 61,54% | 0,607 | 0,615 | 0,609 | 0,562 |
| **SGDText** | 61,54% | 0,379 | 0,615 | 0,469 | 0,500 |
| **SimpleLogistic** | 55,13% | 0,537 | 0,551 | 0,542 | 0,519 |
| **SMO** | 55,13% | 0,531 | 0,511 | 0,537 | 0,495 |
| **VotedPerceptron** | 60,26% | 0,533 | 0,603 | 0,502 | 0,491 |
| **IB1** | 55,13% | 0,524 | 0,551 | 0,531 | 0,491 |
| **IBK** | 55,13% | 0,524 | 0,551 | 0,531 | 0,494 |
| **Kstar** | 41,03% | 0,533 | 0,410 | 0,318 | 0,501 |
| **LWL** | 56,41% | 0,536 | 0,564 | 0,541 | 0,537 |
| **AdaBoostM1** | 55,13% | 0,531 | 0,551 | 0,537 | 0,529 |
| **AtributeSelectedClassifier** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **Bagging** | 56,41% | 0,512 | 0,564 | 0,516 | 0,527 |
| **ClassificationViaRegression** | 62,82% | 0,622 | 0,628 | 0,624 | 0,567 |
| **CVParameterSelection** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **IterativeClassifierOptimizer** | 38,46% | 0,148 | 0,385 | 0,214 | 0,500 |
| **LogitBoost** | 50,00% | 0,503 | 0,500 | 0,501 | 0,449 |
| **MultiBoostAB** | 57,69% | 0,580 | 0,577 | 0,578 | 0,471 |
| **MultiClassClassifier** | 58,97% | 0,575 | 0,590 | 0,579 | 0,545 |
| **MultiScheme** | 61,54% | 0,379 | 0,615 | 0,469 | 0,48 |
| **RandomCommittee** | 55,13% | 0,531 | 0,551 | 0,537 | 0,482 |
| **RandomSubSpace** | 56,41% | 0,465 | 0,564 | 0,479 | 0,381 |
| **Stacking** | 61,54% | 0,379 | 0,615 | 0,469 | 0,48 |
| **Vote** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **InputMappedClassifier** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **DecisionTable** | 56,41% | 0,553 | 0,564 | 0,557 | 0,421 |
| **Jrip** | 61,54% | 0,595 | 0,615 | 0,595 | 0,536 |
| **OneR** | 51,28% | 0,493 | 0,513 | 0,500 | 0,459 |

| | | | | | |
|---|---|---|---|---|---|
| **Part** | 52,56% | 0,464 | 0,526 | 0,479 | 0,467 |
| **ZeroR** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **DecisionStump** | 58,97% | 0,534 | 0,590 | 0,523 | 0,534 |
| **HoeffdingTree** | 61,54% | 0,379 | 0,615 | 0,469 | 0,503 |
| **J48** | 55,13% | 0,451 | 0,551 | 0,471 | 0,435 |
| **LMT** | 55,13% | 0,537 | 0,551 | 0,542 | 0,522 |
| **RandomForest** | 57,69% | 0,535 | 0,577 | 0,535 | 0,497 |
| **RandomTree** | 46,15% | 0,480 | 0,462 | 0,468 | 0,424 |
| **REPTree** | 62,82% | 0,642 | 0,628 | 0,517 | 0,491 |

*Tabla 60:Clasificación para todos los atributos con frontera de un punto.*

Para este conjunto de datos nos puede dar como resultado el siguiente diagrama que podemos ver:

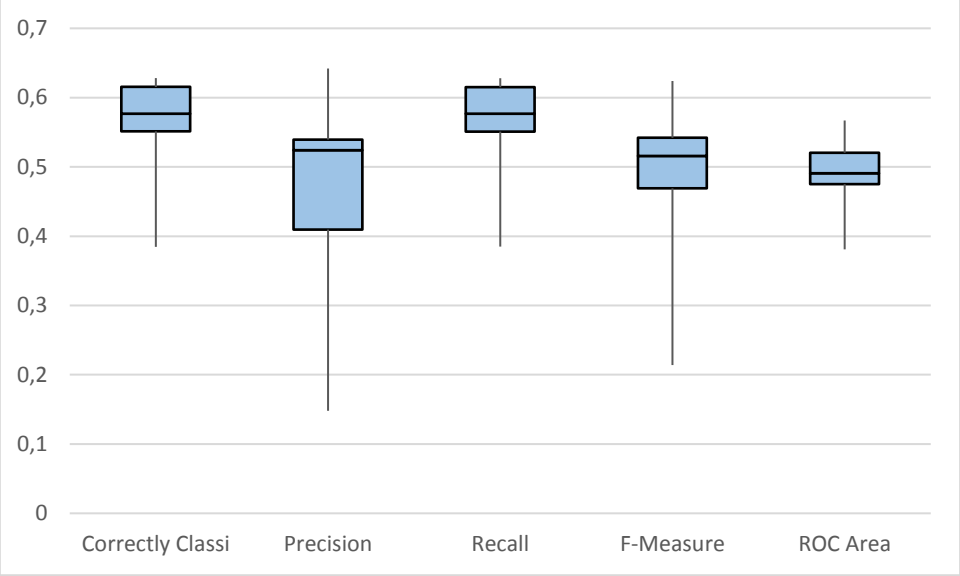



*Ilustración 56:Diagrama para todos los atributos en clasificación con frontera de un punto.*

En este diagrama se puede ver con rapidez que la variable *"Precision"* no se puede utilizar para este caso por que como vemos tenemos mucha dispersión y luego el valor máximo y mínimo está muy alegado de lo que es el valor medio pudiendo llegar a pensar que nos podríamos encontrar con valores de ruido. De las demás que nos queda la que se podría considerar con menos dispersión seria *"ROC Area"* por eso se va a utilizar esta, aunque tenga un desequilibrio ya que un cuartil está más cerca de la media que el otro, pero esto sucede para todas las variables.

Después de haber seleccionado que variable se va a utilizar vamos a proceder a elegir que clasificadores son los mejores para este caso, estos podrían ser los siguientes, por un lado, tenemos "SGD" con un valor de *"ROC Area"* de 0.562 y también se podría elegir el clasificador "ClasificationViaRegression" con un valor de 0.567, ahora como llevamos haciendo vamos a definir en que trata estos clasificadores.

- **SGD:** [33] Implementa el descenso de gradiente estocástico para aprender varios modelos lineales, globalmente reemplaza todos los valores perdidos y transforma atributos nominales en binarios. También normaliza todos los atributos por lo que los coeficientes en la salida se basan en los datos normalizados.

- **Solo Profe:**

| | Correctly Classified Instances | Precision | Recall | F-Measure | ROC Area |
|---|---|---|---|---|---|
| **BayesNet** | 61,54% | 0,379 | 0,615 | 0,469 | 0,510 |
| **NaiveBayes** | 56,41% | 0,512 | 0,564 | 0,516 | 0,444 |
| **NaiveBayesSimple** | 58,44% | 0,456 | 0,584 | 0,479 | 0,438 |
| **Logistic** | 51,28% | 0,519 | 0,513 | 0,515 | 0,481 |
| **MultilayerPerception** | 53,85% | 0,538 | 0,538 | 0,538 | 0,505 |
| **RBFClassifier** | 57,69% | 0,535 | 0,577 | 0,535 | 0,476 |
| **RBFNetwork** | 57,69% | 0,553 | 0,577 | 0,558 | 0,473 |
| **SGD** | 58,97% | 0,575 | 0,590 | 0,579 | 0,532 |
| **SGDText** | 61,54% | 0,379 | 0,615 | 0,469 | 0,500 |
| **SimpleLogistic** | 55,13% | 0,531 | 0,551 | 0,537 | 0,496 |
| **SMO** | 55,13% | 0,524 | 0,551 | 0,531 | 0,487 |
| **VotedPerceptron** | 57,69% | 0,369 | 0,577 | 0,450 | 0,488 |
| **IB1** | 51,28% | 0,493 | 0,513 | 0,500 | 0,463 |
| **IBK** | 51,28% | 0,493 | 0,513 | 0,500 | 0,471 |
| **Kstar** | 39,74% | 0,498 | 0,397 | 0,295 | 0,482 |
| **LWL** | 60,26% | 0,574 | 0,603 | 0,571 | 0,552 |
| **AdaBoostM1** | 53,85% | 0,513 | 0,538 | 0,521 | 0,527 |
| **AtributeSelectedClassifier** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **Bagging** | 65,38% | 0,641 | 0,654 | 0,620 | 0,617 |
| **ClassificationViaRegression** | 65,38% | 0,656 | 0,654 | 0,655 | 0,589 |
| **CVParameterSelection** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **IterativeClassifierOptimizer** | 38,46% | 0,148 | 0,385 | 0,214 | 0,500 |
| **LogitBoost** | 52,56% | 0,529 | 0,527 | 0,481 | 0,500 |
| **MultiBoostAB** | 55,13% | 0,554 | 0,551 | 0,553 | 0,492 |
| **MultiClassClassifier** | 51,28% | 0,519 | 0,513 | 0,515 | 0,481 |
| **MultiScheme** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **RandomCommittee** | 53,85% | 0,506 | 0,538 | 0,514 | 0,489 |
| **RandomSubSpace** | 58,97% | 0,470 | 0,590 | 0,477 | 0,541 |
| **Stacking** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **Vote** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **InputMappedClassifier** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **DecisionTable** | 56,41% | 0,553 | 0,564 | 0,557 | 0,432 |
| **Jrip** | 62,82% | 0,612 | 0,628 | 0,611 | 0,594 |
| **OneR** | 51,28% | 0,493 | 0,513 | 0,500 | 0,459 |

| | | | | | |
|---|---|---|---|---|---|
| **Part** | 51,28% | 0,418 | 0,513 | 0,447 | 0,471 |
| **ZeroR** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **DecisionStump** | 58,97% | 0,534 | 0,590 | 0,523 | 0,534 |
| **HoeffdingTree** | 61,54% | 0,379 | 0,615 | 0,469 | 0,500 |
| **J48** | 55,13% | 0,472 | 0,551 | 0,484 | 0,410 |
| **LMT** | 56,41% | 0,548 | 0,564 | 0,553 | 0,511 |
| **RandomForest** | 57,69% | 0,542 | 0,577 | 0,544 | 0,520 |
| **RandomTree** | 47,44% | 0,484 | 0,474 | 0,478 | 0,417 |
| **REPTree** | 61,54% | 0,573 | 0,615 | 0,419 | 0,471 |

*Tabla 61: Clasificación para SoloProfes con frontera de un punto.*

A continuación, podemos ver el diagrama resultado para este conjunto de datos:

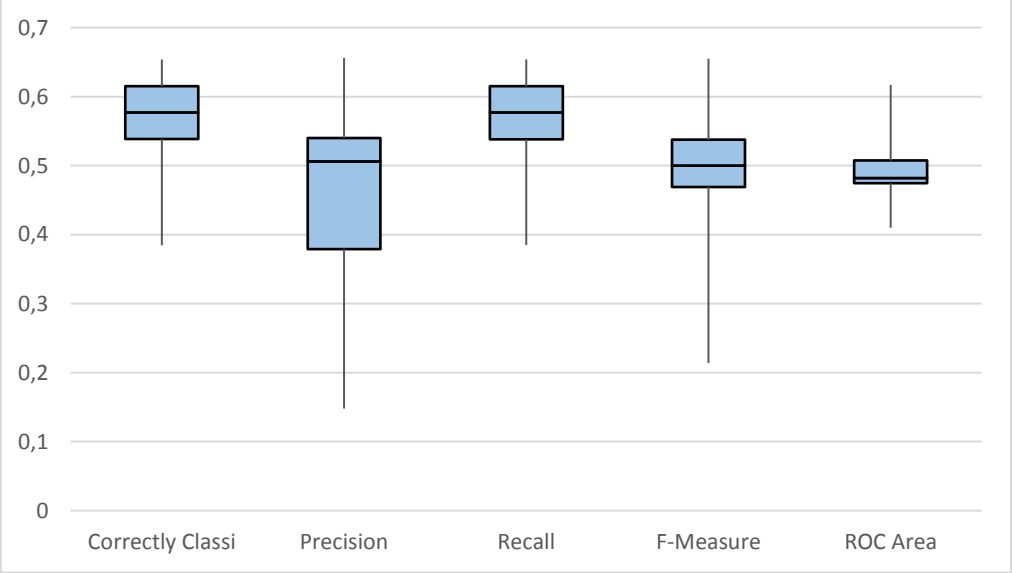



*Ilustración 57: Diagrama par SoloProfes en clasificación con frontera de un punto.*

En este caso se podría decir lo mismo que en el caso anterior, esto es lo siguiente que la variable *"Precision"* tiene mucha dispersión y no nos vale por eso, de las demás variables todas son más o menos iguales menos *"ROC Area"* que como se puede ver tiene mucha menos dispersión y por eso se va a elegir esta.

Entonces ya que tenemos la variable elegida vamos a seleccionar que algoritmo es el mejor, para este caso nos podríamos quedar con los siguientes, "Bagging" con un valor de área de la curva ROC de 0.617 y también con "ClassifficationViaRegresion" con un valor de 0.589.

- **Solo Alumno:**

| | Correctly Classified Instances | Precision | Recall | F-Measure | ROC Area |
|---|---|---|---|---|---|
| **BayesNet** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **NaiveBayes** | 62,82% | 0,606 | 0,628 | 0,519 | 0,577 |
| **NaiveBayesSimple** | 53,85% | 0,533 | 0,538 | 0,535 | 0,522 |
| **Logistic** | 57,69% | 0,503 | 0,577 | 0,501 | 0,486 |
| **MultilayerPerception** | 50,00% | 0,475 | 0,500 | 0,484 | 0,498 |
| **RBFClassifier** | 58,97% | 0,504 | 0,590 | 0,494 | 0,531 |
| **RBFNetwork** | 56,41% | 0,529 | 0,564 | 0,534 | 0,450 |
| **SGD** | 61,54% | 0,379 | 0,615 | 0,469 | 0,500 |
| **SGDText** | 61,54% | 0,379 | 0,615 | 0,469 | 0,500 |
| **SimpleLogistic** | 58,97% | 0,372 | 0,590 | 0,457 | 0,461 |
| **SMO** | 61,54% | 0,379 | 0,615 | 0,469 | 0,500 |
| **VotedPerceptron** | 60,26% | 0,376 | 0,603 | 0,463 | 0,482 |
| **IB1** | 48,42% | 0,465 | 0,487 | 0,474 | 0,452 |
| **IBK** | 48,72% | 0,465 | 0,487 | 0,474 | 0,404 |
| **Kstar** | 57,69% | 0,542 | 0,577 | 0,544 | 0,467 |
| **LWL** | 52,56% | 0,428 | 0,526 | 0,455 | 0,407 |
| **AdaBoostM1** | 55,13% | 0,488 | 0,551 | 0,496 | 0,410 |
| **AtributeSelectedClassifier** | 61,54% | 0,488 | 0,551 | 0,496 | 0,410 |
| **Bagging** | 55,13% | 0,499 | 0,551 | 0,507 | 0,410 |
| **ClassificationViaRegression** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **CVParameterSelection** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **IterativeClassifierOptimizer** | 41,03% | 0,478 | 0,410 | 0,383 | 0,450 |
| **LogitBoost** | 46,15% | 0,416 | 0,462 | 0,433 | 0,320 |
| **MultiBoostAB** | 56,41% | 0,432 | 0,564 | 0,462 | 0,411 |
| **MultiClassClassifier** | 57,69% | 0,503 | 0,577 | 0,501 | 0,486 |
| **MultiScheme** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **RandomCommittee** | 46,15% | 0,455 | 0,462 | 0,458 | 0,425 |
| **RandomSubSpace** | 61,54% | 0,379 | 0,615 | 0,469 | 0,468 |
| **Stacking** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **Vote** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **InputMappedClassifier** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **DecisionTable** | 61,54% | 0,379 | 0,615 | 0,469 | 0,490 |
| **Jrip** | 55,13% | 0,419 | 0,551 | 0,455 | 0,428 |
| **OneR** | 46,15% | 0,428 | 0,462 | 0,441 | 0,429 |
| **Part** | 60,26% | 0,376 | 0,603 | 0,463 | 0,470 |
| **ZeroR** | 61,54% | 0,379 | 0,615 | 0,465 | 0,480 |
| **DecisionStump** | 58,97% | 0,372 | 0,590 | 0,457 | 0,447 |
| **HoeffdingTree** | 58,97% | 0,522 | 0,590 | 0,509 | 0,463 |
| **J48** | 60,26% | 0,376 | 0,603 | 0,463 | 0,475 |
| **LMT** | 58,97% | 0,372 | 0,590 | 0,457 | 0,461 |
| **RandomForest** | 56,41% | 0,521 | 0,564 | 0,526 | 0,453 |

| | | | | | |
|---|---|---|---|---|---|
| **RandomTree** | 62,82% | 0,606 | 0,628 | 0,519 | 0,572 |
| **REPTree** | 60,46% | 0,376 | 0,603 | 0,463 | 0,492 |

*Tabla 62:Clasificación para Solo Alumnos con frontera de un punto.*

Para el conjunto de datos que se tenemos en la tabla 62 se obtendría el diagrama box plot que se puede observar en la ilustración siguiente:

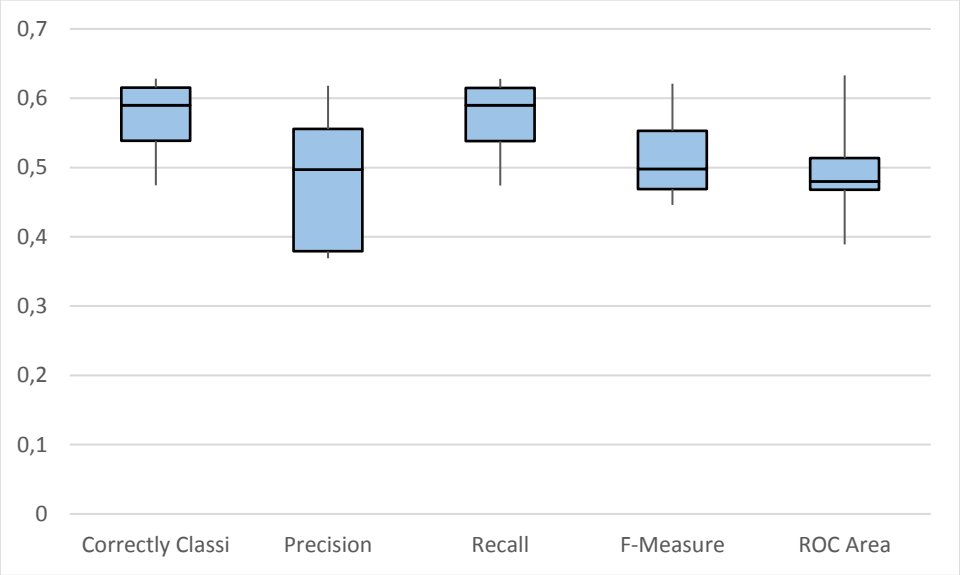



*Ilustración 58: Diagrama para Solo Alumnos en clasificación con frontera de un punto*

Entonces como ya nos viene pasando en todos los diagramas anteriores tenemos una variable la cual tiene una gran dispersión por lo tanto sus valores son muy distintos, en este caso esta variable sería *"Precision"*. Por las demás la que se puede ver con rapidez que es la que menos dispersión tiene y por lo tanto la variable que vamos a elegir sería *"ROC Area".*

Con esto seleccionado vamos a proceder a la elección de los clasificadores que se han considerado como mejores para este Dataset estos podrían ser, "NaiveBayes" con un área ROC de 0.577 y otro podría ser "RandomTree" con un área de 0.572.

Una descripción de lo que realiza estos dos clasificadores podría ser la siguiente:

- **NaiveBayes:** [34] Algoritmo para un clasificador NaiveBayes utilizando clases de estimador, los valores numéricos de precisión del estimador se eligen en base al análisis de los datos de entrenamiento, por esta razón este clasificador no es actualizable, si necesite que sea actualizable se debería usar el clasificador "NaiveBayesUpdateable".

- **Atributo Seleccionado:**

| | Correctly Classified Instances | Precision | Recall | F-Measure | ROC Area |
|---|---|---|---|---|---|
| **BayesNet** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **NaiveBayes** | 53,85% | 0,538 | 0,538 | 0,538 | 0,489 |
| **NaiveBayesSimple** | 52,56% | 0,529 | 0,526 | 0,527 | 0,469 |
| **Logistic** | 62,82% | 0,618 | 0,628 | 0,621 | 0,633 |
| **MultilayerPerception** | 52,56% | 0,529 | 0,526 | 0,527 | 0,436 |
| **RBFClassifier** | 53,85% | 0,497 | 0,538 | 0,506 | 0,462 |
| **RBFNetwork** | 55,13% | 0,554 | 0,551 | 0,553 | 0,514 |
| **SGD** | 53,85% | 0,520 | 0,538 | 0,526 | 0,482 |
| **SGDText** | 61,54% | 0,379 | 0,615 | 0,469 | 0,500 |
| **SimpleLogistic** | 61,54% | 0,595 | 0,615 | 0,595 | 0,571 |
| **SMO** | 58,97% | 0,555 | 0,590 | 0,553 | 0,512 |
| **VotedPerceptron** | 55,13% | 0,472 | 0,551 | 0,484 | 0,426 |
| **IB1** | 58,97% | 0,590 | 0,590 | 0,590 | 0,543 |
| **IBK** | 58,97% | 0,590 | 0,590 | 0,590 | 0,518 |
| **Kstar** | 56,41% | 0,570 | 0,564 | 0,566 | 0,566 |
| **LWL** | 53,85% | 0,460 | 0,538 | 0,476 | 0,389 |
| **AdaBoostM1** | 50,00% | 0,497 | 0,500 | 0,498 | 0,433 |
| **AtributeSelectedClassifier** | 61,54% | 0,379 | 0,615 | 0,469 | 0,48 |
| **Bagging** | 58,97% | 0,566 | 0,590 | 0,568 | 0,509 |
| **ClassificationViaRegression** | 53,85% | 0,487 | 0,538 | 0,498 | 0,571 |
| **CVParameterSelection** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **IterativeClassifierOptimizer** | 47,44% | 0,578 | 0,474 | 0,446 | 0,545 |
| **LogitBoost** | 53,85% | 0,538 | 0,538 | 0,538 | 0,533 |
| **MultiBoostAB** | 48,72% | 0,465 | 0,487 | 0,474 | 0,393 |
| **MultiClassClassifier** | 62,82% | 0,618 | 0,628 | 0,621 | 0,633 |
| **MultiScheme** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **RandomCommittee** | 53,85% | 0,556 | 0,538 | 0,544 | 0,513 |
| **RandomSubSpace** | 60,25% | 0,376 | 0,603 | 0,463 | 0,414 |
| **Stacking** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **Vote** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **InputMappedClassifier** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **DecisionTable** | 61,54% | 0,379 | 0,615 | 0,469 | 0,490 |
| **Jrip** | 56,41% | 0,521 | 0,564 | 0,526 | 0,474 |
| **OneR** | 57,69% | 0,553 | 0,577 | 0,558 | 0,510 |
| **Part** | 53,85% | 0,513 | 0,538 | 0,521 | 0,467 |
| **ZeroR** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **DecisionStump** | 61,54% | 0,379 | 0,615 | 0,469 | 0,403 |
| **HoeffdingTree** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **J48** | 50,00% | 0,475 | 0,500 | 0,484 | 0,485 |
| **LMT** | 60,26% | 0,579 | 0,603 | 0,578 | 0,528 |
| **RandomForest** | 60,26% | 0,579 | 0,603 | 0,578 | 0,493 |

| | | | | | |
|---|---|---|---|---|---|
| **RandomTree** | 47,44% | 0,478 | 0,474 | 0,476 | 0,465 |
| **REPTree** | 57,69% | 0,369 | 0,577 | 0,450 | 0,435 |

*Tabla 63: Clasificación para Atributos Seleccionados con frontera de un punto.*

El diagrama resultante para este Dataset sería el siguiente:

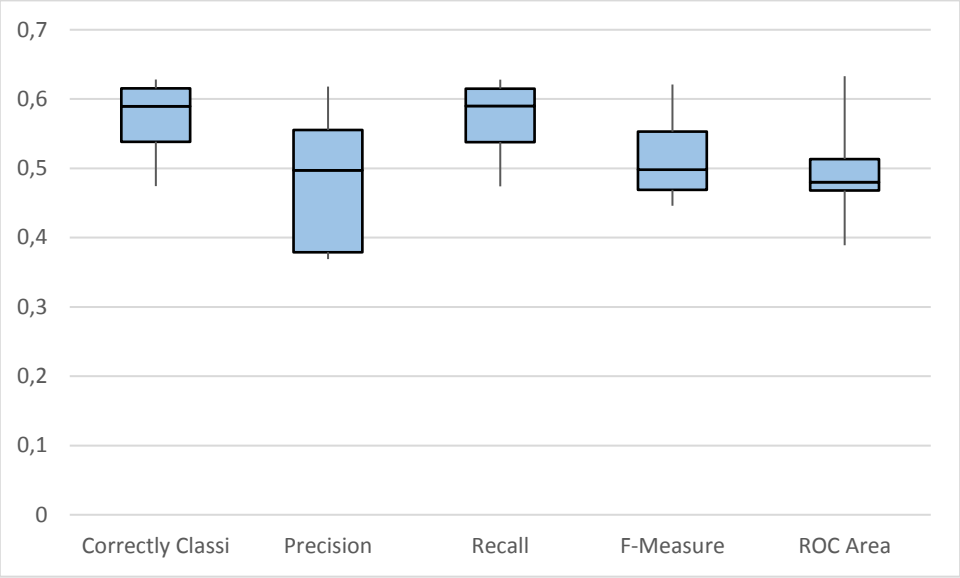



*Ilustración 59: Diagrama para Atributos Seleccionado en clasificación con frontera de un punto.*

Como ya nos viene pasando en todos los anteriores Dataset que tenemos con frontera de un punto la variable que mejor se podría seleccionar seria *"ROC Area"* y por lo tanto los algoritmos que podríamos elegir serían lo siguiente, "Logistic" con un área de 0.633 y "MultiClassClassifier" con un área de 0.633, ahora solo se describirá lo que hace estos algoritmos, pero solo de los que todavía no se ha hablado.

- **MultiClassClassifier:** [35] Es un algoritmo para problemas multiclase este intenta clasificar instancias en una las tres o más clases que tendría.

- **Mejores Correlados:**

| | Correctly Classified Instances | Precision | Recall | F-Measure | ROC Area |
|---|---|---|---|---|---|
| **BayesNet** | 61,56% | 0,379 | 0,615 | 0,469 | 0,480 |
| **NaiveBayes** | 57,69% | 0,574 | 0,577 | 0,576 | 0,454 |
| **NaiveBayesSimple** | 53,85% | 0,538 | 0,538 | 0,538 | 0,435 |
| **Logistic** | 65,38% | 0,652 | 0,654 | 0,653 | 0,640 |
| **MultilayerPerception** | 62,82% | 0,618 | 0,628 | 0,621 | 0,591 |
| **RBFClassifier** | 58,97% | 0,555 | 0,590 | 0,553 | 0,446 |
| **RBFNetwork** | 56,41% | 0,542 | 0,564 | 0,547 | 0,534 |
| **SGD** | 57,69% | 0,553 | 0,577 | 0,558 | 0,514 |
| **SGDText** | 61,54% | 0,379 | 0,615 | 0,469 | 0,500 |
| **SimpleLogistic** | 60,26% | 0,574 | 0,603 | 0,571 | 0,584 |
| **SMO** | 60,26% | 0,376 | 0,603 | 0,463 | 0,494 |
| **VotedPerceptron** | 61,54% | 0,379 | 0,615 | 0,469 | 0,490 |
| **IB1** | 56,41% | 0,564 | 0,564 | 0,564 | 0,524 |
| **IBK** | 56,41% | 0,564 | 0,564 | 0,564 | 0,532 |
| **Kstar** | 56,41% | 0,548 | 0,564 | 0,553 | 0,537 |
| **LWL** | 52,56% | 0,399 | 0,526 | 0,440 | 0,460 |
| **AdaBoostM1** | 51,28% | 0,438 | 0,513 | 0,459 | 0,451 |
| **AtributeSelectedClassifier** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **Bagging** | 58,97% | 0,555 | 0,590 | 0,553 | 0,455 |
| **ClassificationViaRegression** | 61,54% | 0,603 | 0,615 | 0,605 | 0,556 |
| **CVParameterSelection** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **IterativeClassifierOptimizer** | 43,59% | 0,515 | 0,436 | 0,410 | 0,479 |
| **LogitBoost** | 51,28% | 0,485 | 0,513 | 0,494 | 0,494 |
| **MultiBoostAB** | 52,56% | 0,495 | 0,526 | 0,504 | 0,399 |
| **MultiClassClassifier** | 65,38% | 0,652 | 0,654 | 0,653 | 0,630 |
| **MultiScheme** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **RandomCommittee** | 53,85% | 0,533 | 0,588 | 0,535 | 0,477 |
| **RandomSubSpace** | 60,26% | 0,376 | 0,603 | 0,463 | 0,522 |
| **Stacking** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **Vote** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **InputMappedClassifier** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **DecisionTable** | 61,54% | 0,379 | 0,615 | 0,469 | 0,490 |
| **Jrip** | 58,97% | 0,534 | 0,590 | 0,523 | 0,499 |
| **OneR** | 44,87% | 0,394 | 0,449 | 0,415 | 0,411 |
| **Part** | 48,72% | 0,465 | 0,487 | 0,474 | 0,428 |
| **ZeroR** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **DecisionStump** | 61,54% | 0,379 | 0,615 | 0,469 | 0,458 |
| **HoeffdingTree** | 61,54% | 0,379 | 0,615 | 0,469 | 0,480 |
| **J48** | 55,13% | 0,548 | 0,551 | 0,550 | 0,542 |
| **LMT** | 62,82% | 0,609 | 0,628 | 0,606 | 0,601 |
| **RandomForest** | 51,28% | 0,476 | 0,513 | 0,487 | 0,408 |

| | | | | | |
|---|---|---|---|---|---|
| **RandomTree** | 50,00% | 0,490 | 0,500 | 0,494 | 0,470 |
| **REPTree** | 56,41% | 0,486 | 0,564 | 0,493 | 0,428 |

*Tabla 64: Clasificación para Mejores Correlados con frontera de un punto.*

El resultado de construir un diagrama box plot con los datos de la tabla 64, es el que podemos observar a continuación:

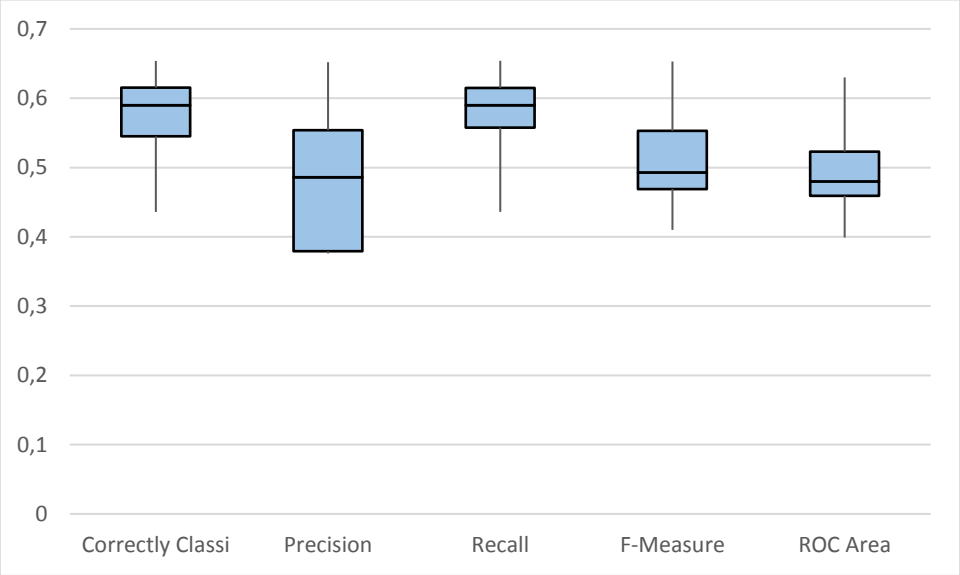



*Ilustración 60: Diagrama para Mejores Correlados en clasificación para frontera de un punto.*

En este diagrama tenemos un poco de variación con los anteriores ya que no esta tan claro la elección de que variable elegir, como pasa en casos anteriores la variable *"Precision"* no se puede seleccionar por la gran dispersión que tiene, pero ahora vemos que las variables *"Recall" y "ROC Area"* son muy parecidas, por lo tanto, se va a tomar las dos variables y se va a elegir un clasificador el cual tenga las dos variables adecuadas y compensadas.

Entonces los algoritmos que hemos decidido escoger serían "Logistic" con un valor de la variable *"Recall"* de 0.654 y un valor de *"ROC Area"* de 0.64 también podemos seleccionar el clasificador "MultiScheme", este tiene un valor de *"Recall"* de 0.654 y de *"ROC Area"* de 0.63.

Estos clasificadores no se van a definir porque ya se ha realizado en puntos anterior.

# 8. Análisis de los Dataset

En este punto lo que vamos a realizar es intentar decidir que Dataset entre los diferentes que tenemos (*TodosAtributos, SoloProfe, SoloAlumno, AtributosSelecionados o MejoresCorrelados)* es el mejor para el caso de regresión y para el caso de clasificación.

La manera de hacer esta elección lo que vamos a hacer es lo mismo que hicimos en el punto 7 para los algoritmos, esto es la construcción de unos diferentes diagramas de cajas o también llamado "box plot".

Entonces lo que vamos a hacer es realizar un diagrama por Dataset y para cada variable, luego vamos a decidir que Dataset es más representativo para una variable, esto teniendo en cuenta por ejemplo que tenga la mínima dispersión y entonces vemos cual es el conjunto de datos que más variables representa y ese sería el mejor para regresión o clasificación para nuestros datos.

## 8.1. REGRESIÓN:

Primero vamos a empezar con la parte de regresión, entonces como ya se ha comentado ahora se va a proceder a poner una serie de diagramas uno por cada variable que hemos sacada por cada algoritmo, en el caso de regresión tendríamos entonces cinco diagramas para las variables *CorrelationCoefficient, MeanAbsoluteError, Root Mean Squared Error, Relative Absolute Error, Root Relative Squared Error* entonces vamos a empezar con el análisis:

- **Correlation Coefficient:**

Primero vamos a empezar con la variable *"CorrelationCoefficient"* entonces hemos generado el diagrama box plot de esta variable para los diferentes conjuntos de datos y nos ha salido lo que podemos ver en la siguiente ilustración:

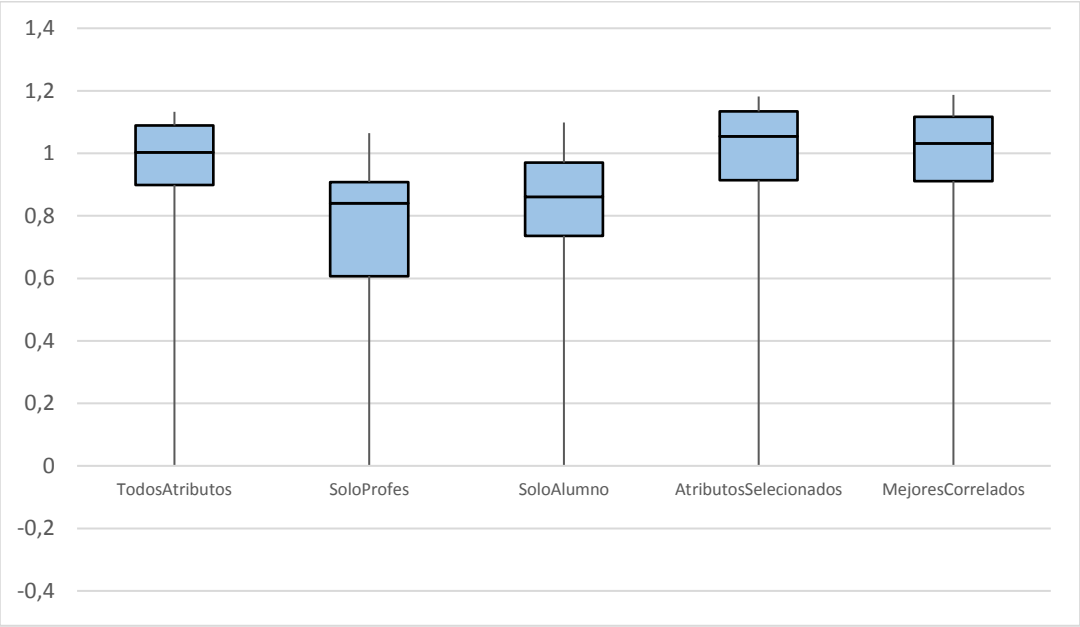



*Ilustración 61:Diagrama de la variable* CorrelationCoefficient *para regresión.*

Podemos ver de forma simple que todos los Dataset tienen un valor mínimo con mucha diferencia del valor medio que tiene cada uno se podría pensar que esto es porque tenemos algún valor de ruido en cada uno de los conjuntos de datos, también podemos observar que el Dataset "SoloProfes" es el que más dispersión tiene con respecto a los demás, y los que menos tiene podrían ser "TodosAtributos" y "MejoresCorrelados", la diferencia con "SoloAlumnos" y "AtributosSelecionado" es muy pequeña por eso se ha tenido que mirar los valores exactos de los cuartiles para ver cual tenía más dispersión quedándonos con los dos que ya hemos dicho, "TodosAtributos" y "MejoresCorrelados".

- **Mean Absolute Error:**

La siguiente variable que vamos a tratar va a ser "Mean Absolute Error" y en la ilustración que tenemos a continuación podemos ver la variación de esta en los diferentes conjuntos de datos que tenemos.

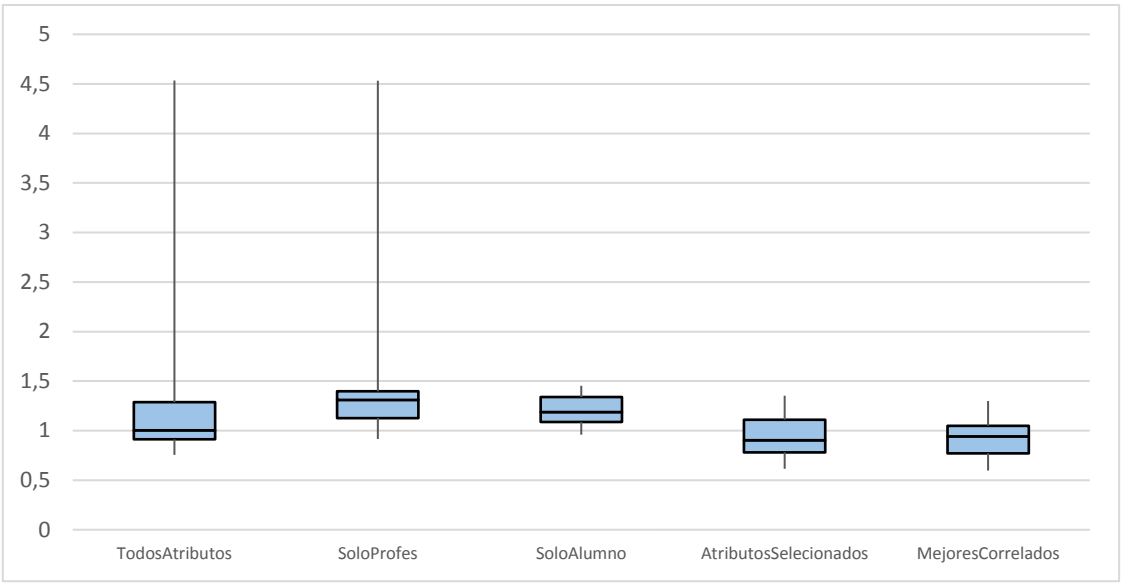



*Ilustración 62: Diagrama de la variable Mean Absolute Error para regresión.*

Vamos a empezar hablado sobre la dispersión en este diagrama, podemos ver que tenemos dos Dataset las cuales tienen más dispersión que las demás y entre ellas no tiene una diferencia muy grande, estas son "TodosAtributos" y "AtributosSelecionados" por lo tanto estas dos variables van a ser eliminados de nuestra elección, de las que nos queda en lo que es tema dispersión podemos ver que son muy parecidas entonces vamos a mirar otras características, la siguiente va a ser máximo y mínimos y aquí podemos decir que el conjunto de datos "SoloProfes" tiene un máximo muy alejado de lo que viene ser la media de valores de los datos y esto nos pude dar a entender que es porque tenemos algún datos que no es válido o que lo podemos tratar como ruido, por lo tanto este conjunto de datos también lo vamos a descartar.

Entonces de los dos que nos queda en tema dispersión y máximo y mínimo están casi igual, por lo tanto, tenemos dos opciones o elegimos las dos o nos figamos en que "SoloAlumnos" tiene más desequilibrio que "MejoresCorrelados" y vamos a elegir esto último y entonces por lo tanto nos quedamos con "MejoresCorrelados".

- **Root Mean Squared Error:**

La siguiente variable que vamos a analizar va a ser "Root Mean Squared Error" y el resultado que nos da a intentar hacer un diagrama box plot es el que se puede ver a continuación:

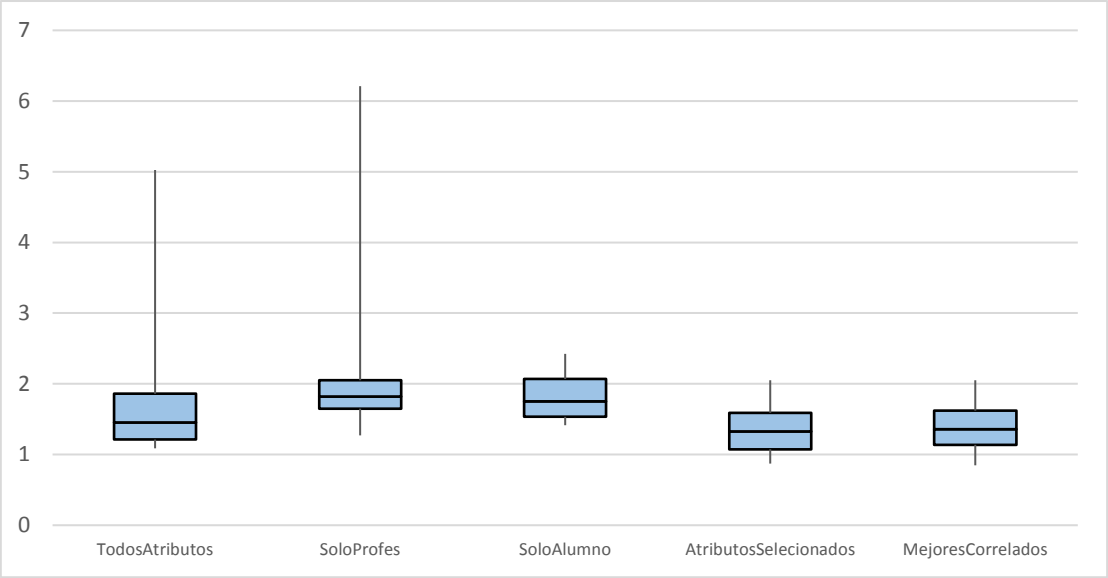



*Ilustración 63: Diagrama de la variable Root Mean Squared Error para regresión.*

Para este caso vamos a empezar mirando los máximos y mínimos de cada uno, como vemos en los Dataset "TodosAtributos" y en "SoloProfes" tenemos un máximo demasiado alegado de lo que son los valores comunes que tenemos en ese conjunto de datos este puede ser por que como ya se ha dicho en otras ocasiones tenemos algún valor malo que nos fastidia los datos, de los otros tres Dataset podemos ver que las diferencias son mínimas pero aunque sea muy poco el conjunto de datos "SoloAlumno" tiene más desequilibrio que los otros dos por lo tanto nos quedaríamos con "AtributosSelecionados" y "MejoresCorrelados" ya que entre estos dos las diferencias son tan mínimos que es imposible decidir entre cual quedarnos.

- **Relative Absolute Error:**

La siguiente variable para tratar sería "Relative Absolute Error" y el diagrama resultante para este en los diferentes conjuntos de datos sería el que podemos ver a continuación:

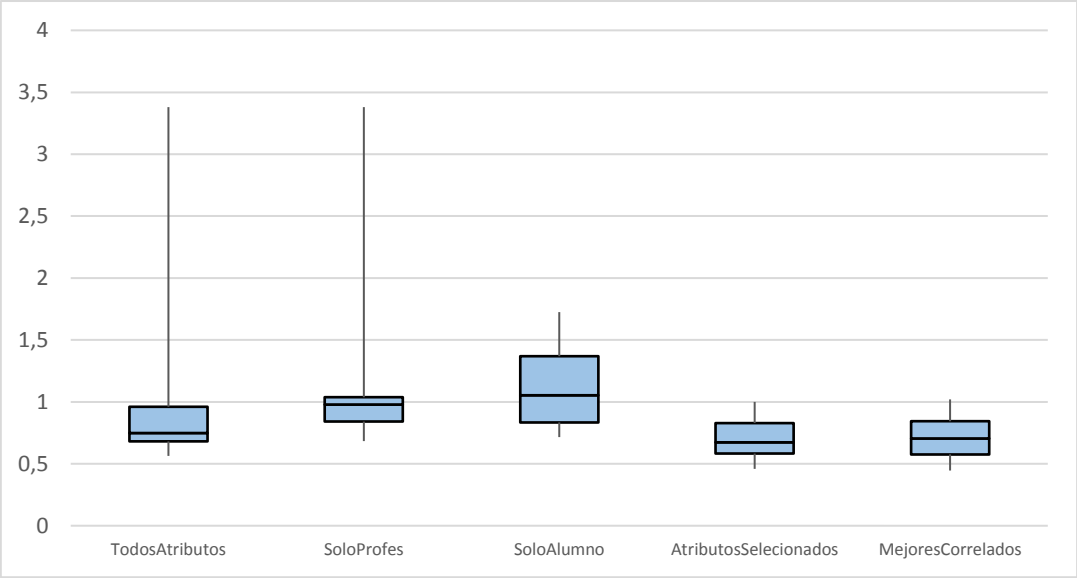



*Ilustración 64: Diagrama de la variable Relative Absolute Error para regresión.*

Como ya ha sucedido en casos anteriores podemos ver que para los Dataset "TodosAtributos" y "SoloProfes" tenemos un valor máximo demasiado alegado de la media y por esta razón no pueden ser elegidos y por qué ya ha sido explicada en varias ocasiones, luego podemos observar que "SoloAlumno" tiene un dispersión muy exagerada en comparación con los otros dos que nos quedan y las diferencias entre estos dos son mínimos por lo tanto nos quedamos con las dos y estas son "AtributoSelecionados" y "MejoresCorrelados".

- **Root Relative Squared Error:**

La siguiente variable para tratar sería "Root Relative Squared Error" y el diagrama resultante para este en los diferentes conjuntos de datos sería el que podemos ver a continuación:

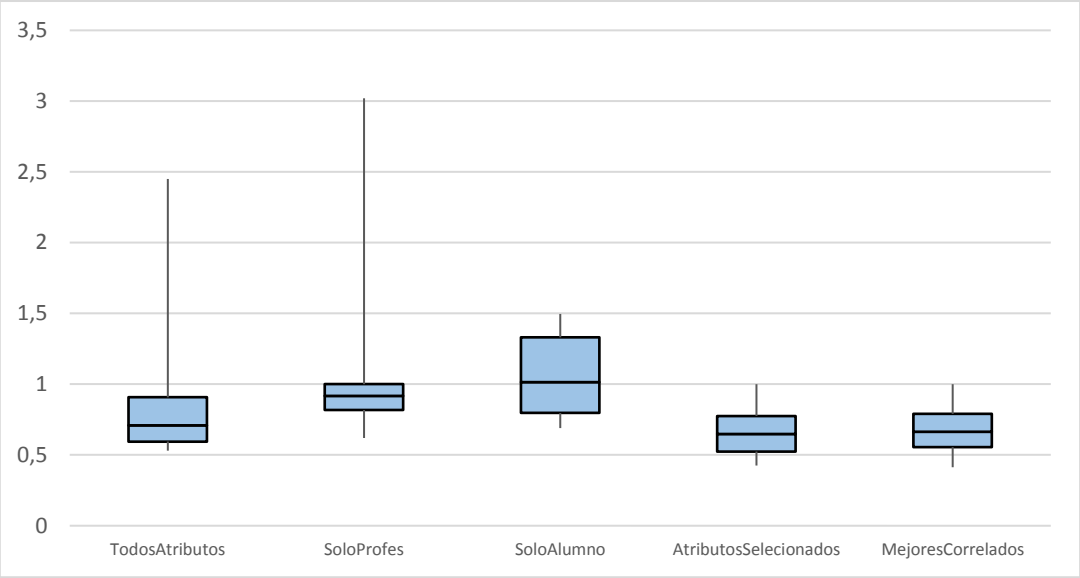



*Ilustración 65: Diagrama de la variable Root Relative Squared Error para regresión.*

En este caso pasa prácticamente lo mismo que en el caso justamente anterior por lo tanto no se va a repetir lo mismo y solo vamos a decir que como en ese caso nos quedamos con “AtributoSelecionados” y “MejoresCorrelados”.

Entonces ahora que hemos analizados los diferentes diagramas hemos visto que en cada uno nos hemos quedado con alguno de los diferentes Dataset, en la tabla 65 podemos ver el número de veces que hemos elegido cada conjunto de datos.

| Dataset | Veces elegido |
|---|---|
| **TodoAtributos** | 1 |
| **SoloProfe** | 0 |
| **SoloAlumno** | 0 |
| **AtributosSelecionados** | 3 |
| **MejoresCorrelados** | 5 |

*Tabla 65: Resultados para regresión.*

Por lo tanto, podemos concluir que para la parte de regresión el mejor Dataset que podríamos utilizar seria **“MejoresCorrelados”.**

Todo esto también se puede observar si miramos el valor promedio de cada Dataset para cada una de las diferentes variables, esto lo podemos ver en la tabla 66, se ve que para el conjunto de datos “MejoresCorrelados” tiene los mejores valores.

| | Correlation Coef | Mean Absolute error | Root mean squared error | Relative Absolute error | Root relative squared error |
|---|---|---|---|---|---|
| **Todos Atributos** | 0.566 | 1.180 | 1.64 | 87.29% | 80.08% |
| **Solo Profe** | 0.429 | 1.402 | 2.089 | 104% | 102% |
| **Solo Alumno** | 0.414 | 1.206 | 1.778 | 89.91% | 86.68% |
| **Atributos Seleccionados** | 0.609 | 0.973 | 1.415 | 72.65% | 69.01% |
| **Mejores Correlados** | 0.616 | 0.954 | 1.384 | 71.19% | 67.49% |

*Tabla 66: Promedios de regresión.*

## 8.2. CLASIFICACIÓN:

Ahora vamos a realizar la misma investigación que hemos hecho para la parte de regresión, pero para clasificación, en esta parte tenemos que ver que se divide en dos, para una frontera de un punto y para una frontera de medio punto.

Para este caso tendríamos 5 diagramas uno por cada variable las cuales son, *Correctly Classified Instances, Precision, Recall, F-Measure, ROC Area.*

- **Frontera de un punto:**

Entonces vamos a empezar con una frontera de un punto es decir nos creemos la nota si la diferencia entre la calificación dada y la sacada es de un punto o menos.

  - **Correctly Classified Instances:**

La variable que vamos a tratar ahora sería “Correctly Classifies Instances” y el diagrama resultante para esta variable en los diferentes conjuntos de datos sería el que podemos ver a continuación:

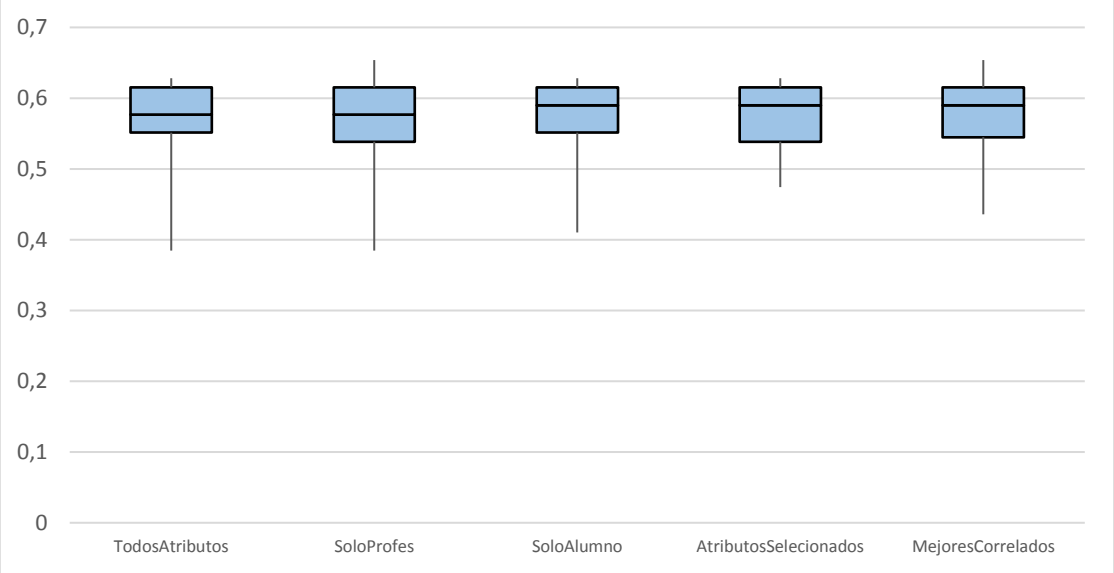



*Ilustración 66: Diagrama de la variable Correctly Classifies Instances para clasificación con frontera de un punto.*

Podemos ver que la diferencia en dispersión es muy parecida en cada conjunto de datos por lo tanto se hace difícil el decidir con cual nos podríamos quedar, entonces para tomar la decisión hemos mirado los valores exactos que nos ha dado para cada cuartil con el que se forma estos diagramas, con esto llegamos a decidir que los conjuntos de datos que vamos a seleccionar con los siguientes “TodosAtributos” y “SoloAlumno” porque hemos observado que son los que tiene menos dispersión de los cinco.

- **Precision:**

La variable que vamos a tratar ahora sería “Precision” y el diagrama resultante para esta variable en los diferentes conjuntos de datos sería el que podemos ver a continuación:

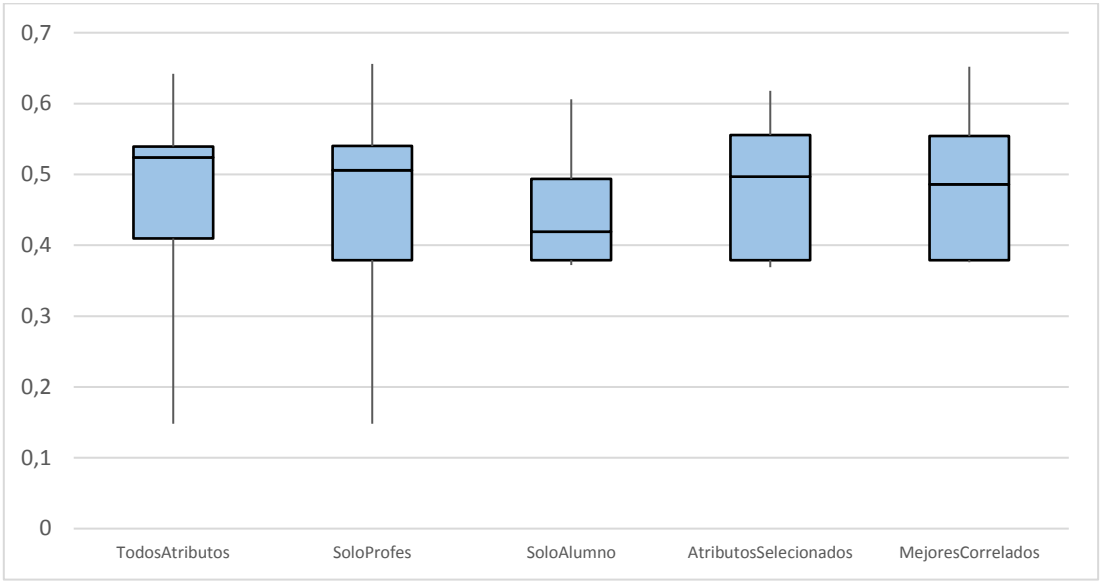



*Ilustración 67:Diagrama de la variable Precision para clasificación con frontera de un punto.*

En este caso se puede tomar una decisión rápidamente ya que está claro la variable que tiene menos dispersión y mejores mínimo y máximo, entonces el conjunto de datos que vamos a elegir para este caso es “SoloAlumno”.

- **Recall:**

La variable que vamos a tratar ahora sería “Recall” y el diagrama resultante para esta variable en los diferentes conjuntos de datos sería el que podemos ver a continuación:

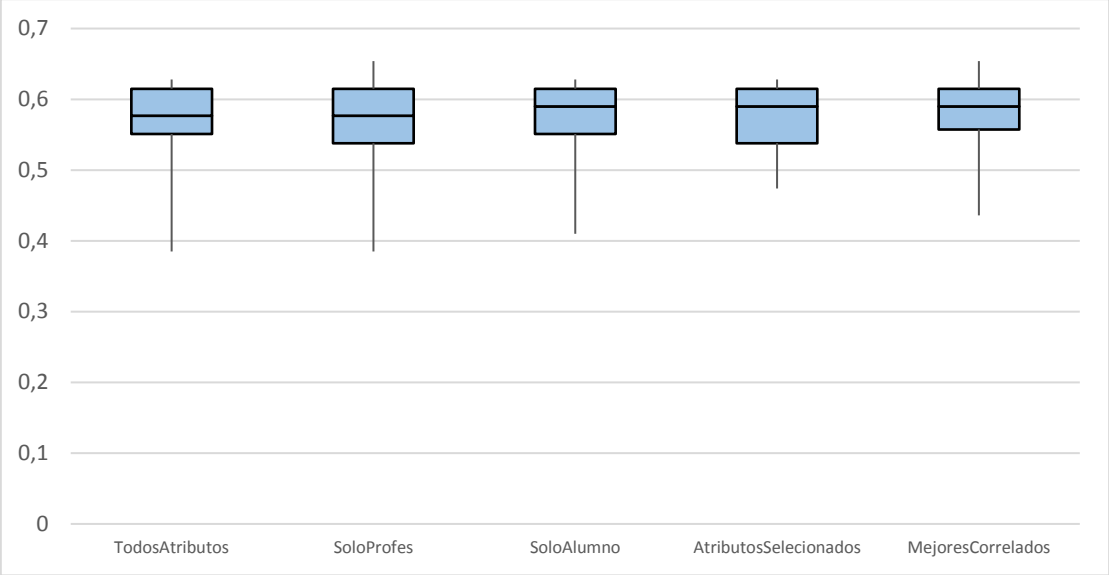



*Ilustración 68:Diagrama de la variable Recall para clasificación con frontera de un punto.*

En este caso podemos tener el mismo problema que con la variable “Correctly Classified Instances”, esto es que las diferencias son tan mínimos que cuesta verlas y por lo tanto hemos tenido que ver los valores exactos como en ese caso y entonces por dispersión nos quedaríamos con “TodosAtributos”, “SoloAlumno” y “MejoresCorrelados” y para afinar un poco más podríamos ver los máximos y mínimos de cada uno de estos y con esto entonces solo nos quedaríamos con “AtributosSelecionados” que son el que tiene mejores máximos y mínimos.

- **F-Measure:**

La variable que vamos a tratar ahora sería “F-Measure” y el diagrama resultante para esta variable en los diferentes conjuntos de datos sería el que podemos ver a continuación:

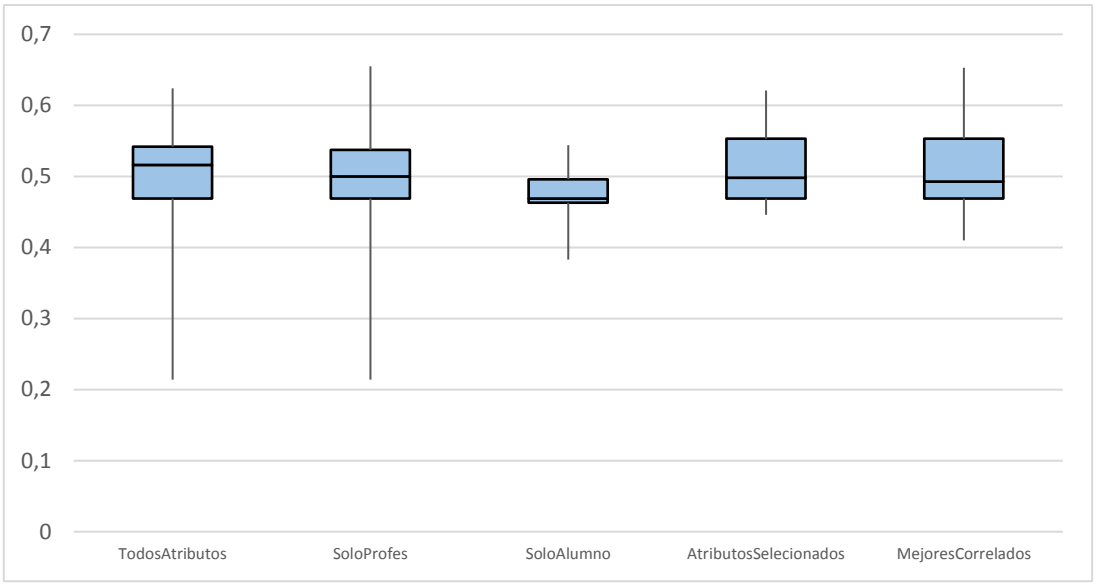



*Ilustración 69:Diagrama de la variable F-Measure para clasificación con frontera de un punto.*

En este caso tenemos más facilidad para elegir que Dataset tenemos que elegir y esto es porque podemos ver que el conjunto de datos "SoloAlumno" es el que menos dispersión tiene y por eso nos quedamos con este.

- **ROC Area:**

La variable que vamos a tratar ahora sería "ROC Area" y el diagrama resultante para esta variable en los diferentes conjuntos de datos sería el que podemos ver a continuación:

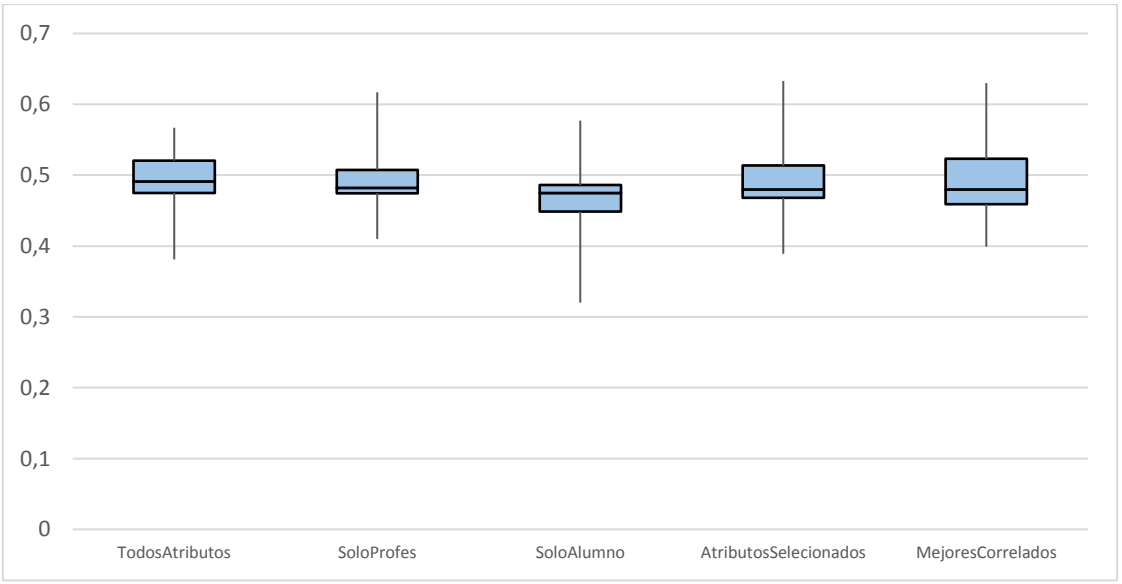



*Ilustración 70:Diagrama de la variable ROC Area para clasificación con frontera de un punto.*

Para este caso tendríamos que discutir entre dos Dataset para ver cual nos quedamos ya que son los que menos dispersión tiene y estos dos son “SoloProfes” y “SoloAlumno” y entre estos dos nos quedaríamos con “SoloProfes” la razón sería porque su mínimo y máximo está muy cerca de los valores medios que tiene a diferencia de “SoloAlumno”.

Entonces ya que hemos terminado de realizar este análisis podemos ver cuántas veces se ha elegido cada Dataset en la tabla 67.

| Dataset | Veces elegido |
|---|---|
| **TodoAtributos** | 1 |
| **SoloProfe** | 1 |
| **SoloAlumno** | 3 |
| **AtributosSelecionado** | 1 |
| **MejoresCorrelados** | 0 |

*Tabla 67: Resultado clasificación para frontera de un punto.*

Entonces podemos decir que para el caso de una frontera de un punto en clasificación el mejor conjunto de datos con el que nos podemos quedar sería “SoloAlumno”.

Todo esto también se puede observar si miramos el valor promedio de cada Dataset para cada una de las diferentes variables, esto lo podemos ver en la tabla 68, se ve que para el conjunto de datos “SoloAlumno” están varios de los mejores valores.

| | Correcty Class | Precision | Recall | F-Mesure | ROC Area |
|---|---|---|---|---|---|
| **Todos Atributos** | 57.30% | 0.4384 | 0.571 | 0.477 | 0.466 |
| **Solo Profe** | 56.61% | 0.481 | 0.565 | 0.499 | 0.493 |
| **Solo Alumno** | 57.11% | 0.491 | 0.570 | 0.508 | 0.491 |
| **Atributos Seleccionados** | 57.22% | 0.486 | 0.571 | 0.513 | 0.491 |
| **Mejores Correlados** | 57.72% | 0.479 | 0.496 | 0.512 | 0.494 |

*Tabla 68:Promedios de clasificación con frontera de un punto.*

- **Frontera de medio punto:**

Entonces ahora vamos a empezar con una frontera de medio punto es decir nos creemos la nota si la diferencia entre la dada y la sacada es de medio punto o menos.

  - **Correctly Classified Instances:**

La variable que vamos a tratar ahora sería “Correctly Classifies Instances” y el diagrama resultante para esta variable en los diferentes conjuntos de datos sería el que podemos ver a continuación:

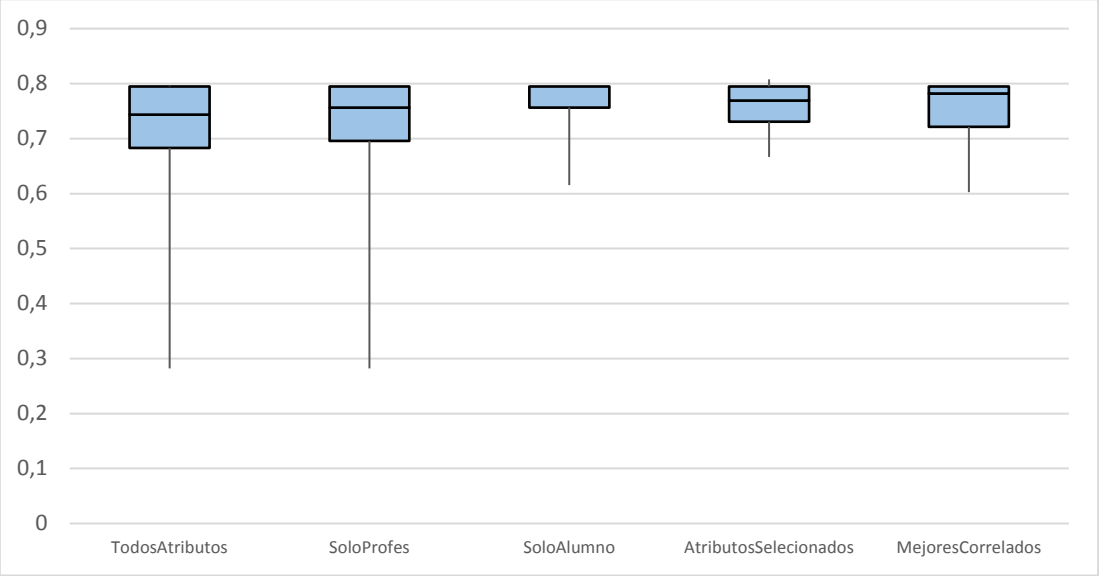



*Ilustración 71:Diagrama de la variable Correctly Classifies Instances para clasificación con frontera de medio punto.*

Para empezar, vamos a decir que los Dataset “TodosAtributos” y “SoloProfes” van a ser eliminados por la razón de tener un mínimo muy alegado de los valores medios y por lo tanto puede llegar a ser ruido. Entonces de los conjuntos que nos quedan nos podríamos quedar por ejemplo con “SoloAlumnos” ya que es el que menos dispersión tiene, pero este tiene un problema y es que tiene un desequilibrio muy grande en comparación con los demás, el siguiente con menos dispersión y que apenas tiene desequilibrio sería “AtributosSelecionados”, por lo tanto, para que tengamos un poco de equilibrio entonces vamos a elegir los Dataset “SoloAlumnos” y “AtributosSelecionados”.

  - **Precision:**

La variable que vamos a tratar ahora sería “Precision” y el diagrama resultante para esta variable en los diferentes conjuntos de datos sería el que podemos ver a continuación:

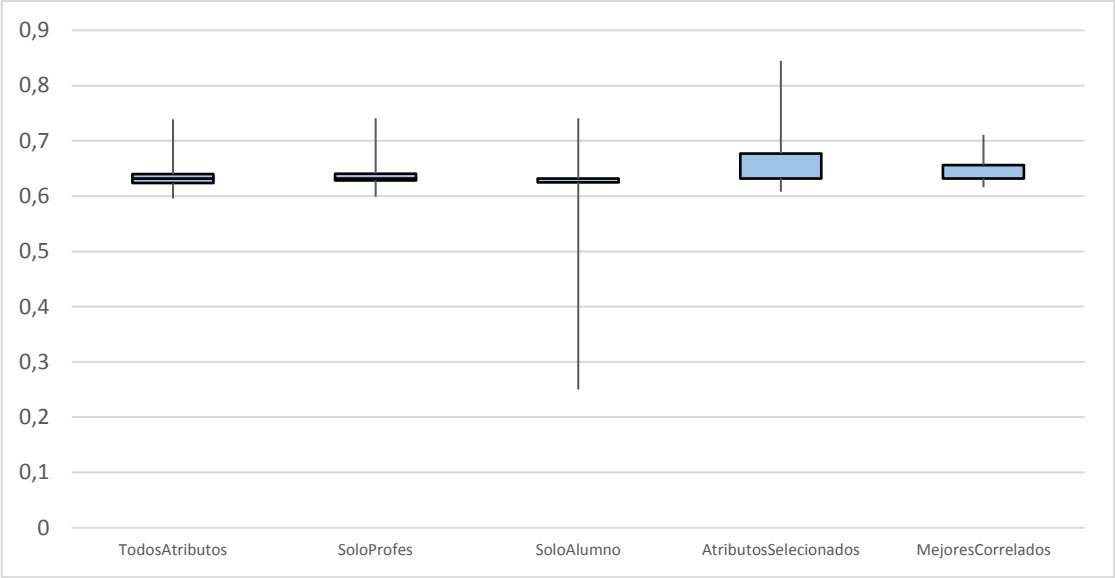



*Ilustración 72:Diagrama de la variable Precision para clasificación con frontera de medio punto.*

En este caso tenemos varias cajas o boxes con una dispersión muy parecida por lo tanto va a ser difícil tomar una elección, primero vamos a eliminar los Dataset "AtributoSelecionados" y "MejoresCorrelados" por tener un poco más de dispersión que los demás, de los que nos queda vamos a eliminar "SoloAlumno" por la razón de que tiene un mínimo muy alegado de la media, de los dos que nos queda tenemos la situación de antes "SoloProfes" tiene menos dispersión pero más desequilibrio que "TodosAtributos" por lo tanto vamos a hacer lo mismo que antes para tener un equilibrio vamos a elegir los dos.

- **Recall:**

La variable que vamos a tratar ahora sería "Recall" y el diagrama resultante para esta variable en los diferentes conjuntos de datos sería el que podemos ver a continuación:

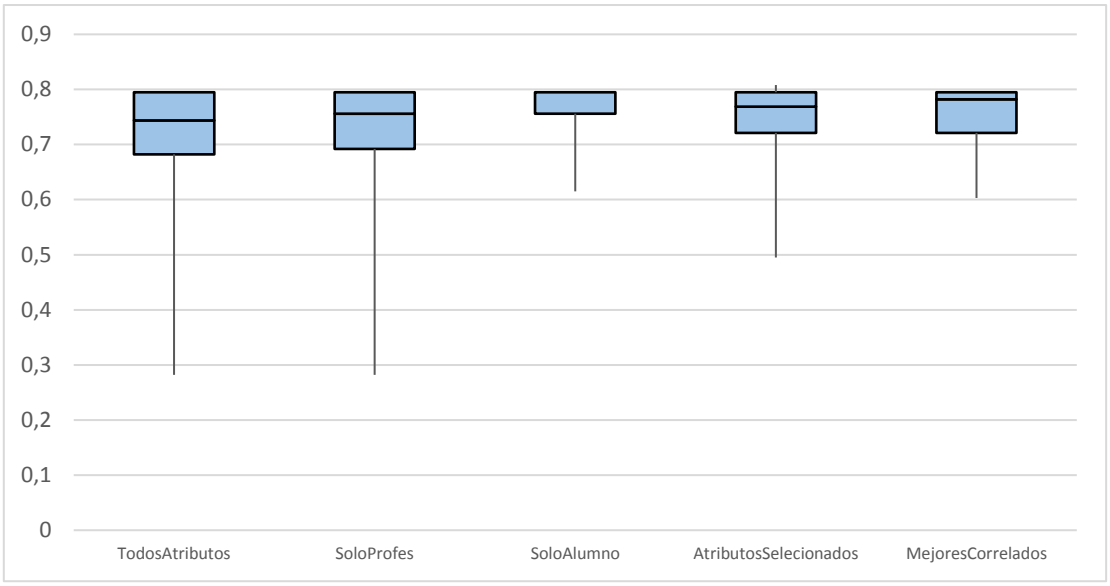



*Ilustración 73:Diagrama de la variable Recall para clasificación con frontera de medio punto.*

En este caso se ve rápido que el que menos dispersión tiene es “SoloAlumnos” y aunque tiene desequilibrio es demasiada la diferencia en dispersión con los demás Dataset por lo tanto nos vamos a quedar con el nombrado anteriormente.

- **F-Measure:**

La variable que vamos a tratar ahora sería “F-Measure” y el diagrama resultante para esta variable en los diferentes conjuntos de datos sería el que podemos ver a continuación:

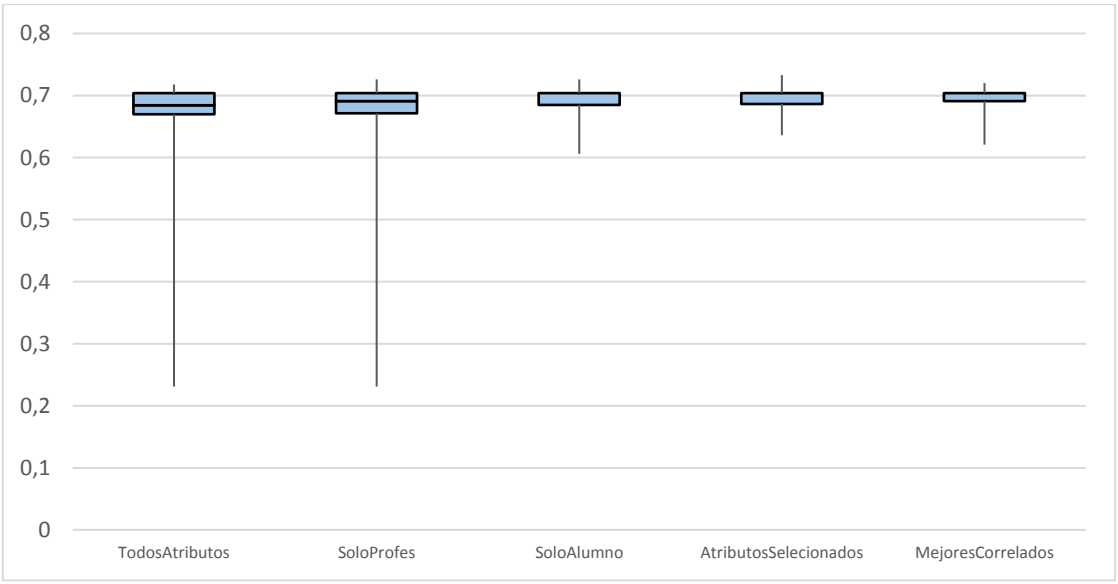



*Ilustración 74:Diagrama de la variable F-Measure para clasificación con frontera de medio punto.*

Para empezar, vamos a eliminar los conjuntos de datos “TodosAtributos” y “SoloProfes” por la razón de tener un mínimo tan alegado de la media, de los tres que nos queda tienen en común el desequilibrio que tienen, pero hay una diferencia y esta es que “MejoresCorrelados” tiene menos dispersión por lo tanto vamos a elegir ese Dataset.

- **ROC Area:**

La variable que vamos a tratar ahora sería “ROC Area” y el diagrama resultante para esta variable en los diferentes conjuntos de datos sería el que podemos ver a continuación:

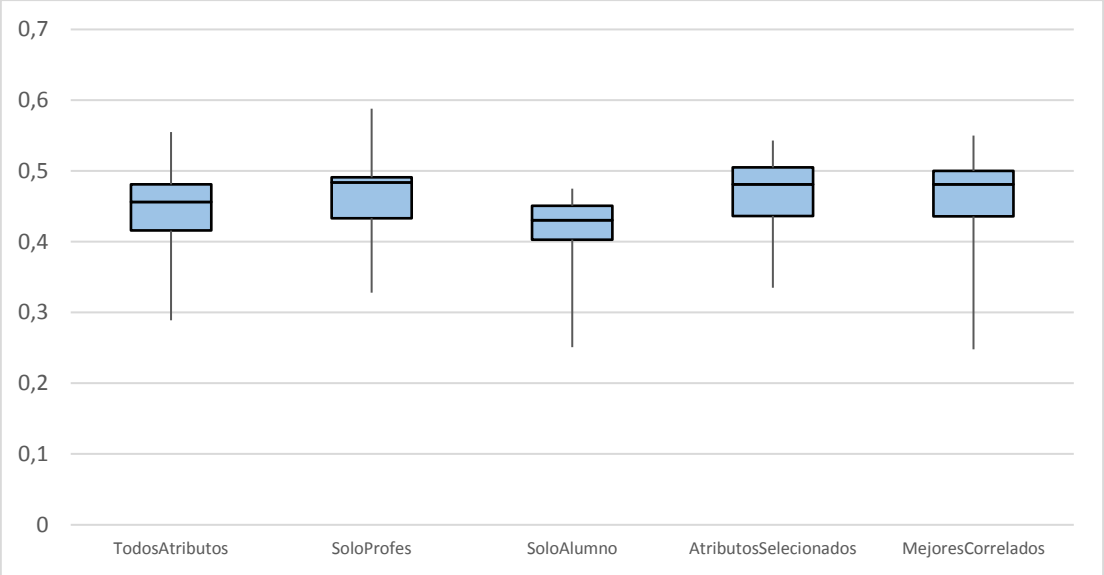



*Ilustración 75:Diagrama de la variable ROC Area para clasificación con frontera de medio punto.*

Como se puede observar todos tienen un mínimo alejado de los valores medios por lo tanto esto no nos sirve, si miramos los valores máximos podríamos entonces eliminar los Dataset “TodosAtributos” y “SoloProfes”, entonces de los tres que nos queda vamos a elegir “SoloAlumnos” por la razón de tener una menor dispersión.

Entonces después de terminar con el análisis de esta parte podríamos ver en la tabla 69 la veces que ha sido elegido cada uno de los conjuntos de datos.

| Dataset | Veces elegido |
|---|---|
| **TodoAtributos** | 1 |
| **SoloProfe** | 1 |
| **SoloAlumno** | 3 |
| **AtributoSelecionados** | 1 |
| **MejoresCorrelados** | 1 |

*Tabla 69: Resultado clasificación para medio punto.*

Entonces podemos decir que para el caso de una frontera de medio punto en clasificación el mejor conjunto de datos con el que nos podemos quedar sería "SoloAlumno".

Todo esto también se puede observar si miramos el valor promedio de cada Dataset para cada una de las diferentes variables, esto lo podemos ver en la tabla 70, se ve que para el conjunto de datos "SoloAlumno" están varios de los mejores valores.

| | Correcty Class | Precision | Recall | F-Mesure | ROC Area |
|---|---|---|---|---|---|
| **Todos Atributos** | 72.62% | 0.640 | 0.752 | 0.670 | 0.444 |
| **Solo Profe** | 72.56% | 0.642 | 0.720 | 0.670 | 0.453 |
| **Solo Alumno** | 76.10% | 0.624 | 0.761 | 0.688 | 0.432 |
| **Atributos Seleccionados** | 75.79% | 0.656 | 0.750 | 0.697 | 0.465 |
| **Mejores Correlados** | 75.92% | 0.646 | 0.759 | 0.695 | 0.458 |

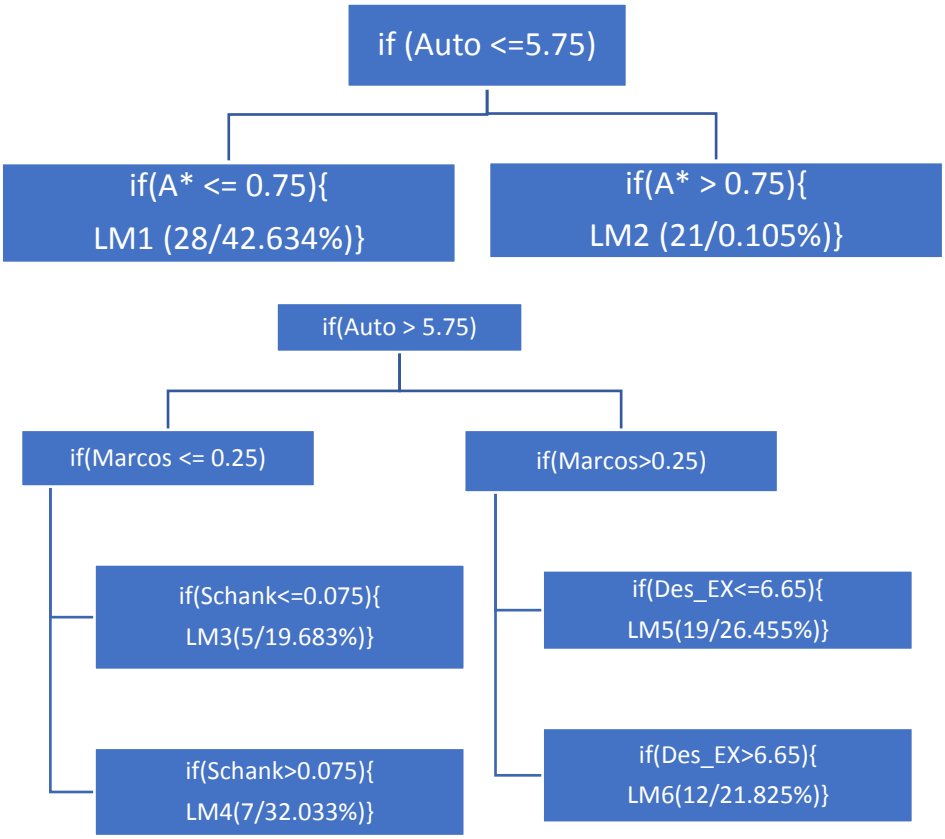
*Tabla 70: Promedios para clasificación con frontera de medio punto.*

# 9. Elección final y resultados:

En apartados anteriores hemos elegido cuales son los mejores algoritmos para cada conjunto de datos y luego que Dataset es mejor para regresión y clasificación, entonces en este apartado lo que vamos a hacer es hablar más detallado sobre los algoritmos de nuestra elección final.

Entonces como se comentó en apartado anteriores para regresión hemos decidido que el mejor Dataset es "MejoresCorrelados" y para este conjunto los algoritmos que hemos decidido son *"M5P" y "M5Rules".*

Entonces vamos a empezar hablando de *"M5P",* como ya comentamos antes este algoritmo lo que crea es una serie de rutina para poder generar un árbol y reglas para obtener el resultado que buscamos. Por lo tanto, cuando hemos aplicado este algoritmo en Weka con nuestro conjunto de datos el árbol que nos ha generado tiene dos ramas una si la variable "Auto" es menos de 5.75 y la otra si esta variable es mayor de 5.75, estas ramas son la siguientes:



Como vemos en este árbol cada rama llega al final a una regla que tiene un porcentaje de instancias que acierta, estas reglas nos darían el resultado de la regresión y estas son la siguientes:

LM1:

$$Nota\_EX = 0.8137 * 3.a.Logica + 1.4548 * Schank + 0.4124 * Marcos + 0.3756 * A * +0.1306 * Bidireccional + 0.1189 * Auto - 0.2602 * Des\_EX + 3.1887$$

LM2:

$$Nota\_EX = 0.4271 * 3.a.Logica + 0.6116 * Schank + 0.4124 * Marcos + 0.405 * A * +0.1306 * Bidireccional + 0.1189 * Auto - 0.0605 * Des\_EX + 3.295$$

LM3:

$$Nota\_EX = 0.2468 * 2.b.Bayer + 0.2697 * 3.a.Logica + 0.2976 * 3.c.Logica + 1.586 * Schank + 1.5066 * Marcos + 1.7422 * A * +0.7095 * Bidireccional + 0.2588 * Auto - 0.7435$$

LM4:

$$Nota\_EX = 0.2468 * 2.b.Bayer + 0.2697 * 3.a.Logica + 0.2976 * 3.c.Logica + 1.5402 * Schank + 1.5066 * Marcos + 2.1166 * A * +0.7095 * Bidireccional + 0.2588 * Auto - 0.5577$$

LM5:

$$Nota\_EX = 0.1449 * 2.b.Bayer + 0.2697 * 3.a.Logica + 0.1747 * 3.c.Logica + 0.7975 * Schank + 0.107 * complejidad + 2.1893 * Marcos + 0.7297 * A * +0.9426 * Bidireccional + 0.2532 * Auto + 0.1082 * Des\_EX + 0.2736$$

LM6:

$$Nota\_EX = 0.1449 * 2.b.Bayer + 0.2697 * 3.a.Logica + 0.1747 * 3.c.Logica + 0.7975 * Schank + 0.3739 * complejidad + 1.3371 * Marcos + 0.7786 * A * +0.476 * Bidireccional + 0.3885 * Auto + 0.1363 * Des\_EX + 0.1605$$

Ahora vamos a empezar a hablar de M5Rules, como ya hemos dicho nos genera una lista de decisiones para problemas de regresión con una metodología de "Separa y Conquista" llegando al final a construir un árbol.

Entonces para nuestro caso se ha generado 6 reglas las cuales son la siguientes:

- Rule 1:

IF
Auto <= 5.75
A* <= 0.75
THEN
Nota_EX = 0.8137 * 3.a.Logica + 1.4548 * Schank + 0.4124 * Marcos + 0.3756 * A* + 0.1306 * Bidireccional + 0.1189 * Auto - 0.2602 * Des_EX + 3.1887

- Rule 2:

IF
Auto <= 5.5
THEN
Nota_EX = 0.4677 * Schank + 0.2748 * complejidad + 0.7008 * Marcos + 0.8538 * A* + 0.3404 * Bidireccional + 0.2786 * Auto + 1.7471

- Rule 3:

IF
Marcos > 0.25
Des_EX <= 6.65
THEN
Nota_EX = 0.1954 * 2.b.Bayer + 0.2356 * 3.c.Logica + 0.5443 * Schank + 0.1443 * complejidad + 2.3391 * Marcos + 0.6504 * A* + 1.0771 * Bidireccional + 0.1646 * Auto + 0.146 * Des_EX + 0.8701

- Rule 4:

IF
Des_EX > 5.745
THEN
Nota_EX = 2.1583 * 3.a.Logica + 1.208 * Schank + 1.4645 * complejidad - 0.5002 * Marcos + 0.6789 * Auto + 0.1923 * Des_EX - 2.9432

- Rule 5:

IF
Schank <= 0.075
THEN
Nota_EX = 2.1736 * Schank + 0.533

- Rule 6:

| Nota_EX = -1.1526 * 2.b.Bayer + 3.0505 |
|---|

Ahora vamos a ponernos con la parte de clasificación en esta parte teníamos dos subdivisiones ya que habíamos hecho el estudio para una frontera de un punto y otra para una frontera de medio punto, después de terminar la investigación nos hemos dado cuenta de que por casualidad para las dos partes el mejor conjunto de datos ha sido el de “SoloAlumnos”.

Si después miramos los algoritmos que habíamos elegido para estos Dataset vemos que tiene uno en común, este es “RandomTree” y este es el algoritmo del cual vamos a hablar en esta parte.

Como ya se comentó este clasificador lo que hace es construir un árbol de forma aleatoria, este considera k atributos elegidos al azar en cada nodo. Ahora se va a mostrar los arboles resultantes de este algoritmo para los diferentes casos.

- **Para frontera de un punto:**

```
Auto < 9.75
|   Azure < 51.5
|   |   Hetero < 8.34
|   |   |   Azure < 49.5
|   |   |   |   Azure < 42
|   |   |   |   |   Des_EX < 3.6 : SI (2/0)
|   |   |   |   |   Des_EX >= 3.6
|   |   |   |   |   |   Hetero < 2.78
|   |   |   |   |   |   |   Auto < 7.5
|   |   |   |   |   |   |   |   Auto < 5.75
|   |   |   |   |   |   |   |   |   Des_EX < 5.5 : NO (2.16/0)
|   |   |   |   |   |   |   |   |   Des_EX >= 5.5 : SI (1/-0)
|   |   |   |   |   |   |   |   Auto >= 5.75 : SI (1.16/-0)
|   |   |   |   |   |   |   Auto >= 7.5 : NO (1/-0)
|   |   |   |   |   |   Hetero >= 2.78
|   |   |   |   |   |   |   Hetero < 6.67 : SI (2/0)
|   |   |   |   |   |   |   Hetero >= 6.67 : NO (0.16/0)
|   |   |   |   Azure >= 42
|   |   |   |   |   Auto < 7.75
|   |   |   |   |   |   Azure < 43.5
|   |   |   |   |   |   |   Auto < 5.75 : NO (1.08/0)
|   |   |   |   |   |   |   Auto >= 5.75 : NO (1.04/0.04)
|   |   |   |   |   |   Azure >= 43.5
|   |   |   |   |   |   |   Des_EX < 3.88 : NO (2.19/0)
|   |   |   |   |   |   |   Des_EX >= 3.88
|   |   |   |   |   |   |   |   Des_EX < 5
|   |   |   |   |   |   |   |   |   Hetero < 5 : SI (2.19/0)
|   |   |   |   |   |   |   |   |   Hetero >= 5
|   |   |   |   |   |   |   |   |   |   Auto < 6 : SI (1.19/0.19)
|   |   |   |   |   |   |   |   |   |   Auto >= 6 : NO (1/0)
|   |   |   |   |   |   |   |   Des_EX >= 5
|   |   |   |   |   |   |   |   |   Hetero < 1.67 : SI (1/0)
|   |   |   |   |   |   |   |   |   Hetero >= 1.67 : NO (3/0)
|   |   |   |   |   Auto >= 7.75 : SI (1/0)
|   |   |   |   Azure >= 49.5
|   |   |   |   |   Hetero < 5.55
|   |   |   |   |   |   Des_EX < 3.88 : NO (0.05/0)
|   |   |   |   |   |   Des_EX >= 3.88 : SI (3.05/0)
|   |   |   |   Hetero >= 5.55 : NO (0.05/0)
|   |   Hetero >= 8.34
|   |   |   Auto < 5.25 : NO (5/0)
|   |   |   Auto >= 5.25 : SI (1/0)
|   Azure >= 51.5
|   |   Hetero < 1.67
|   |   |   Auto < 7.63
|   |   |   |   Auto < 5.63
|   |   |   |   |   Hetero < 1.06
|   |   |   |   |   |   Azure < 55.5
|   |   |   |   |   |   |   Des_EX < 3.88
|   |   |   |   |   |   |   |   Des_EX < 3.63 : NO (1/0)
|   |   |   |   |   |   |   |   Des_EX >= 3.63 : SI (1.22/0.22)
|   |   |   |   |   |   |   Des_EX >= 3.88 : NO (2/0)
|   |   |   |   |   |   Azure >= 55.5
|   |   |   |   |   |   |   Des_EX < 4.63
|   |   |   |   |   |   |   |   Des_EX < 3.5 : SI (1/0)
|   |   |   |   |   |   |   |   Des_EX >= 3.5 : NO (1.22/0)
|   |   |   |   |   |   |   Des_EX >= 4.63 : SI (2/0)
|   |   |   |   |   Hetero >= 1.06 : NO (3/0)
|   |   |   |   Auto >= 5.63 : SI (2.44/0)
|   |   |   Auto >= 7.63 : NO (2/0)
|   |   Hetero >= 1.67
|   |   |   Auto < 8.75
|   |   |   |   Hetero < 8.34 : NO (13.44/0)
|   |   |   |   Hetero >= 8.34
|   |   |   |   |   Hetero < 9.45 : SI (1/0)
|   |   |   |   |   Hetero >= 9.45 : NO (1/0)
|   |   |   Auto >= 8.75 : SI (1/0)
Auto >= 9.75 : NO (5/0)
```

- **Para frontera de medio punto:**

```
Auto < 9.75
|   Des_EX < 5.38
|   |   Hetero < 5
|   |   |   Hetero < 1.67
|   |   |   |   Azure < 52.5 : NO (9.78/0)
|   |   |   |   Azure >= 52.5
|   |   |   |   |   Hetero < 1.06
|   |   |   |   |   |   Azure < 68.5
|   |   |   |   |   |   |   Azure < 59.5
|   |   |   |   |   |   |   |   Azure < 53.5 : SI (1.06/0.06)
|   |   |   |   |   |   |   |   Azure >= 53.5
|   |   |   |   |   |   |   |   |   Des_EX < 3.38 : SI (1/0)
|   |   |   |   |   |   |   |   |   Des_EX >= 3.38 : NO (4.3/0)
|   |   |   |   |   |   |   Azure >= 59.5
|   |   |   |   |   |   |   |   Des_EX < 4.38 : NO (0.12/0)
|   |   |   |   |   |   |   |   Des_EX >= 4.38 : SI (2/0)
|   |   |   |   |   |   Azure >= 68.5 : NO (2.12/0)
|   |   |   |   |   Hetero >= 1.06 : NO (4.61/0)
|   |   |   Hetero >= 1.67
|   |   |   |   Azure < 40 : SI (2/0)
|   |   |   |   Azure >= 40
|   |   |   |   |   Azure < 49.5 : NO (4/0)
|   |   |   |   |   Azure >= 49.5
|   |   |   |   |   |   Azure < 51 : SI (1/0)
|   |   |   |   |   |   Azure >= 51 : NO (2/0)
|   |   Hetero >= 5 : NO (12/0)
|   Des_EX >= 5.38
|   |   Auto < 9.25
|   |   |   Des_EX < 6.53
|   |   |   |   Azure < 58.5
|   |   |   |   |   Hetero < 0.56 : SI (2/0)
|   |   |   |   |   Hetero >= 0.56
|   |   |   |   |   |   Auto < 5.75
|   |   |   |   |   |   |   Hetero < 3.33 : SI (1/0)
|   |   |   |   |   |   |   Hetero >= 3.33
|   |   |   |   |   |   |   |   Auto < 5.25 : NO (2/0)
|   |   |   |   |   |   |   |   Auto >= 5.25 : SI (1/0)
|   |   |   |   |   |   Auto >= 5.75 : NO (5/0)
|   |   |   |   Azure >= 58.5 : SI (2/0)
|   |   |   Des_EX >= 6.53
|   |   |   |   Auto < 8.25 : NO (8/0)
|   |   |   |   Auto >= 8.25
|   |   |   |   |   Azure < 55.5 : SI (2/0)
|   |   |   |   |   Azure >= 55.5 : NO (3/0)
|   |   Auto >= 9.25 : SI (1/0)
Auto >= 9.75 : NO (5/0)
```

Como vemos los árboles que genera son muy grandes, tan grande que se ha tenido que poner en el formato que se puede observar porque de otra manera no entraría o se podría quedar de forma más liosa para el lector, y la razón de este tamaño es porque este algoritmo no realiza ninguna poda en todo su proceso, por esto podemos ver que hay ramas las cuales repite y a veces de seguido el mirar la misma variable pero con diferentes valores.

# Sección II: Herramienta programada

# 10. Introducción para la aplicación.

Mientras que en capítulos anteriores se ha ido realizando una descripción de las características más generales así como la explicación de toda nuestra investigación y punto más importante de este proyecto, en los capítulos que aparecen a continuación se irán explicando cómo va a diseñarse una herramienta la cual ha sido creada a partir de la información obtenida en la investigación anterior.

Vamos a estudiar y planificar como deberá ser implementado el modelo de datos de la aplicación, requisitos a cumplir, cual deberá ser su arquitectura, se diseñará la interfaz de usuario y se describirán los procedimientos para las operaciones.

El diseño de la aplicación se presentará utilizando una metodología UML, describiendo como se han desarrollado los módulos necesarios para la aplicación final. Para ellos se realizará un modelado de los datos, de los procedimientos y de la interfaz.

Entonces el diseño de la aplicación contempla los siguientes aspectos:

- Diseño de datos: Descripción del modelo de datos, conforme a la especificación de requisitos, y las modificaciones que habrá que realizar en dicho modelo para que se ajuste lo más adecuadamente posible a la tecnología elegida para su implementación.

- Diseño de la interfaz: Descripción detallada del aspecto definitivo que presentará la interfaz gráfica de usuario de la aplicación, conforme a la especificación de esta realizada anteriormente.

# 11. Especificaciones de requisitos

Una vez ofrecida una visión del problema vamos a realizar un análisis de requisitos, que representa el segundo escalón en el proceso del proyecto. Concretamente deben identificarse aquí las funciones que queremos que realice nuestro software, el rendimiento esperado y los interfaces necesarios.

Vamos a tratar de dar una visión detallada de los requisitos que ha de cumplir la herramienta que se va a desarrollar. Esta visión se dará desde dos puntos de vista, el funcional y el no funcional.

Un requisito funcional define el comportamiento interno del software, como cálculos, detalles técnicos, etc. y los requisitos no funcionales, en cambio, se enfoca directamente en el diseño o la implementación.

## 11.1. Requisitos funcionales

**RF-1:** La herramienta debe estar programada en java y poder usada en cualquier sistema operativo.

**RF-2:** La herramienta debe ser fácil de descargar, instalar y desinstalar.

**RF-3:** Se podría cargar el archivo de datos en la herramienta.

**RF-4:** La herramienta deberá proporcionar todos los tipos de algoritmos resultantes como mejores en nuestra investigación.

**RF-5:** La herramienta deberá mostrar mensajes de error siempre que se origine alguno.

**RF-6:** La herramienta mostrara un mensaje cada vez que tengamos un resultado.

**RF-7:** La herramienta podría mostrar una ayuda sobre el programa y también mensajes para saber qué hacer.

**RF-8:** La herramienta podría generar un PDF con todos los resultados que hemos obtenido de aplicar algún algoritmo.

**RF-9:** El usuario podrá introducir los datos de un alumno a mano en la herramienta.

## 11.2. Requisitos no funcionales

En cuanto a los requisitos no funcionales, se han definido los siguientes requisitos para la realización de este sistema:

**RNF-1:** El tiempo de respuesta de la aplicación debe ser el más reducido posible, en función del equipo con el que se trabaje, por lo que se desea el máximo aprovechamiento de los recursos de que se dispongan en la máquina en que se ejecuta la aplicación.

**RNF-2:** El proceso de desarrollo se hará siguiendo un paradigma iterativo e incremental.

**RNF-3:** La interfaz debe ser sencilla e intuitiva, debe permitir al usuario realizar análisis y representaciones gráficas de los mismos de manera rápida y eficaz.

**RNF-4:** La herramienta desarrollada estará protegida bajo la licencia GNU Public, englobándose dentro del conjunto de herramientas conocidas como herramientas o aplicaciones de software libre.

# 12. Descripción funcional

Para describir las funciones que se han desarrollado para la aplicación se hará uso de los diagramas de casos de uso y de secuencias que proporciona el lenguaje de modelado UML 2.0.

- **Diagrama de casos de uso:** Identifican la interacción del sistema con el actor o los actores que participan en el desarrollo de la aplicación, son las actividades que ejecuta el sistema.

- **Diagrama de secuencia:** Representa la interacción con el sistema tal y como ocurre en el tiempo. Se forman a partir de los casos de uso.

## 12.1. Diagrama de caso de uso

Los casos de uso se emplean para capturar el comportamiento deseado del sistema en desarrollo, sin tener que especificar como se implementa ese comportamiento. Los casos de uso proporcionan un medio para que los desarrolladores, los usuarios finales del sistema y los expertos del dominio lleguen a un entendimiento común del sistema. Además nos ayuda a validar la arquitectura y a verificar el sistema mientras evoluciona a lo largo del desarrollo.

Un caso de uso representa un requisito funcional del sistema global e involucra la interacción de actores y el sistema u otros sujetos. Un actor representa un conjunto coherente de roles que juegan un papel en estos.

Los diagramas de caso de uso se podrían decir que se utilizan para modelar la vista funcional de un sistema, son importantes para modelar el comportamiento de un sistema y cada uno muestra un conjunto de casos de uso, actores y sus relaciones.

A continuación vamos a definir la plantilla que se utilizara para especificar cada uno de los casos de uso, hay que decir que esta plantilla no es obligatoria se puede modificar según las circunstancias que estemos explicando en cada momento.

| **Nombre del caso de uso** | |
|---|---|
| **Nivel** | Nivel de abstracción del caso de uso. |
| **Actores** | Actores que intervienen en el caso de uso. |
| **Propósito** | Descripción de la funcionalidad. |
| **Contesto de uso** | Circunstancias que se requiere para su desarrollo. |
| **Escenario principal** | Flujo natural de las acciones. |
| **Escenario** | Flujos alternativos. |

*Tabla 71: Plantilla de casos de uso.*

### 12.1.1. Diagrama de caso de uso CU0: Contexto del sistema

En la ilustración 76 Se puede observar el diagrama de casos de uso CU0: contexto del sistema.

El conjunto de casos de uso para definir el contexto del sistema son el siguiente, *cargar fichero y ayuda.*

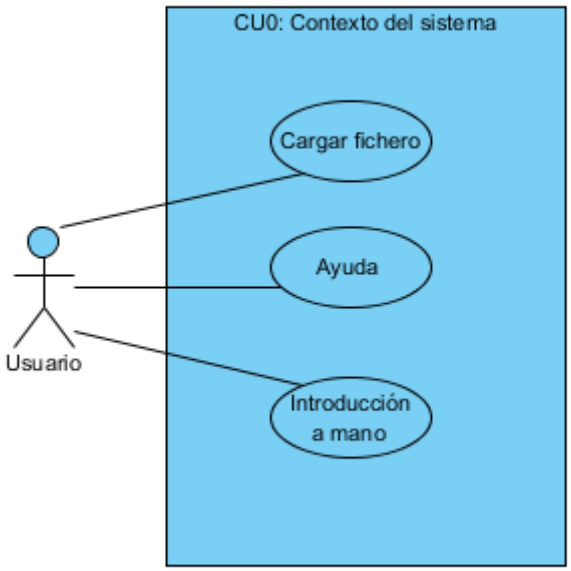



*Ilustración 76: Diagrama de CU0: Contexto del sistema.*

La descripción de este caso de uso la podemos encontrar a continuación:

| **CU0: Contexto del sistema** | |
|---|---|
| **Nivel** | 0 |
| **Actores** | Usuario. |
| **Propósito** | Son los elementos del sistema más principales que lleva el contexto del sistema. |
| **Contesto de uso** | Ejecución del programa. |
| **Casos de uso:** | • **CU1: Cargar fichero:** Se encarga del procesado de los datos y ventana de algoritmos para su utilización.<br>• **CU0.1: Ayuda:** Se encarga de toda la ayuda del sistema, es tan |

| | |
|---|---|
| | insignificante que no le dedicaremos ningún estudio.<br>• **CU4: Introducción a mano:** Se encarga de poder meter los datos que puede tener un alumno a mano por el usuario y ver que solución obtener. |

*Tabla 72: Especificaciones del CU0: Contexto del sistema.*

### 12.1.2. Diagrama de caso de un CU1: Cargar fichero

En la ilustración 77 podemos observar el diagrama del caso de uso CU1.

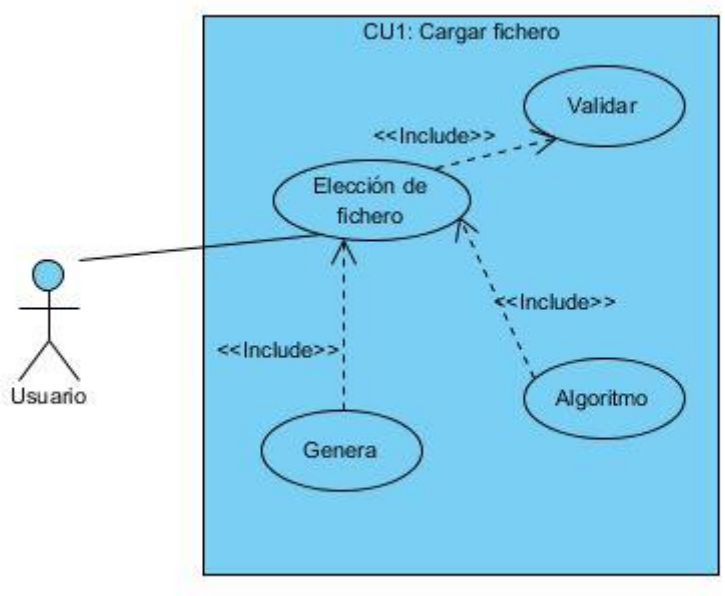



*Ilustración 77: Diagrama de CU1: Cargar fichero.*

La descripción de este caso de uso la podemos ver a continuación:

| **CU1: Cargar fichero** | |
|---|---|
| **Nivel** | 1 |
| **Actores** | Usuario |
| **Propósito** | Descripción de la funcionalidad. |
| **Contesto de uso** | El usuario hace la petición de cargar un fichero. |
| **Escenario principal** | • El usuario hace la petición.<br>• El sistema valida el fichero<br>• El sistema guarda los datos. |
| **Escenario** | Sería igual que el principal pero cuando valida si el fichero no es apto, entonces no se produciría ningún cambio en el programa. |

*Tabla 73: Especificaciones del CU1: Cargar fichero.*

### 12.1.3. Diagrama de caso de un CU2: Algoritmos

En la ilustración 78 podemos observar el CU2: Algoritmos.

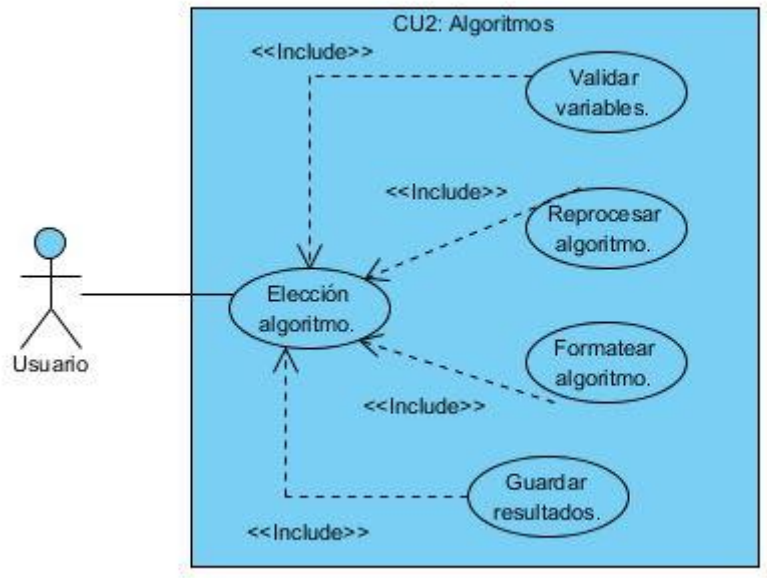



*Ilustración 78: Diagrama de CU2: Algoritmos.*

La descripción de las especificaciones del caso de uso la podemos ver a continuación:

| **CU2: Algoritmos** | |
|---|---|
| **Nivel** | 2 |
| **Actores** | Usuario |
| **Propósito** | El que el usuario pueda aplicar alguno de los diferentes algoritmos. |
| **Contesto de uso** | Las circunstancias serían que el programa tengo cargado un fichero de datos correcto y que el usuario haga la petición de aplicar un algoritmo. |
| **Escenario principal** | • El usuario pide aplicar un algoritmo.<br>• El sistema comprueba si tiene datos guardados.<br>  ○ Si es que si le pregunta al usuario si lo quiere borrar o mantener.<br>  ○ Si es que no entonces no hace nada.<br>• El sistema ejecuta el algoritmo. |
| **Escenario** | Los únicos flujos alternativos que podemos tener en este caso de uso podrían ser por la respuesta del usuario en qué hacer con los datos, si dice borrarlos pasa lo del principal y si dice que no lo borrar entonces no sigue el principal y se acaba sin realizar ningún cambio. |

*Tabla 74: Especificaciones del CU2: Algoritmo.*

### 12.1.4. Diagrama del caso de uso CU3: Generar:

En la ilustración 79 podemos observar el CU3: Generar

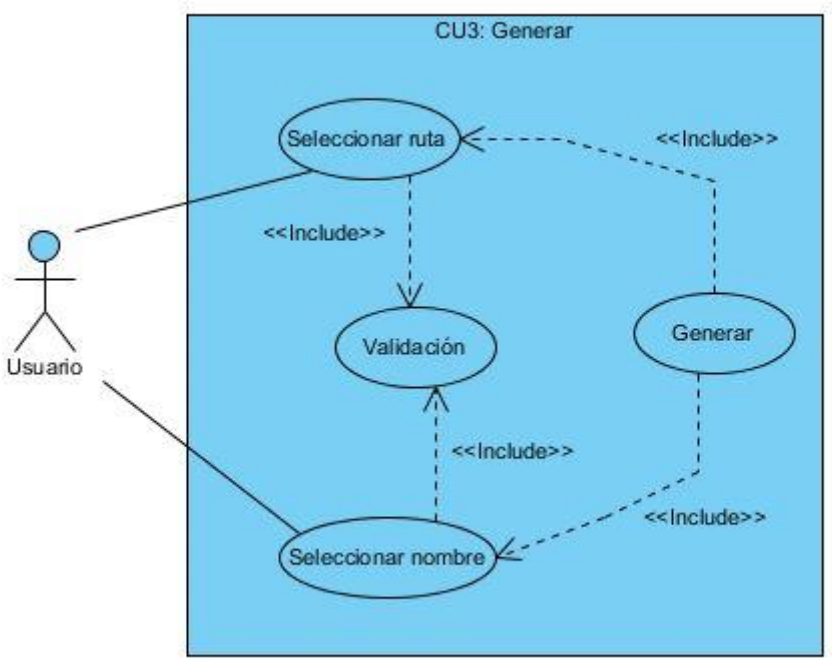



*Ilustración 79: Diagrama del CU3: Generar.*

La descripción de las especificaciones del caso de uso la podemos ver a continuación:

| **CU3: Generar** | |
|---|---|
| **Nivel** | 3 |
| **Actores** | Usuario |
| **Propósito** | El propósito es poder proporcionar al usuario la posibilidad de generar un PDF con los resultados. |
| **Contesto de uso** | Las circunstancias para que se dé este caso de uso serían que tengamos un fichero cargado, tengamos un algoritmo correctamente ejecutado y el usuario haga la petición de generar el PDF. |
| **Escenario principal** | • El usuario pide generar un PDF.<br>• El sistema valida si hay datos guardados.<br>• Si es que si entonces le pide los datos que le sea necesarios para generar el PDF. |

| | |
|---|---|
| | • Genera el PDF y notifica al usuario. |
| **Escenario** | Flujos alternativos podrían ser, que no tengamos datos guardado por lo tanto no se realizaría nada más y se le notifica al usuario. |

*Tabla 75: Especificaciones del CU3: Generar.*

### 12.1.5. Diagrama de caso de un CU4: Introducción a mano

En la ilustración 80 podemos observar el CU4: Introducción a mano.

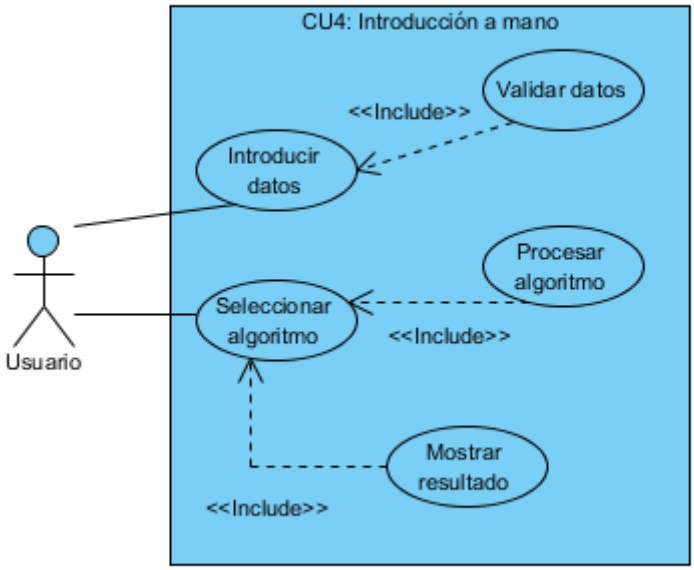



*Ilustración 80: Diagrama CU4: Introducción a mano.*

La descripción de las especificaciones del caso de uso la podemos ver a continuación:

| **CU4: Introducción a manos** | |
|---|---|
| **Nivel** | 4 |
| **Actores** | Usuario |
| **Propósito** | El propósito es poder introducir por el usuario los diferentes valores que puede tener un alumno para ver el resultado en los diferentes algoritmos. |
| **Contesto de uso** | Las circunstancias para que se dé este caso sería solo ejecutar el programa. |
| **Escenario principal** | • El usuario pide introducir los datos y los introduce. |

| | |
|---|---|
| | • El sistema valida si hay datos que no son válidos.<br>• El usuario elige un algoritmo.<br>• El sistema ejecuta el algoritmo y muestra la solución. |
| **Escenario** | Flujos alternativos podrían ser, que algún dato no sea válido que entonces el sistema mostraría un mensaje de error. |

*Tabla 76: Especificaciones CU4: Introducción a manos.*

## 12.2. Diagramas de secuencia

Los diagramas de secuencia describen la colaboración existente entre los objetos que componen el sistema. Se trata de un diagrama de interacción que detalla cómo se lleva a cabo las operaciones, qué mensaje son enviados y cuándo organizándolo todo en torno al tiempo. Para una fácil compresión de este tipo de diagramas, hay que tener en cuenta que la secuencia temporal avanza "hacia abajo" en el diagrama, y que los objetos involucrados en las operaciones se listan de izquierda a derecha, según su orden de participación dentro de la secuencia de paso de mensajes.

En el contexto de los casos de uso, un diagrama de secuencia permite representar un escenario, que a su vez, representa un flujo particular de la acción asociada al caso de uso. En los apartados siguientes sólo se desarrollan los diagramas de secuencia más significativos dentro de la funcionalidad requerida.

Este diagrama es uno de los más efectivos para modelar interacción de caso de uso permite el modelado de una vista *business* del escenario.

### 12.2.1. Diagrama de secuencia: Cargar datos

Con este proceso se pretende conseguir cargar en nuestro programa todos los datos necesarios para poder luego ser utilizado los diferentes algoritmos.

Los pasos que se seguirán en este proceso son los que se detalla a continuación:

1. El usuario pide poder cargar un fichero de datos.
2. El sistema le muestra un buscador para que selecciona su fichero.
3. El usuario le selecciona el fichero que desea cargar.
4. El sistema valida y carga el fichero.
5. El sistema prepara la ventana con los algoritmos.
6. El sistema le muestra dicha ventana al usuario.

Todo este proceso puede ser observado en el siguiente diagrama:

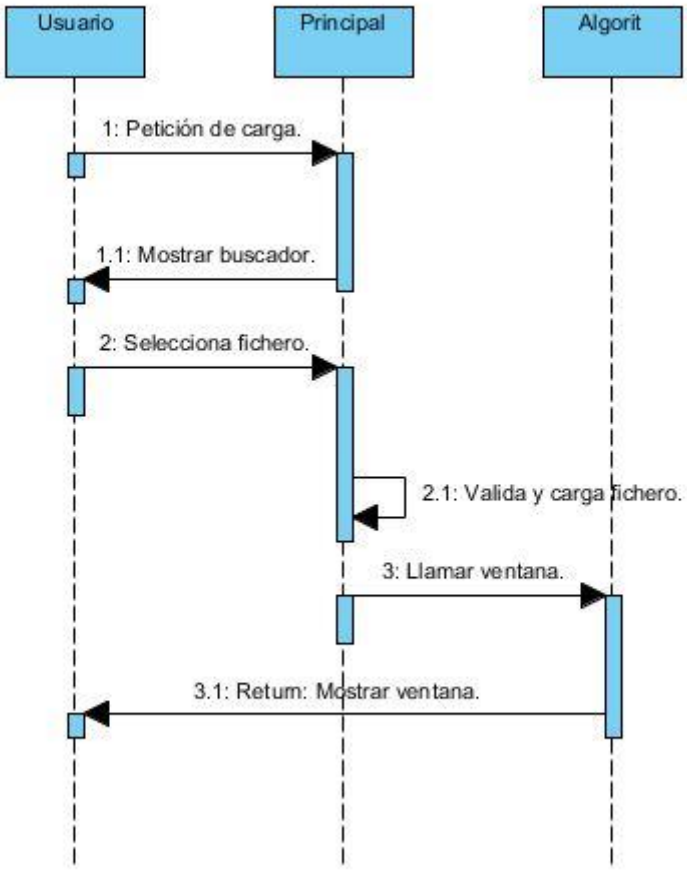



*Ilustración 81: Diagrama de secuencia: Cargar fichero.*

### 12.2.2. Diagrama de secuencia: Aplicar algoritmos

Con este proceso se pretende explicar cómo se ejecutaría un de los diferentes algoritmos que podemos encontrarnos.

Los pasos para seguir serían los siguientes:

1. El usuario selecciona un algoritmo.
2. Si el vector o la estructura que se use no está vacía se le pregunta al usuario si lo quiere borrar.
    a. El usuario decide qué hacer con esos datos.
3. El sistema ejecuta el algoritmo
4. El sistema avisa de que ha terminado.

Todo este proceso se puede observar en el diagrama que tenemos a continuación:

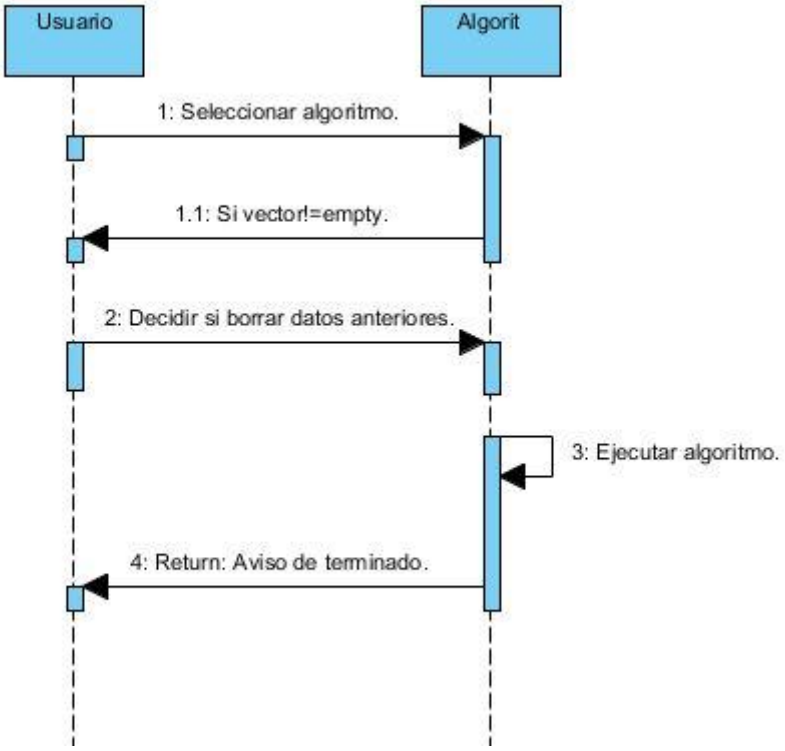



*Ilustración 82: Diagrama de secuencia: Aplicar algoritmo.*

### 12.2.3. Diagrama de secuencia: Generar PDF

El objetivo de este proceso es el de generar un fichero PDF con los datos que hemos obtenido para poder guardarlos y no perderlos.

Los pasos que se seguirán son los siguientes:

1. El usuario pide generar un PDF.
2. El sistema le pide al usuario los datos necesarios para este proceso.
3. El usuario le da al sistema todos los datos necesarios.
4. El sistema llama a la clase que se encarga de la generación del PDF
5. Esta clase genera el PDF.
6. El sistema avisa al usuario que se ha terminado.

Todo este proceso se pude observar en el siguiente diagrama:

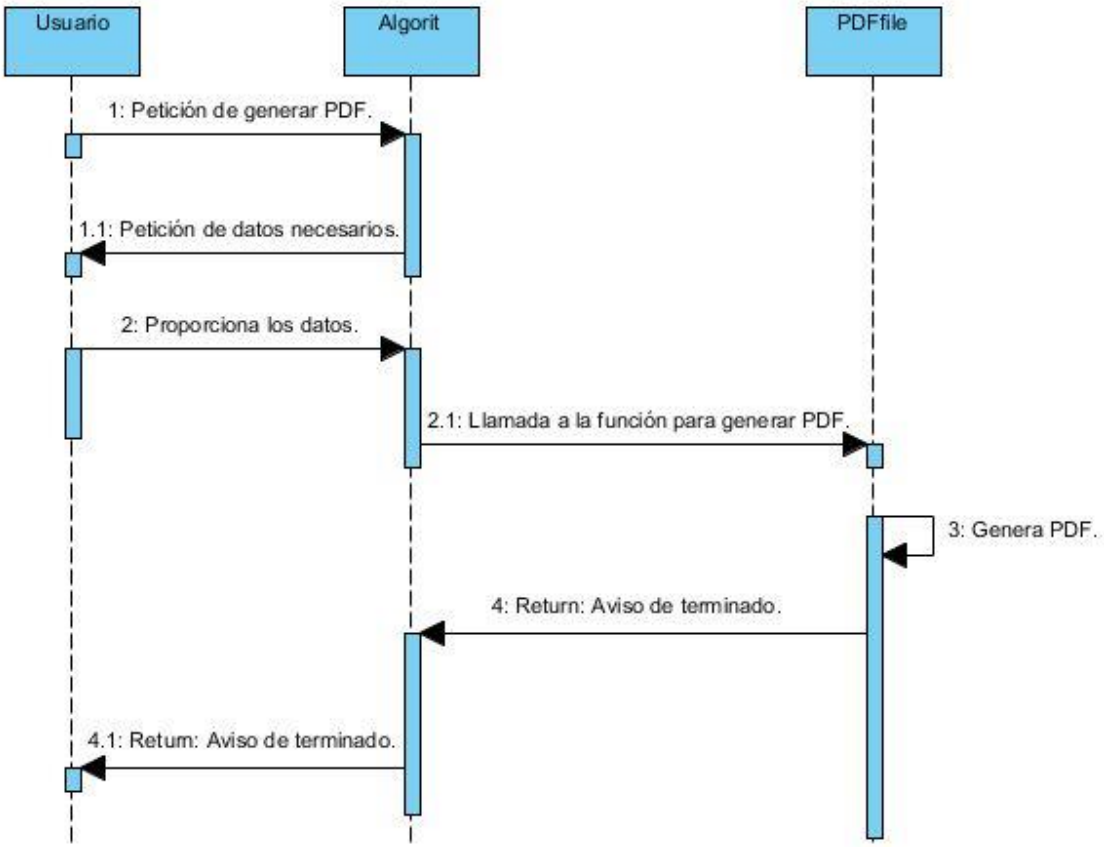



*Ilustración 83: Diagrama de secuencia: Generar PDF.*

# 13. Diseño de datos

## 13.1. Diagrama de clases

Una clase define un conjunto de objetos que tienen estado y comportamiento. El estado se describe mediante atributos y asociaciones mientras que el comportamiento lo describe las operaciones.

A continuación, se realizará una descripción de cada una de las clases del modelo de clases especificando para cada una de ellas sus características significativas. La representación gráfica de cada clase se realizará utilizando la notación UML.

### 13.1.1. Clase de base CA1: TFGPruebas

Esta es la clase base que nos genera la ventana principal de todo el programa, a continuación se puede observar la representación de esta clase.

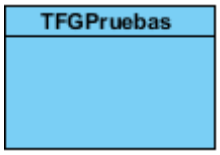



*Ilustración 84: CA1: TFGPruebas*

La especificación de esta clase de análisis puede verse en la siguiente tabla:

| **CA1: TFGPruebas** | |
|---|---|
| **Nombre** | TFGPruebas |
| **Descripción** | Esta clase se utiliza como clase base para la generación de la ventana principal. |
| **Atributos** | |
| **Métodos** | |

*Tabla 77:Especificación de CA1: TFGPruebas.*

### 13.1.2. Clase Alumno CA2: Alumno.

Esta clase se encarga de guardar toda la información referente a un alumno y a todos sus datos, así como todo lo necesario para la gestión.

La representación de la clase puede observarse a continuación.

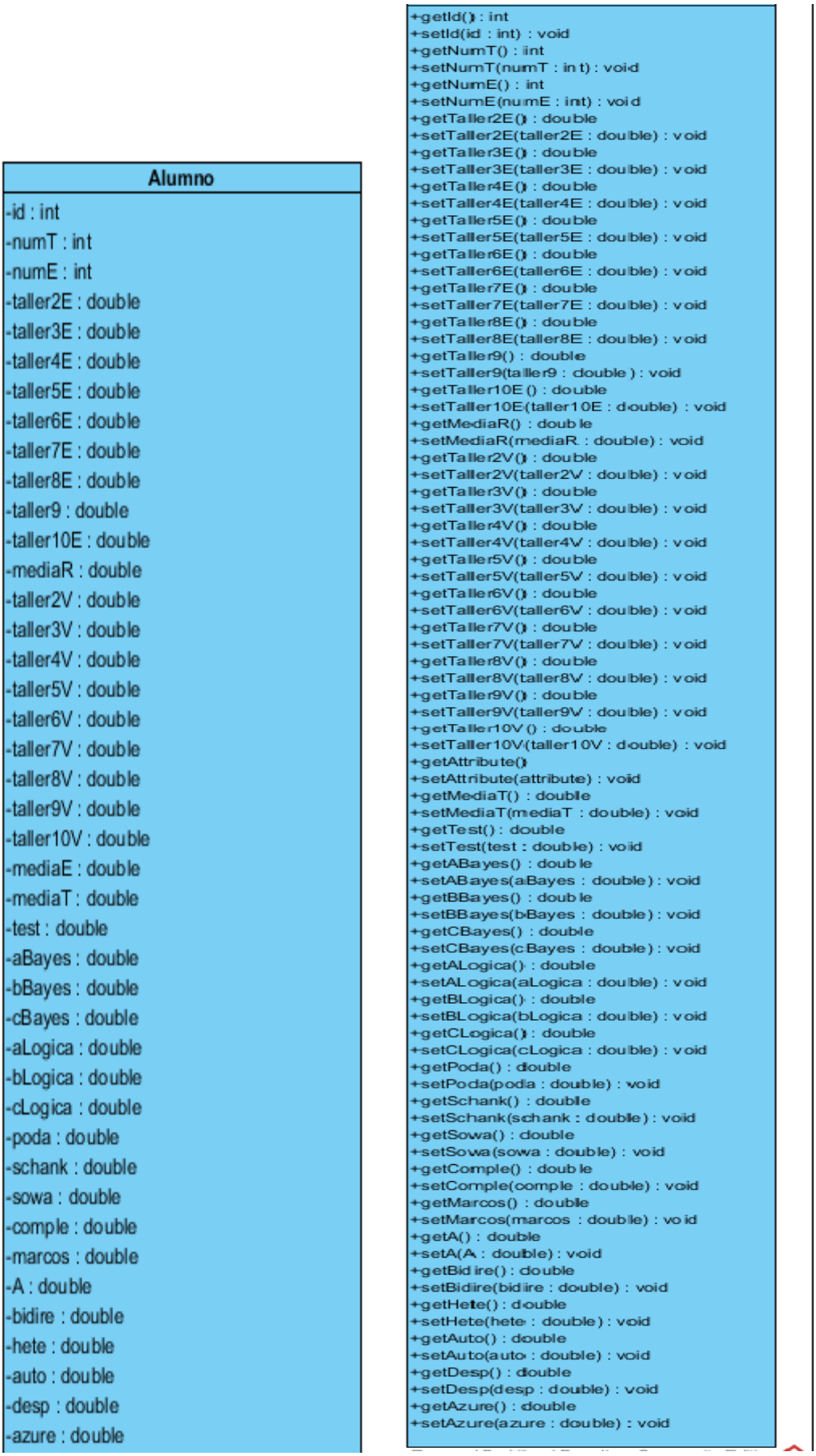



*Ilustración 85:CA2: Alumno*

La especificación de esta clase de análisis puede verse en la siguiente tabla:

| **CA2: Alumno** | |
|---|---|
| **Nombre** | Alumno |
| **Descripción** | Clase donde se guardaría y modificaría toda la información de un alumno. |
| **Atributos** | **Id:** Identificador de cada alumno.<br>Lo demás atributos que tiene esta clase son los mismos que ya hemos definidos en apartados anteriores, es decir los atributos que tenía nuestro conjunto de datos de investigación. |
| **Métodos** | Todos los métodos de esta clase serían todas las funciones set y get de cada uno de los atributos dichos anteriormente. |

*Tabla 78:Especificación de CA2: Alumno.*

### 13.1.3. Clase Principal CA3: Principal.

Utiliza la clase alumno para la gestión del archivo de datos.

La representación de la clase es la que podemos ver a continuación:

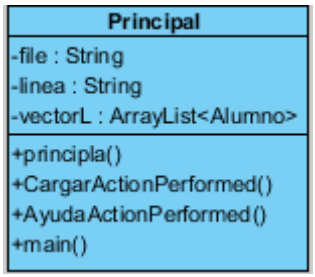



*Ilustración 86:CA3: Principal.*

La especificación de esta clase de análisis puede verse en la siguiente tabla:

| **CA3: Principal** | |
|---|---|
| **Nombre** | Principal |
| **Descripción** | Clase que nos genera la ventana principal y de gestión de archivo. |
| **Atributos** | **File:** Ruta del archivo donde guardamos los datos.<br>**Línea:** String donde vamos a ir guardando las líneas del archivo de datos.<br>**vectorL:** Vector donde vamos guardando los diferentes datos de cada alumno. |
| **Métodos** | **Principal():** Función donde se genera todos los componentes y atributos.<br>**CargarActionPerformed():** Función para la carda del archivo de datos en el programa.<br>**AyudaActionPerformed():** Función que activa y forma la clase de ayuda.<br>**Main():** Función que genera toda la ventana principal. |

*Tabla 79:Especificación de CA3: Principal.*

### 13.1.4. Clase Algoritmo CA4: Algorit

Clase para gestionar todos los diferentes algoritmos.

La representación de la clase es la que podemos ver a continuación:

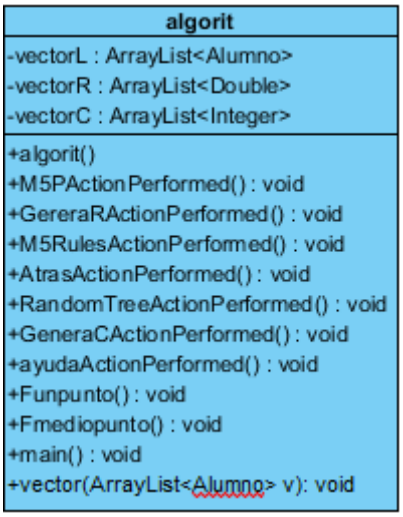



*Ilustración 87: CA4: Algorit.*

La especificación de esta clase de análisis puede verse en la siguiente tabla:

| **CA4: Algorit** | |
|---|---|
| **Nombre** | Algorit |
| **Descripción** | Clase para gestionar todos los diferentes algoritmos. |
| **Atributos** | **vectorL:** Vector para guardar los diferentes alumnos.<br>**vectorR:** Vector para guardar los resultados de regresión.<br>**vectorC:** Vector para guardar los resultados de clasificación. |
| **Métodos** | **algorit():** Función donde se genera todos los componentes y atributos.<br>**M5PActionPerformed():** Función para generar el resultado del algoritmo M5P.<br>**GeneraR():** Función para la creación del PDF para la solución de regresión.<br>**M5RulesActionPerformed():**Función para generar el resultado del algoritmo M5Rules.<br>**AtrasActionPerformed():** Función para poder volver a la ventana principal.<br>**RandomTreeActionPerformed():**Función para generar el resultado del algoritmo RandomTree.<br>**GeneraC():** Función para la creación del PDF para la solución de clasificación.<br>**Main():** Función que genera toda la ventana principal. |

*Tabla 80:Especificación de CA4: algorit.*

### 13.1.5. Clase PDFfile CA5: PDFfile

Clase para gestionar la creación de los archivos PDF.

La representación de la clase es la que podemos ver a continuación:


*Ilustración 88: CA5: PDFfile.*

La especificación de esta clase de análisis puede verse en la siguiente tabla:

| **CA5: PDFfile** | |
|---|---|
| **Nombre** | PDFfile |
| **Descripción** | Clase para gestionar la creación de los archivos PDF. |
| **Atributos** | |
| **Métodos** | **createPDF():** Función para crear el PDF con los resultados de regresión.<br>**createPDFC():** Función para crear el PDF con los resultados de clasificación. |

*Tabla 81:Especificación de CA5:PDFfile.*

### 13.1.6. Clase Ayuda CA6: ayuda

Clase para gestionar la ventana de ayuda del programa.

La representación de la clase es la que podemos ver a continuación:

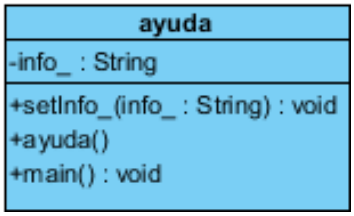



*Ilustración 89: CA6: ayuda.*

La especificación de esta clase de análisis puede verse en la siguiente tabla:

| **CA6: ayuda** | |
|---|---|
| **Nombre** | ayuda |
| **Descripción** | Clase para gestionar la ventana de ayuda del programa. |
| **Atributos** | **Info_:** La información que se va a mostrar como ayuda. |
| **Métodos** | **Setinfo_(string):** Función para recibir la información que se va a mostrar.<br>**algorit():** Función donde se genera todos los componentes y atributos.<br>**Main():** Función que genera toda la ventana principal. |

*Tabla 82: Especificación de CA6:ayuda.*

## 13.2. Especificación de las relaciones entre clases

Una vez descritas las clases que conforma el sistema se pasara a especificar las relaciones existentes entre ellas.

Para cada una de las relaciones se especificará el tipo de relación que mantienen, indicando la cardinalidad con la que participan cada una de las clases, y una breve descripción de la relación.

### 13.2.1. Relación Alumno – Principal

En la siguiente tabla se muestra la relación existente entre las dos clases:

| Clase | Cardinalidad | Relación | Cardinalidad | Clase |
|---|---|---|---|---|
| Alumno | 1,n | Son leídos | 1,1 | Principal |
| **Descripción:**<br>Uno o varios alumnos son leídos del fichero de datos por una clase principal. | | | | |

*Tabla 83: Relación Alumno – Principal*

### 13.2.2. Relación Principal – Algorit

En la siguiente tabla se muestra la relación existente entre las dos clases:

| Clase | Cardinalidad | Relación | Cardinalidad | Clase |
|---|---|---|---|---|
| Principal | 1,1 | es utilizada | 1,1 | algorit |
| **Descripción:**<br>La clase principal es utilizada por una clase algorit para ejecutar los diferentes algoritmos. | | | | |

*Tabla 84: Relación Principal – algorit*

### 13.2.3. Relación Algorit - PDFfile

En la siguiente tabla se muestra la relación existente entre las dos clases:

| Clase | Cardinalidad | Relación | Cardinalidad | Clase |
|---|---|---|---|---|
| algorit | 1,1 | Es utilizada | 1,1 | PDFfile |
| **Descripción:**<br>La clase algorit es utilizada por la clase PDFfile para generar los PDF con los resultados obtenidos. | | | | |

*Tabla 85: Relación algorit – PDFfile*

## 13.3. Diagrama de clases del sistema

Una vez analizadas las diferentes clases y sus relaciones, se puede generar el diagrama de clases del modelo estructural del sistema. Mediante este diagrama se pretende mostrar los aspectos estructurales acerca de las clases consideradas dentro del dominio del problema y de las relaciones principales necesarias entre dichas clases para cumplir los requisitos funciones del sistema.

A continuación, se muestra el diagrama de clases del sistema utilizando UML, los diagramas de clases pueden alcanzar distintos grados de especificación, en este caso tan solo se ha implementado un diagrama de clases en el que se indican las clases que deben aparecer en el sistema. Existen además otras clases que son utilizadas de manera auxiliar y de forma interna por otras, pero con el objetivo de no dificultar la compresión del diagrama de clases dichas relaciones no se han incluido.

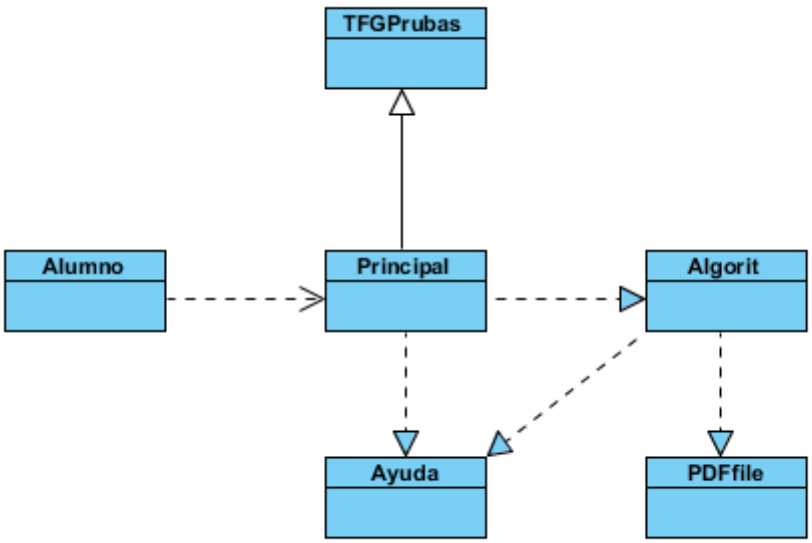



*Ilustración 90: Diagrama de clases del sistema.*

# 14. Diseño de la interfaz

La interacción del usuario con el sistema propuesto se realiza a través de la interfaz de usuario o IGU. Esta interfaz se compone de un conjunto de ventanas organizadas jerárquicamente que proporcionan soporte a toda la funcionalidad implementada en nuestra herramienta.

La especificación de la IGU consiste en la identificación de cada una de las ventanas con las que puede interaccionar el usuario indicando su finalidad (funcionalidad que aporta) y descripción de cada uno de sus componentes gráficos.

Desde la perspectiva del diseño durante la evolución de la IGU se han tenido en cuenta una serie de directrices en cuanto a ergonomía, facilidad de uso, consistencia entre otras.

Partiendo de un tipo de usuario final del sistema que puede tener un nivel bajo de conocimientos sintácticos de la aplicación, a lo largo del desarrollo de la IGU se han tenido en cuenta los siguientes aspectos:

- **Optimizar el consumo de recursos del sistema:** Se evita en la medida de lo posible mantener estructuras de datos en memoria que no están siendo utilizadas en ninguna tarea de procesamiento.
- **Ofrecer un tiempo de respuesta optimo ante las acciones de control:** Debido a que la interacción hombre-máquina de la IGU está basada fundamentalmente en la respuesta ante eventos, este aspecto puede ser especialmente crítico cuando el sistema realiza tareas que consumen una gran cantidad de recursos computacionales. Por este motivo es fundamental optimizar la implementación de las funciones de negocio con el objetivo de que se puedan ejecutar lo más eficientemente posible.
- **Establecer un sistema de gestión de errores:** Cada mensaje o error que puede emitir el sistema está codificado. De este modo se puede ofrecer información acerca de la naturaleza del error, sus consecuencias y su recuperación.
- **Organización de la interfaz:** Los componentes se organizan de acuerdo con la funcionalidad implementada.

Para el diseño de una interfaz se debe estar en permanente contacto con el cliente que la solicita y presentar varios prototipos antes de la interfaz gráfica definitiva. Es frecuente que aparezcan errores cuando el uso de la herramienta o aplicación aumenta. Por esto es necesario que se pruebe el sistema con usuarios potenciales y en condiciones similares a las que se encontrará el producto una vez finalizado, de manera que podamos detectar posibles errores ocultos.

Ahora se va a describir todos los componentes gráficos de la interfaz del sistema. Para casa ventana se especifican sus características en cuanto a finalidad, entradas de datos disponibles, salidas de datos proporcionadas, opciones disponibles y elementos constituyentes.

## 14.1. Ventana principal

Se trata de la ventana que nos aparece nada más iniciemos la aplicación, en esta ventana nos podemos encontrar solamente dos accesos, por lo tanto, tendríamos dos acciones disponibles y estas son el cargar los datos o mostrarnos la ayuda.

A continuación podríamos ver como veríamos dicha ventana.


*Ilustración 91: Ventana principal.*

Las diferentes acciones que puedes realizar el usuario son:

- **Cargar los datos:** Pulsando en el botón "Cargar Archivo" se le abriría un buscador de archivos para que busque en su ordenador el documento con los datos para su carga en el programa.

- **Datos a manos:** Sirve para introducir por el usuario los diferentes datos que un alumno puede tener y luego ejecutar uno de los algoritmos para esos datos.

- **Pedir ayuda:** Pulsando en el botón que pone "Ayuda" el usuario puedo pedir que se le muestre una ventana con una ayuda sobre el programa.

## 14.2. Ventana de algoritmos

Esta ventana es la que nos aparece cuando hemos cargado el archivo con los datos correctamente, en ella podremos aplicar los diferentes algoritmos que hemos sacado de la investigación así como la generación de los documentos PDF con los resultados.

La apariencia de esta ventana es la que podemos ver a continuación.

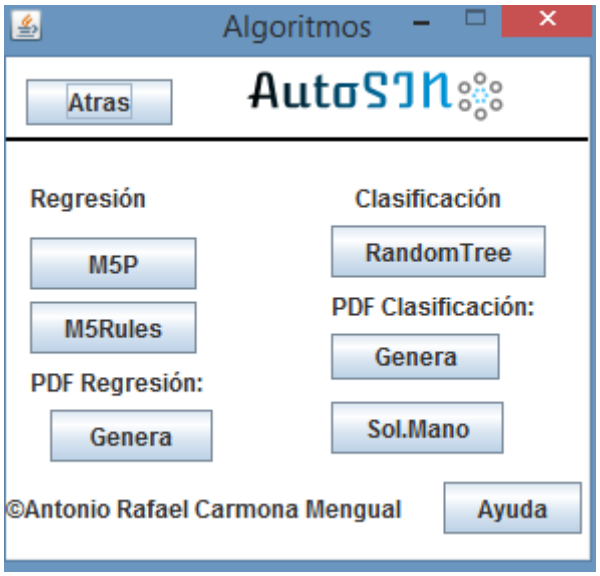



*Ilustración 92: Ventana algoritmos.*

Las diferentes acciones que puede realizar el usuario en esta ventana podrían ser:

- **Aplicar M5P:** Pulsando en el botón “M5P”, el usuario pude aplicar el resultado que hemos obtenido en el algoritmo M5P a los datos que hemos cargado.

- **Aplicar M5Rules:** Pulsando en el botón “M5Rules”, el usuario pude aplicar el resultado que hemos obtenido en el algoritmo M5Rules a los datos que hemos cargado.

- **Aplicar RandomTree:** Pulsando en el botón “RandomTree”, el usuario pude aplicar el resultado que hemos obtenido en el algoritmo RandomTree a los datos que hemos cargado, decidiendo posteriormente que nivel de frontera de los dos que tenemos se quiere aplicar.

- **Genera:** Pulsando en el botón “Genera”, el usuario puede construir el PDF con los resultados que hemos obtenido de algún algoritmo, como vemos tenemos dos unos para el PDF de regresión y otro para clasificación, para diferenciar cual es cada uno tiene una etiqueta encima de cada uno para diferenciarlos.

- **Pedir ayuda:** Pulsando en el botón que pone “Ayuda” el usuario puedo pedir que se le muestre una ventana con una ayuda sobre el programa.

- **Volver:** Pulsando el usuario el botón “Atrás”, el usuario puede volver a la ventana principal, la explicada en el punto anterior.

- **Solución a Mano:** Si el usuario pulsa este botón entonces se nos abriría una ventana para que el usuario pueda poner los valores de un usuario y saber que solución podría obtener para los diferentes algoritmos.

## 14.3. Ventana de solución a mano.

Esta ventana es la que nos aparece cuando hemos pulsado en el botón “Sol.Mano” en la ventana de algoritmos o también pulsando “Datos a mano” en la ventana principal, en ella podremos aplicar los diferentes algoritmos que hemos sacado de la investigación pero esta vez para unos datos introducidos por el usuario a manos.

NOTA: El valor -1 se identifica como dato perdido o que no se tiene del alumno.

La apariencia de esta ventana es la que podemos ver a continuación.

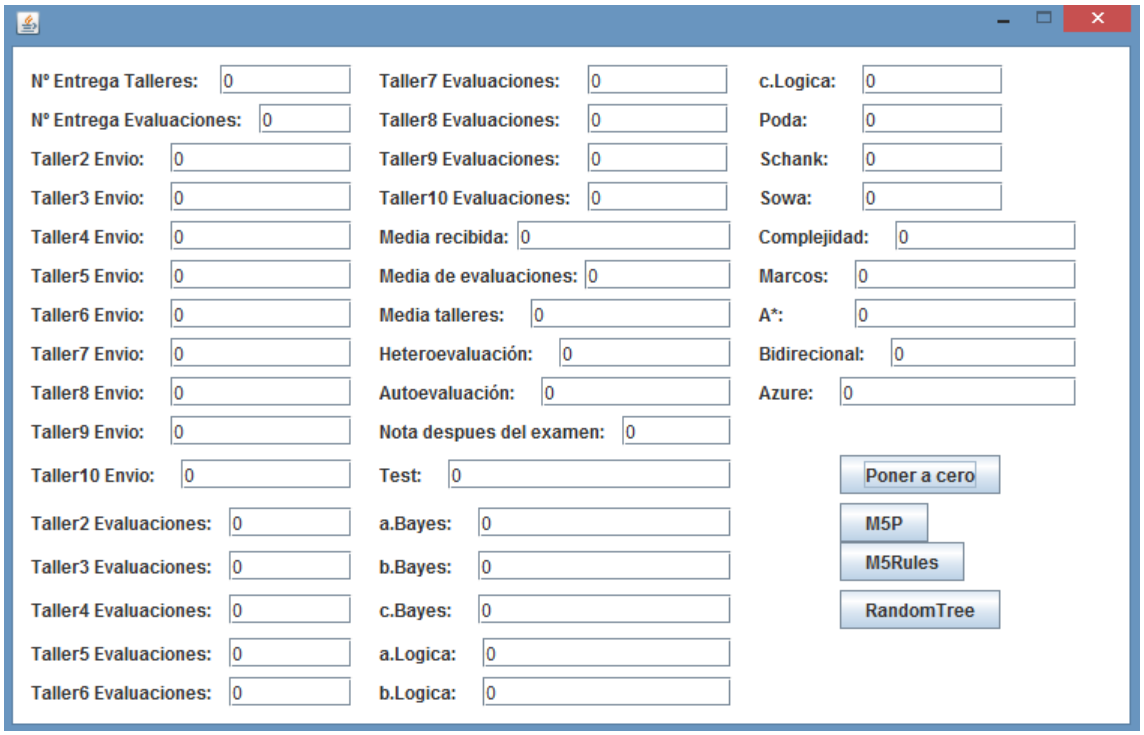



*Ilustración 93: Ventana solución a mano.*

Las diferentes acciones que puede realizar el usuario en esta ventana podrían ser:

- Introducir un valor para cada variable que tengamos.
- Pulsar el botón **M5P** para aplicar este algoritmo a esos datos.
- Pulsar el botón **M5Rules** para aplicar este algoritmo a los datos introducidos.
- Pulsar el botón **RandomTree** para aplicar algoritmo.
- Pulsando el botón **Poner a cero** todas las variables se ponen con el valor 0.

# Sección III: Conclusiones y futuras mejoras

# 15. Conclusiones y futuras mejoras

En esta sesión lo que se va a realizar es explicar las conclusiones que hemos podido observar después de realizar esta investigación y el informe resultante, además también se hablara sobre futuras mejoras para poder tener un mejor resultado en esta investigación.

Nuestra investigación lo que ha tratado es ver si una metodología para la predicción de la decisión de un profesor acerca de la aceptación de una nota propuesta o también si se es capaz de obtener la nota que este alumno sacaría en el examen sabiendo una serie de datos sobre este alumno, es decir estamos intentando saber la nota final de un alumno en una asignatura mediante el uso de la Minería de Datos, Autoevaluación, Regresión y Clasificación.

Por lo tanto, lo primero que podemos decir sobre este tema es que después de ver todo lo realizado en esta investigación todavía podemos decir que el tema de la autoevaluación esta todavía un poco verde por lo tanto todavía no podemos llegar a la conclusión de poder eliminar por completo la realización de un examen final para poder evaluar los conocimientos de un alumno en una materia, esto es porque como hemos visto no se ha podido encontrar un algoritmo el cual nos dé un rendimiento tan bueno como para poder decir que no tiene errores.

También podemos llegar a decir o llegar a la conclusión de que los errores tan altos que podemos tener pueden ser por varias razones, estas podrían ser la siguientes, una razón podría ser por los datos que tenemos los cuales están compuestos por muy pocas instancias y alguna de muy mala calidad, pero esto no es culpa de quien recogió los datos sino por otras razones, como por ejemplo porque una gran mayoría de los alumnos durante la duración del curso no hacen las actividades que son mandadas por el profesor por la razón de que no le apetece hacerlas o por qué le parece demasiado difícil dichas actividades y no se atreven hacerlas, otra razón de los errores podría ser por lo algoritmos que hemos intentado usar ya que a lo mejor no hemos dado todavía con el algoritmo el cual es el bueno para nuestro conjunto de datos.

En el futuro nos gustaría poder realizar experimentos con muchas más nuevas instancias y con una mejor calidad para ver si podemos llegar al final a conseguir obtener un clasificador realmente bueno para poder llegar a obtener la nota del alumno o si nos creemos la que nos propone, además también nos gustaría poder realizar más pruebas, pero en diferentes asignaturas no solo en "*Sistemas Inteligentes*" como se ha realizado para la investigación de este proyecto.

Todas esas nuevas pruebas que nos gustaría realizar serían para poder llegar en un futuro a obtener resultados buenos lo antes posible, con esto lo que queremos decir es que con las notas que se pueden obtener durante el curso antes de realizar el examen llegar a poder sacar la nota que puede obtener el alumno, cosa que ahora mismo no se

ha podido conseguir por que como ya hemos dicho no tenemos demasiados datos buenos en los que tengamos esas notas principales antes de la realización del examen.

Referente a la herramienta se podría mejorar los resultados que nos da si mejoramos lo anteriormente dicho, además se podría ver más errores y mejorarlos para que no sucedan, así como la interfaz todavía tiene mucho trabajo para poder realizar mejoras.

# Bibliografía


[1] Aprendizaje automático, disponible en: https://es.wikipedia.org/wiki/Aprendizaje_autom%C3%A1tico, última visita el 9 de noviembre de 2017.

[2] Minería de datos, disponible en: https://es.wikipedia.org/wiki/Miner%C3%ADa_de_datos, última visita el 11 de noviembre de 2017.

[3] Azure, disponible en: https://www.microsoft.com/cognitive-services/en-us/text-analytics-api, última visita el 22 de febrero de 2018.

[4] Weka, disponible en: https://es.wikipedia.org/wiki/Weka_(aprendizaje_autom%C3%A1tico), última visita el 17 de noviembre de 2017.

[5] Aprendizaje supervisado, disponible en: https://es.wikipedia.org/wiki/Aprendizaje_supervisado, última visita el 12 de diciembre de 2017.

[6] Aprendizaje no supervisado, disponible en: http://www.cs.us.es/~fsancho/?e=77, última visita el 12 de diciembre de 2017.

[7] Feedback, disponible en: https://definicion.de/feedback/, última visita el 23 de noviembre de 2017.

[8] Correlación, disponible en: https://es.wikipedia.org/wiki/Correlaci%C3%B3n, última visita el 28 de abril de 2018.

[9] Correlación de Pearson, disponible en: https://es.wikipedia.org/wiki/Coeficiente_de_correlaci%C3%B3n_de_Pearson, última visita el 28 de mayo de 2018.

[10] Regresión, disponible en: https://es.wikipedia.org/wiki/An%C3%A1lisis_de_la_regresi%C3%B3n, última visita el 30 de mayo de 2018.

[11] Tipo de regresión, disponible en: http://www.ub.edu/stat/GrupsInnovacio/Statmedia/demo/Temas/Capitulo13/B0C13m1t2.htm, última visita el 30 de mayo de 2018.

[12] Mean Absolute error, disponible en: https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d, última visita 20 de junio de 2018.

[13] Root mean squared error, disponible en: http://www.statisticshowto.com/rmse/, última visita 20 de junio de 2018.

[14] Relative Absolute error, disponible en: https://www.gepsoft.com/gxpt4kb/Chapter10/Section2/SS15.htm, última visita 20 de junio de 2018.

[15] Root relative squared error, disponible en: https://www.researchgate.net/post/What_is_root_relative_squared_error, última visita 20 de junio de 2018.

[16] RandomCommittee, disponible en: http://weka.sourceforge.net/doc.dev/weka/classifiers/meta/RandomCommittee.html, última visita 2 de julio de 2018.

[17] RandomForest, disponible en: https://es.wikipedia.org/wiki/Random_forest, última visita 2 de julio de 2018.

[18] Bagging, disponible en: https://en.wikipedia.org/wiki/Bootstrap_aggregating, última visita 2 de julio de 2018.

[19] M5Rules, disponible en: http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/M5Rules.html, última visita 2 de julio de 2018.

[20] M5P, disponible en: https://wiki.pentaho.com/display/DATAMINING/M5P, última visita 2 de julio de 2018.

[21] RBFRegressor, disponible en: https://wiki.pentaho.com/display/DATAMINING/RBFRegressor, última visita 2 de julio de 2018.

[22] Clasificación, disponible en: https://es.wikipedia.org/wiki/Clasificaci%C3%B3n_estad%C3%ADstica, última visita 10 de julio de 2018.

[23] Precision y Recall, disponible en: https://en.wikipedia.org/wiki/Precision_and_recall#Precision, última visita 10 de julio de 2018.

[24] F-Measure, disponible en: https://metacademy.org/graphs/concepts/f_measure, última visita 10 de julio de 2018.

[25] ROC-Area, disponible en: https://es.wikipedia.org/wiki/Curva_ROC, última visita 10 de julio de 2018.

[26] Kstar, disponible en: http://weka.sourceforge.net/doc.dev/weka/classifiers/lazy/KStar.html, última visita 10 de julio de 2018.

[27] RandomTree, disponible en: http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/RandomTree.html, última visita 10 de julio de 1018.

[28] ClassificationViaRegression, disponible en: http://weka.sourceforge.net/doc.dev/weka/classifiers/meta/ClassificationViaRegression.html, última visita el 10 de julio de 2018.

[29] Logistic, disponible en: http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/Logistic.html, última visita el 10 de julio de 2018.

[30] Jrip, disponible en: http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/JRip.html, última visita el 10 de julio de 2018.

[31] REPTree, disponible en: http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/REPTree.html, última visita el 10 de julio de 2018.

[32] IterateClassifierOptimizer, disponible en: http://weka.sourceforge.net/doc.dev/weka/classifiers/meta/IterativeClassifierOptimizer.html, última visita el 10 de julio de 2018.

[33] SGD, disponible en: http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SGD.html, última visita el 18 de julio de 2018.

[34] NaiveBayes, disponible en: http://weka.sourceforge.net/doc.dev/weka/classifiers/bayes/NaiveBayes.html, última visita el 18 de julio de 2018.

[35] MultiClassClassifier, disponible en: https://en.wikipedia.org/wiki/Multiclass_classification, última visita el 18 de julio de 2018.

Anexo I

# Anexo I: Weka

# 1. Introducción al programa WEKA.

## 1.1.Aspectos generales

El software Weka para el estudio de la MD es gratuito y puede ser descargado en la siguiente dirección: http://www.cs.waikato.ac.nz/ml/WEKA/downloading.html.

Una vez instalada, para nuestro caso la versión 3.8, la ejecutamos y nos aparecerá la interface principal, que se corresponde con la ventana de la ilustración 94, que ofrece cinco posibilidades de trabajo: Explorer, Experimenter, KnowledgeFlow, Workbench y Simple CLI.

¿Para qué sirven cada una de estas opciones? Pues bien, hablaremos un poco de todas ellas, aunque nos centraremos en Explorer, que es el modo más usado para la MD y nos permite acceder a la mayoría de las funciones que tiene WEKA y realizar las operaciones sobre un archivo de datos.

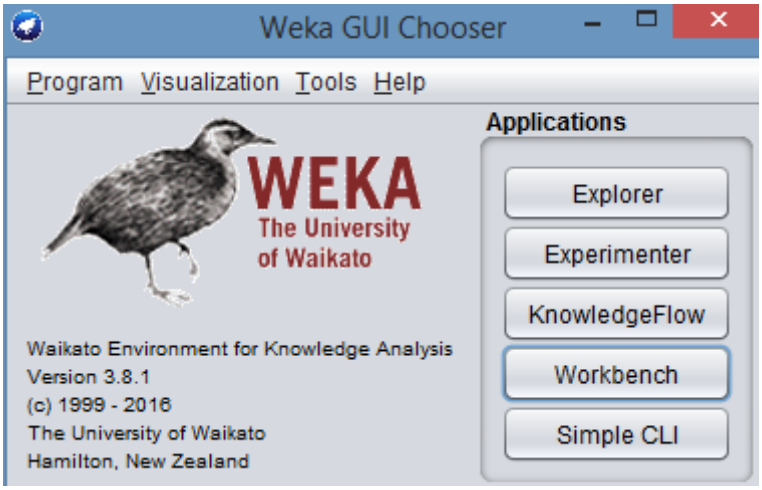



*Ilustración 94: Anexo I: Interfaz principal Weka.*

- **Explorer:** Es la interface visual de WEKA para trabajar de manera gráfica de una manera sencilla. Este modo permite procesar, clasificar, asociar y visualizar datos de una manera fácil e intuitiva sobre un solo archivo de datos, es decir nos permite el preprocesamiento de datos y la experimentación usando una gran variedad de algoritmos.

- **Experimenter:** Es un modo útil para aplicar uno o varios métodos de clasificación de manera automática. Con esta ventana se facilita la realización de experimentos a gran escala, para resumir sería coma el *Explore* pero a gran escala.

- **KnowledgeFlow:** Entorno que permite arrastrar componentes de Weka y conectarlos para hacer experimentos.

- **Workbench:** Es para entrar en una especie de interfaz en la que se puede encontrar todas las aplicaciones que tiene Weka en su diferentes opciones pero juntas en la misma ventana.

- **Simple CLI:** Se le conoce como interface línea de comandos y se usa para llamar directamente a los paquetes de Java que WEKA incorpora.

Una vez abierta la ventana **Explorer,** que será la más útil para nuestro caso, se pueden observar seis pestañas que nos permitirán realizar las siguientes tareas:

- **Preproces:** Es el primer paso para poder empezar a trabajar y definir el origen de los datos. Las herramientas de preprocesamiento en WEKA se llaman filtros y contiene, entre otros, filtros para la discretización, normalización, reemplazamiento y combinación de atributos. El tipo de filtros más utilizados son los no supervisados sobre los atributos. Aquellos que son independientes de los algoritmos aplicados.
  Es evidente que lo primero será cargar el conjunto de datos y esto puede hacerse de cuatro formas diferentes:
  - Abriendo un archivo que tengamos en nuestro equipo mediante **Open File**.
  - Abriendo un archivo mediante una dirección de internet con **Open URL.**
  - Abriendo una base de datos con **Open DB.**
  - Generando el conjunto de datos mediante **Generate.**

- **Classify:** En la segunda pestaña podemos entrar en el modo clasificación, conocida en algunas ocasiones como aprendizaje supervisado. Con esta opción, podemos clasificar los datos haciendo uso de técnicas, entre otras, de clasificación y regresión.

- **Cluster:** Esta opción dispone de distintos algoritmos con los que se puedan agrupar los datos en base a uno o varios criterios.

- **Associate:** En esta cuarta pestaña se pueden encontrar reglas de asociaciones entre los datos. Se considera la opción más fácil de utilizar de las seis y consiste en elegir el método deseado para encontrar asociaciones entre los datos, así como la posibilidad de configurarlo.

- **Select atributes:** En esta pestaña podemos acceder a la selección de atributos, cuyo objetivo es identificar qué conjunto de datos poseen atributos similares con el objetivo, entre otros, de reducir su número. Dentro de esta opción tenemos que seguir dos pasos, primero seleccionar el método para la evaluación de atributos a través de *Atribute Evaluator*, que sirve para asignar a cada atributo un peso específico. El segundo y último sería elegir el método de búsqueda.

- **Visualize:** Este apartado ofrece la posibilidad de ver de forma gráfica cómo se distribuyen todos los atributos que representan los posibles pares de combinaciones, con el objetivo de extraer información por medio de técnicas visuales.

## 1.2.Descripción de la base de datos de ejemplo elegida (IRIS)

Para los siguientes apartados se ha utilizado un conjunto de datos para poder realizar diferentes pruebas para poder enseñar más detallado lo que se estaría explicando en ese momento en ese anexo.

Entonces se ha decidido utilizar la base de datos llamada “Iris”, está la podemos encontrar para su descarga en el siguiente enlace, http://www.cas.mcmaster.ca/~cs4tf3/iris.arff, esta es quizás la base de datos más conocida que se puede encontrar en la literatura de reconocimiento de patrones, el artículo de *Fisher* es un clásico en este campo y todavía se sigue mencionado con mucha frecuencia hasta el día de hoy, esta base de datos sirve para la clasificación de un tipo de flor llamada iris, la clasificación se hace en tres subconjuntos que existe de este tipo de flor.

Este conjunto de datos está compuesto por 112 patrones de entrenamiento y 38 patrones de test, posee 4 variables independientes de entrada las cuales son:

- **Sepal length:** Representa la longitud en cm del sépalo.
- **Sepal width:** Representa el ancho en cm del sépalo.
- **Petal length:** Representa la longitud en cm del pétalo.
- **Petal width:** Representa el ancho en cm del pétalo.

La base de datos consta como salida 3 clases diferentes, estas son **setosa, virginica** o **versicolor.**

## 1.3.Manejo de archivos en WEKA

El software trabaja con un formato de archivo que se denomina arff (Atribute Relation File Format), y que consta de tres partes:

1. **Cabecera:** Donde se define el nombre del archivo a través de la expresión *@relation <nombre>.*
2. **Declaración de atributos:** Se corresponde con el segundo bloque y es el lugar donde se define los atributos que vamos a estudiar, así como su tipo, por medio de expresión *@atribute<nombre><tipo>.* Los tipos de atributos con los que puede trabajar WEKA serían:
    - **Números reales:** *NUMERIC.*
    - **Números enteros:** *INTEGER.*
    - **Fechas:** Cuyo formato es:
        - Día: dd
        - Mes: MM
        - Año: yyyy
        - Horas: HH
        - Minutos: mm
        - Segundos: ss
    - **Cadenas de texto:** *STRING.*
    - **Enumerados:** Se expresa entre llaves y separadas por comas los distintos valores del atributo. *@attribute ejemplo {caso1,caso2,caso3,..}.*
3. Es la parte final del archivo, se inicia con *@data* y es el lugar donde insertamos los datos que componen nuestra base de datos. Los atributos deben estar separados por comas y con saltos de línea cada instancia.

Existe la posibilidad de pasar de un archivo en formato Excel a un archivo con el formato arff, para lo cual deberíamos realizar los siguientes pasos:

- Desde la base de datos en formato Excel, eliminamos las columnas cuyos atributos son irrelevantes para nuestro estudio y también las instancias que no son adecuadas.
- A continuación, guardamos el archivo de Excel en formato CSV con una cosa importante que es delimitarlo por comas, para poder manipularlo antes de abrirlo con WEKA.
- Con esto tenemos el fichero en formato csv, entonces lo abrimos por medio de un bloc de notas (botón derecho/abrir con/bloc de notas) y realizamos los siguientes cambios, reemplazamos las comas (,) por puntos(.) y los puntos y comas (;) por comas (,).
- Con esto ya lo tenemos casi terminado lo último seria escribir en el archivo la cabecera, las características de los atributos y los datos que queremos estudiar y grabamos el archivo con la extensión arff.

## 2.Análisis de los datos con WEKA

Una vez que ya tenemos nuestro archivo en formato arff pasaremos a la fase de análisis, para ello abriremos el programa y pulsaremos en la pestaña de *Explorer* (explorar) para poder cargar nuestro archivo, del cual hemos hablado antes.

Pulsando en cada atributo podemos obtener información sobre cada uno de ellos, como por ejemplo, acerca de qué tipo se trata, si es nominal o numérico, el valor máximo o mínimo que pueden obtener los atributos numéricos…,etc. Observemos, como aspecto muy importante, que en la parte inferior derecha (propiedades de los atributos) puede verse representado geométricamente un histograma con los valores que toma el atributo seleccionado. Existe también la posibilidad de ver un histograma con todos los atributos a la vez.

### 2.1.Preprocesamiento de los datos

El primer paso para un análisis de datos con WEKA es el preprocesamiento de estos en la pestaña *preprocess*  Ilustración 95. Esta operación se lleva a cabo mediante el uso de filtros, que pueden ser aplicados a los atributos o a las instancias. En general, el tipo de filtro es no supervisado, esto es, el resultado obtenido es independiente del tipo de algoritmos que se utilice a posteriori.

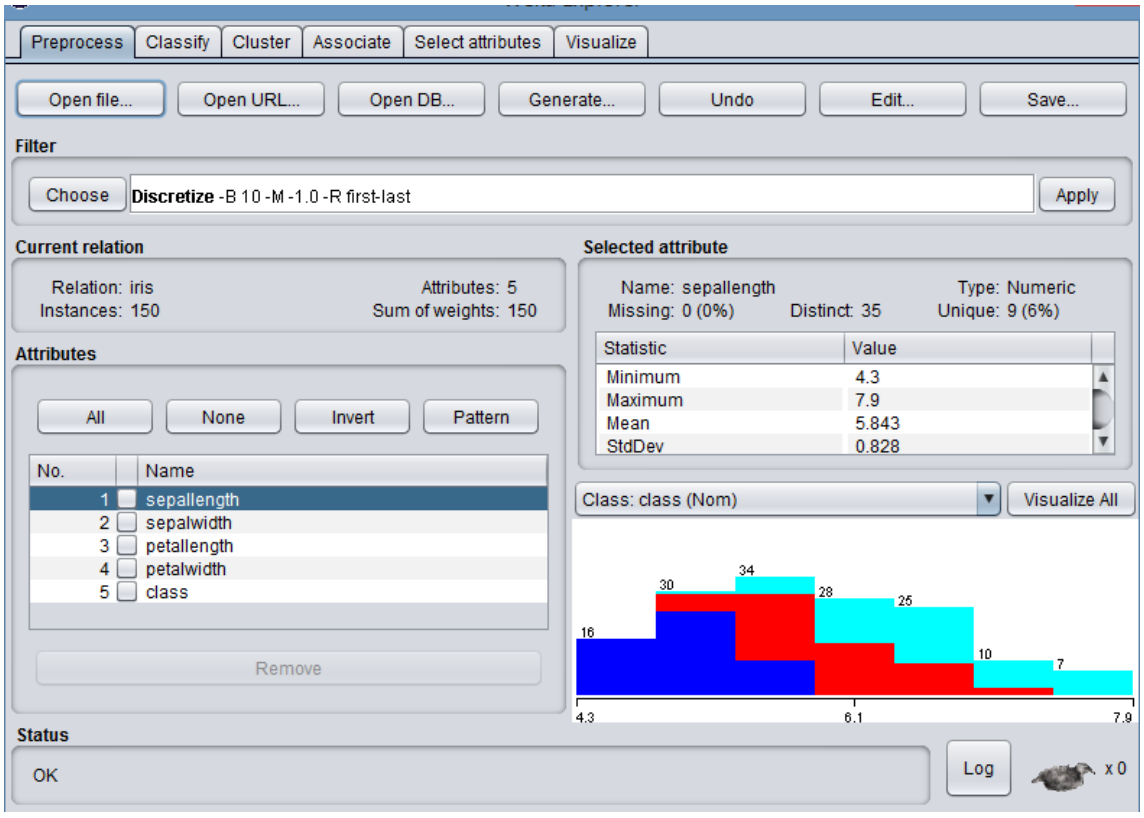



*Ilustración 95: Anexo I: Interfaz de Preprocess.*

WEKA permite usar numerosos filtros, por lo que podemos realizar transformaciones de todo tipo sobre nuestros datos. Para poder hacer uso de esta herramienta, se debe seleccionar el botón *choose*, donde tendremos acceso a un gran número de opciones, entre las que se encuentran:

- Filtrar atributos.
- Modificar el tipo de atributos (como por ejemplo, discretizar).
- Realizar muestreos sobre los datos.
- Unificar los valores de un atributo.
- Normalizar los atributos numéricos.

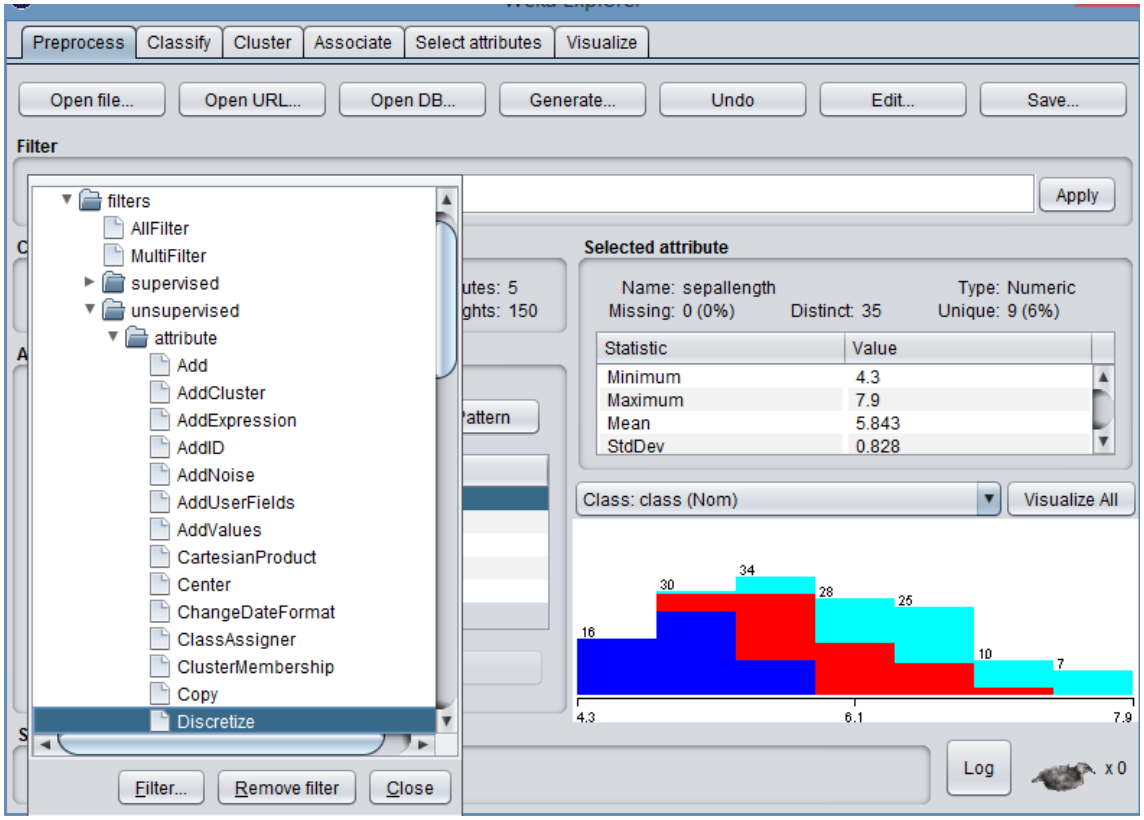



*Ilustración 96: Anexo I: Interfaz con filtros.*

Vamos a hablar de algunos filtros los cuales están en la carpeta no supervisados, en la opción *atribute* que vemos en la ilustración 96, existiendo la posibilidad de usar filtros como por ejemplo:

- **Add**: aporta la posibilidad de añadir un atributo más. Debemos proporcionar el nombre de nuestro atributo, qué posición va a ocupar y los posibles valores separados entre comas. Si estos valores no se especifican el programa supondrá que este nuevo atributo será numérico.

- **AddExpressión**: es uno de los filtros más útiles ya que podemos añadir al final de un atributo el valor de una función.

- **Copy**: realiza una copia del conjunto de atributos en los datos. Este filtro sirve para guardar los atributos originales ya que algunos filtros al utilizarlo destruyen los datos originales.

- **Normalize**: normaliza todos los datos de manera que pasen a tomar los valores 0 o 1.

## 2.2.Visualización de los datos

El modo *visualize* nos permite ver gráficamente cómo es la distribución de todos los atributos en gráficas de dos dimensiones, en la que se comparan a través de dicha representación la combinación de los atributos de par en par, permitiéndonos ver correlaciones y asociaciones entre los atributos de forma gráfica.
En el caso en el que estemos interesados en encontrar este tipo de correlaciones o asociaciones únicamente para dos atributos cualesquiera, se podrían hacer de forma más sencilla a través de una recta de regresión utilizando el software Excel.
Ahora bien, esta opción es relativamente sencilla cuando el número de atributos es de dos, en cambio, si el número de atributos es elevado, como suceda en varios conjuntos de datos, esta tarea resulta más complicada. Para ello, cuando disponemos de numerosos atributos, y queremos tener una primera impresión global de las posibles relaciones entre pares de ellos, utilizaremos la opción *Visualize* de WEKA seleccionando la pestaña correspondiente de la pantalla inicial. Al hacerlo podremos observar la ventana que aparece en la ilustración 97.

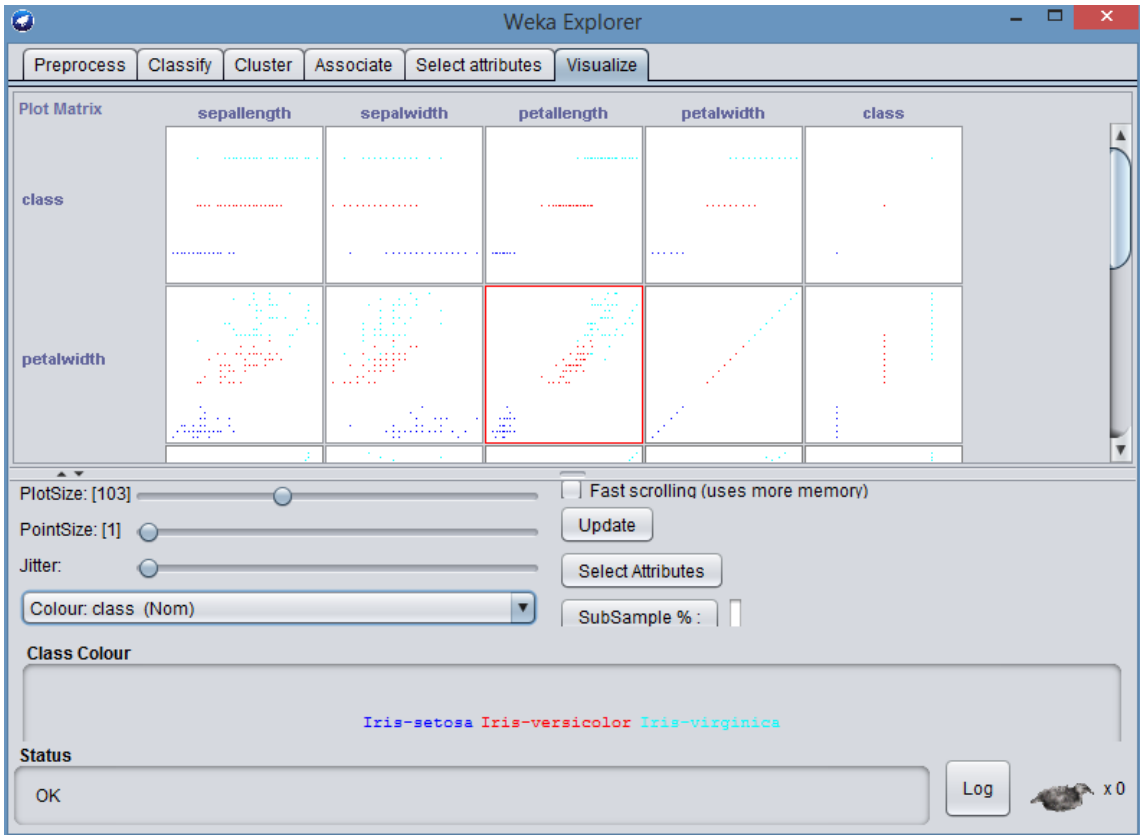



*Ilustración 97: Anexo I: Interfaz Visualize.*

A través de esta ventana, podemos apreciar todas las comparaciones gráficas posibles entre pares de atributos de nuestra base de datos, con la posibilidad de poder sacar conclusiones sobre algunas de ellas. Por ejemplo, en la ilustración 98, se podría observar que relaciones tendríamos entre dos variables de nuestro conjunto de datos.

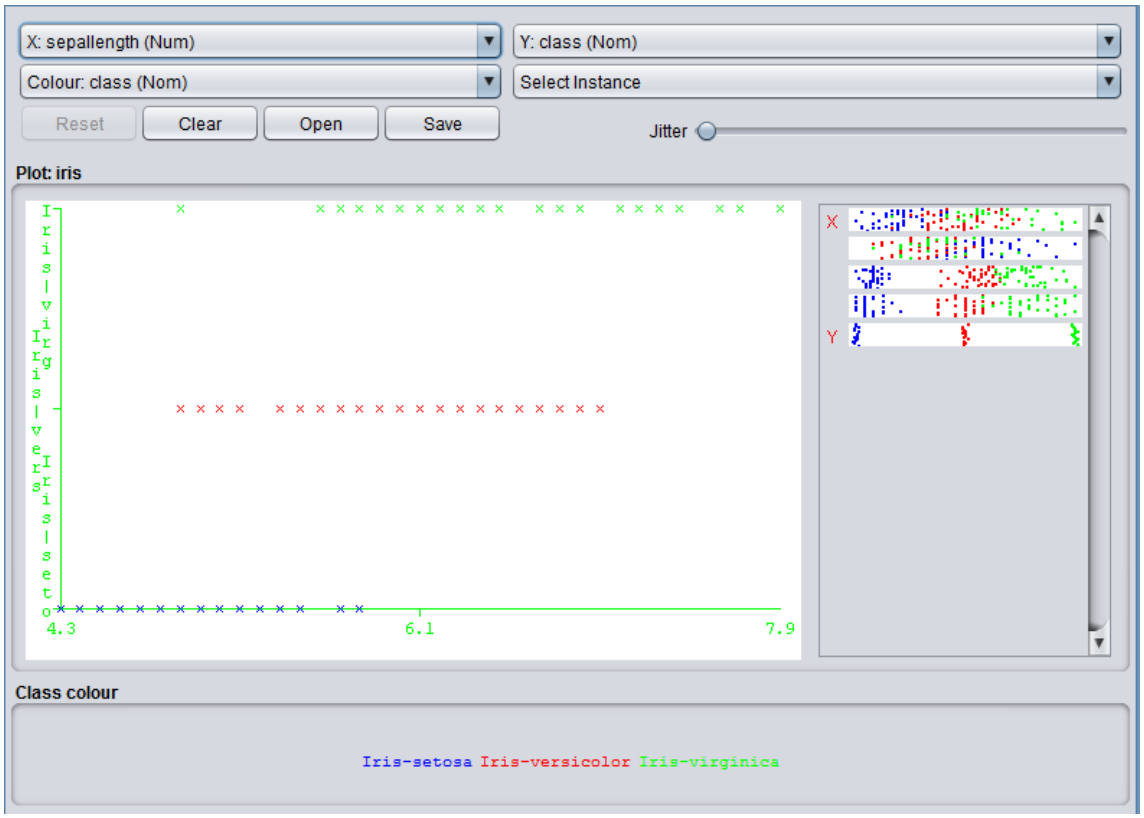



*Ilustración 98: Anexo I: Ejemplo de visualize*

## 2.3.Clasificación

Cuando estamos interesados en encontrar patrones de comportamientos entre los datos se recurre a la tarea de clasificación, que suele ser la más frecuente entre las realizadas en minería de datos. El objetivo será el de encontrar relaciones entre los atributos que permitan saber cuáles son las posibilidades de si una instancia de un conjunto de datos pertenece a una clase o a la contraria. Esta tarea se lleva a cabo con la pestaña *Classify* ilustración 99.

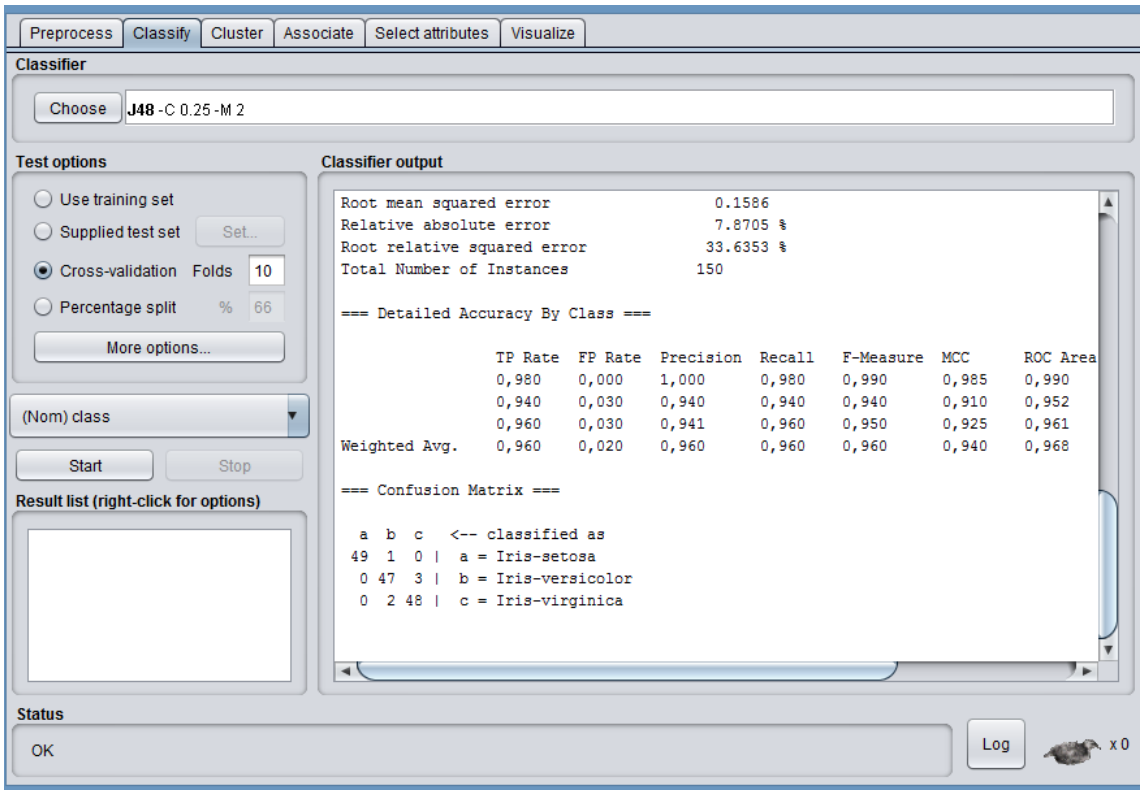



*Ilustración 99: Anexo I: Interfaz de classify.*

Al igual que en el apartado anterior, a través del botón *choose* se puede elegir el método de clasificación que queremos utilizar, entre los que se encuentran:

- **Bayes**: son métodos basados en el aprendizaje de Bayes, que son aquellos que intentan encontrar de entre todas las hipótesis la más probable, a partir de un conjunto de entrenamiento. El algoritmo más utilizado en este apartado es el de *NaiveBayes*.

- **Funciones**: se corresponden con los métodos que están basados en modelos matemáticos, como por ejemplo: las redes neuronales, o los diferentes tipos de regresiones.

- **Lazy**: en este tipo de algoritmos, cada una de las instancias se compara con el resto del conjunto de datos, definiéndose una "medida de distancia", son métodos donde el objetivo es "encontrar al vecino más cercano".

- **Meta**: son los métodos que se obtienen al combinar distintos tipos de aprendizaje.

- **Trees**: métodos expresados a través de árboles de decisión. En este caso se construye un árbol desde la raíz hasta las hojas, de tal manera que las ramas se dividen en función de los valores que toman los atributos. Entre todos ellos, el más popular es el *J48* que es una mejora del árbol inicial *C4.5* diseñado en 1945.

- **Rules**: son algoritmos que se expresan a través de reglas y que tienen la particularidad de ser autoaprendizajes.

Además de elegir el tipo de método a usar, en esta ventana también existe la posibilidad de elegir el tipo de validación del modelo, que puede ser:

- **Use training set**: con esta opción el programa utilizará el método elegido con todos los datos disponibles y luego realizará una evaluación sobre los mismos datos.

- **Supplied test set**: podemos realizar una evaluación sobre un conjunto de datos que hemos elegido previamente, que normalmente serán distintos a los datos del aprendizaje.

- **Cross-validation**: la evaluación se realizará mediante una técnica de validación cruzada, cuyo objetivo es asegurarse de que los análisis estadísticos realizados son independientes. De todas las posibilidades, esta opción es la que más tiempo computacional consume. Con *Folds* se puede elegir el número de evaluaciones que deseamos llevar a cabo, dividiendo el conjunto de datos en datos de prueba y datos de entrenamiento.

- **Percentage split**: en esta última opción podemos definir un porcentaje con el que aprende el modelo, haciéndose la evaluación con los datos restantes.

Como se ha comentado, el método que más se utiliza es el de árboles de decisión, por lo que realizaremos nuestro ejemplo a través de este algoritmo. Para ello pulsaremos sobre el botón *Choose* y entraremos en la carpeta *Trees* para seleccionar el *J48*.
Una vez pulsado el botón S*tart*, se ejecutará nuestro árbol de decisión. Si no ha habido problemas, el programa nos debe dar la información solicitada por medio de la ventana *Classifier Output*.

## 2.4.Resultado obtenido

Lo primero que nos daría con resultado la información sobre el tipo de algoritmo utilizado, el nombre del archivo, el número de atributos, sus nombres, el número de instancias y el modo del test realizado.

```
Scheme:       Weka.classifiers.trees.J48 -C 0.25 -
M 2
Relation:     iris
Instances:    150
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:    10-fold cross-validation
```

A continuación se facilita el resultado del clasificador. En nuestro caso se ha obtenido un árbol de decisión el cual es el siguiente:

```
petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
|   petalwidth <= 1.7
|   |   petallength <= 4.9: Iris-versicolor (48.0/1.0)
|   |   petallength > 4.9
|   |   |   petalwidth <= 1.5: Iris-virginica (3.0)
|   |   |   petalwidth > 1.5: Iris-versicolor (3.0/1.0)
|   petalwidth > 1.7: Iris-virginica (46.0/1.0)
```

Posteriormente se ofrece un resumen del test realizado, donde lo destacado para observar es el número de instancias que se ha clasificado correctamente así como si los niveles de errores son bajos.

| | | |
|---|---|---|
| Correctly Classified Instances | 144 | 96% |
| Incorrectly Classified Instances | 6 | 4% |
| Kappa statistic | 0.94 | |
| Mean absolute error | 0.035 | |
| Root mean squared error | 0.1586 | |
| Relative absolute error | 7.8705 % | |
| Root relative squared error | 33.6353 % | |
| Total Number of Instances | 150 | |

Como se puede observar en nuestro caso se ha realizado una clasificación muy aceptable ya que ha clasificado bien el 96% de las instancias con unos errores muy bajos.
La información global de la precisión obtenida también se ofrece para cada una de las clases del atributo clasificado. Para nuestro caso sería el siguiente resultado.

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|---|---|---|---|---|---|---|---|---|---|
| | 0,980 | 0,000 | 1,000 | 0,980 | 0,990 | 0,985 | 0,990 | 0,987 | Iris-setosa |
| | 0,940 | 0,030 | 0,940 | 0,940 | 0,940 | 0,910 | 0,952 | 0,880 | Iris-versicolor |
| | 0,960 | 0,030 | 0,941 | 0,960 | 0,950 | 0,925 | 0,961 | 0,905 | Iris-virginica |
| Weighted Avg. | 0,960 | 0,020 | 0,960 | 0,960 | 0,960 | 0,940 | 0,968 | 0,924 | |

Para finalizar, aparece la matriz de confusión. Se trata de una matriz de orden 3X3, donde en la fila podemos encontrar las instancias de cada clase y en las columnas aparece las predichas por el algoritmo, la matriz resultado para nuestro caso sería la siguiente:

```
a   b   c   <-- classified as
49  1   0 |  a = Iris-setosa
0   47  3 |  b = Iris-versicolor
0   2  48 |  c = Iris-virginica
```

Entonces podemos ver que en nuestro caso tenemos 3 instancias que han sido predichas por el algoritmo como “Iris-virginica” cuando de verdad pertenece a la clase “Iris-versicolor”.

El árbol de decisión si lo expresamos de la forma anterior puede llegar a ser muy confuso. Weka ofrece la opción de visualizarlo de manera gráfica. Para ello es necesario pulsar sobre el resultado *(Result list)* con el botón derecho del ratón y tendremos acceso al gráfico dándole *Visualize tree* el resultado para nuestro caso sería el de la ilustración 100.

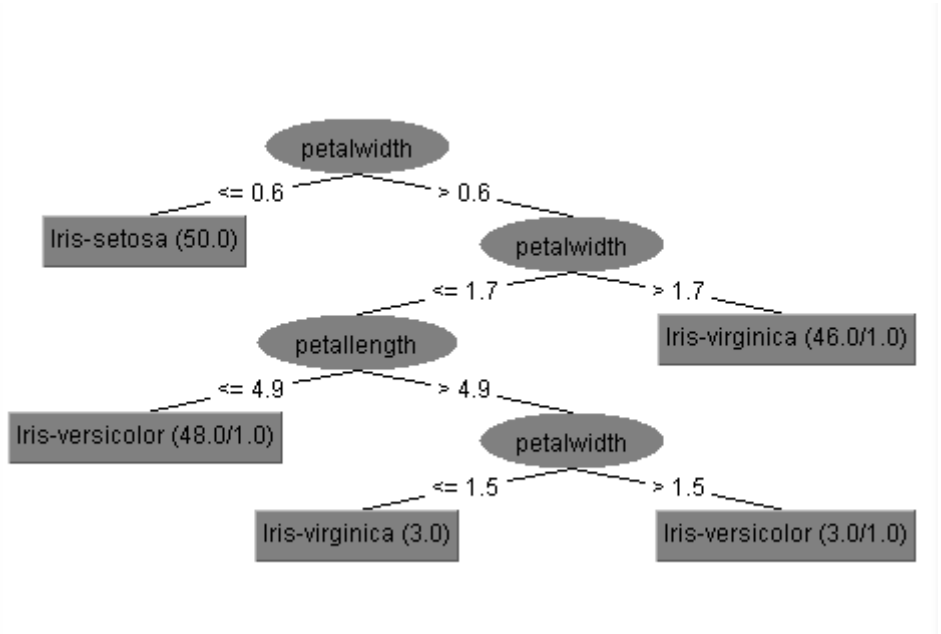



*Ilustración 100: Anexo I: Árbol resultado de J48.*

Entonces como podemos ver en la ilustración 100 que tenemos un árbol el cual es más intuitivo y fácil de entender, se podría decir que lo primero que miramos sería la variable *petalwidth* si su valor es de menos de 0.6 entonces directamente podríamos decir que es del tipo *"Iris-setosa"* sino deberíamos mirar la variable petalwidth que si es mayor de 1.7 entonces es del tipo *"Irirs-virginica"* y si no se mira la siguiente variable que muestra así seguimos hasta llegas a la última hoja del árbol.
Como hemos podemos pensar, si tenemos un número de atributos elevado se hace difícil conseguir un árbol que permitan unas reglas más sencillas. Podemos simplificar pero a costa de perder precisión. La pregunta clave entonces sería, **¿cuáles son los atributos que podemos eliminar al ser los que menos influencia ofrecen en la predicción?** El programa dispone de una opción que da respuesta a esta pregunta. Elegimos la pestaña *Select attibutes* y dejamos los mismos algoritmos de evaluación de atributos y el método de búsqueda que aparecen por defecto, a continuación elegimos como modo de selección de atributos el de validación cruzada y pulsamos el botón *Start*. El programa ofrece como respuesta el porcentaje correspondiente a cada uno de los atributos y podemos ver que con un valor del 100% se encuentra los atributos *petallength y petalwidth*.

Con esta información, volvemos a la pestaña inicial de *Preprocess,* marcamos los atributos que no han salido anteriormente y los borramos pulsando el botón *Remove.* Con este cambio volvemos a aplicar el algoritmo anterior el *J48* y obtenemos como resultado el siguiente:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Correctly Classified Instances | 144 | 96% | | | | | | | |
| Incorrectly Classified Instances | 6 | 4% | | | | | | | |
| Kappa statistic | 0.94 | | | | | | | | |
| Mean absolute error | 0.035 | | | | | | | | |
| Root mean squared error | 0.1586 | | | | | | | | |
| Relative absolute error | 7.8705 % | | | | | | | | |
| Root relative squared error | 33.6353 % | | | | | | | | |
| Total Number of Instances | 150 | | | | | | | | |
| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
| | 0,980 | 0,000 | 1,000 | 0,980 | 0,990 | 0,985 | 0,990 | 0,987 | Iris-setosa |
| | 0,940 | 0,030 | 0,940 | 0,940 | 0,940 | 0,910 | 0,952 | 0,880 | Iris-versicolor |
| | 0,960 | 0,030 | 0,941 | 0,960 | 0,950 | 0,925 | 0,961 | 0,905 | Iris-virginica |
| Weighted Avg. | 0,960 | 0,020 | 0,960 | 0,960 | 0,960 | 0,940 | 0,968 | 0,924 | |

Como podemos observar en nuestro caso a ser un conjunto de datos tan pequeño y tan optimizado el resultado al que llega es el mismo que si no fuéramos realizado ningún cambio, en otros casos fuéramos observado alguna mejoría.

## 2.5.Agrupamiento

Las técnicas de *Cluster* permiten realizar agrupamientos de las instancias de la base de datos basándose en las semejanzas y diferencias que existen entre los datos que componen la muestra.
Como ya tenemos cargado nuestro archivo, nos vamos a la pestaña *Cluster*, donde al igual que en las anteriores pestañas volvemos a pinchar en el botón *Choose* y escogeremos un algoritmo, entre los que se encuentra el de *SimpleKMeans (*ilustración 101), que es uno de los más utilizados por su sencillez.

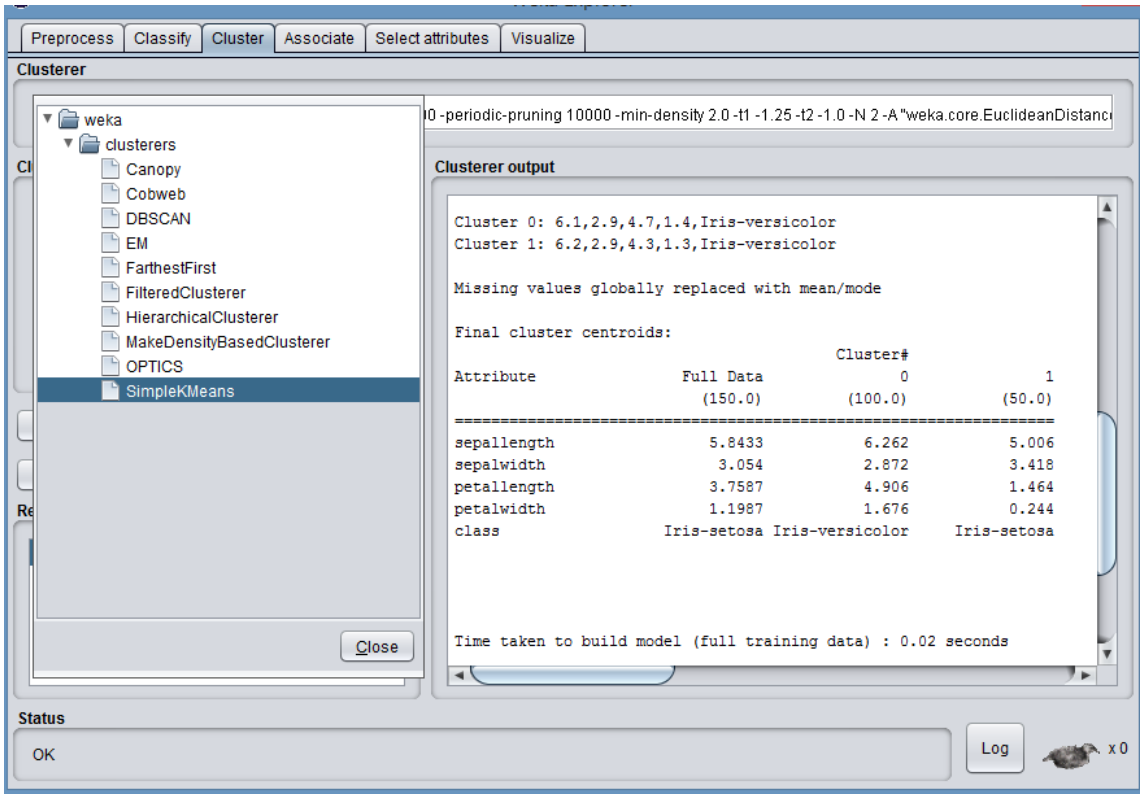



*Ilustración 101: Anexo I: Ilustración clúster.*

El paso siguiente sería el de seleccionar los atributos que queremos que entren en nuestro estudio, en nuestro caso todas las variables aportan algo de información o si no podríamos quitar la que comentamos anteriormente pero no tendríamos ningún cambio. Para eliminar los atributos deberíamos pinchar en *Ignore Atributtes* y seleccionamos los atributos a ignoras.

Procedemos de manera similar a como lo hicimos con la técnica de clasificación. En primer lugar, podemos modificar las propiedades del algoritmo pulsando con el ratón sobre su nombre, por ejemplo, se puede modificar el número K de clúster que deseamos hacer, de esta manera el programa seleccionará de forma aleatoria k instancias que representarán el centro de cada uno de los agrupamientos. A continuación seleccionamos como modo del clúster, *Use training set*, de esta manera usamos la misma muestra como entrenamiento y como comprobación del resultado obtenido. Finalmente, pulsaremos sobre el botón *start* para ejecutar el algoritmo.

Entonces si ejecutamos este algoritmo para nuestro caso obtenemos el siguiente resultado:

| | | Cluster# | |
|---|---|---|---|
| Attribute | Full Data | 0 | 1 |
| | (150.0) | (100.0) | (50.0) |
| sepallength | 5.8433 | 6.262 | 5.006 |
| sepalwidth | 3.054 | 2.872 | 3.418 |
| petallength | 3.7587 | 4.906 | 1.464 |
| petalwidth | 1.1987 | 1.676 | 0.244 |
| class | Iris-setosa | Iris-versicolor | Iris-setosa |

Entonces podemos ver que hemos obtenido un clúster formado por dos centros uno con 100 instancias y el otro con 50 de un total de 150. Donde se nos indica cuáles deben ser los valores de las variables para que una flor sea iris-versicolor o iris-setosa. Como en el caso anterior, existe la posibilidad de ver estos dos clústeres de forma gráfica a través del botón derecho pinchando en *Visualize Cluster Assignments*. Con esta pantalla podremos generar múltiples gráficas eligiendo cualquier tipo de combinación para cada uno de los ejes. Un ejemplo podría ser el que se observa en la ilustración 102.

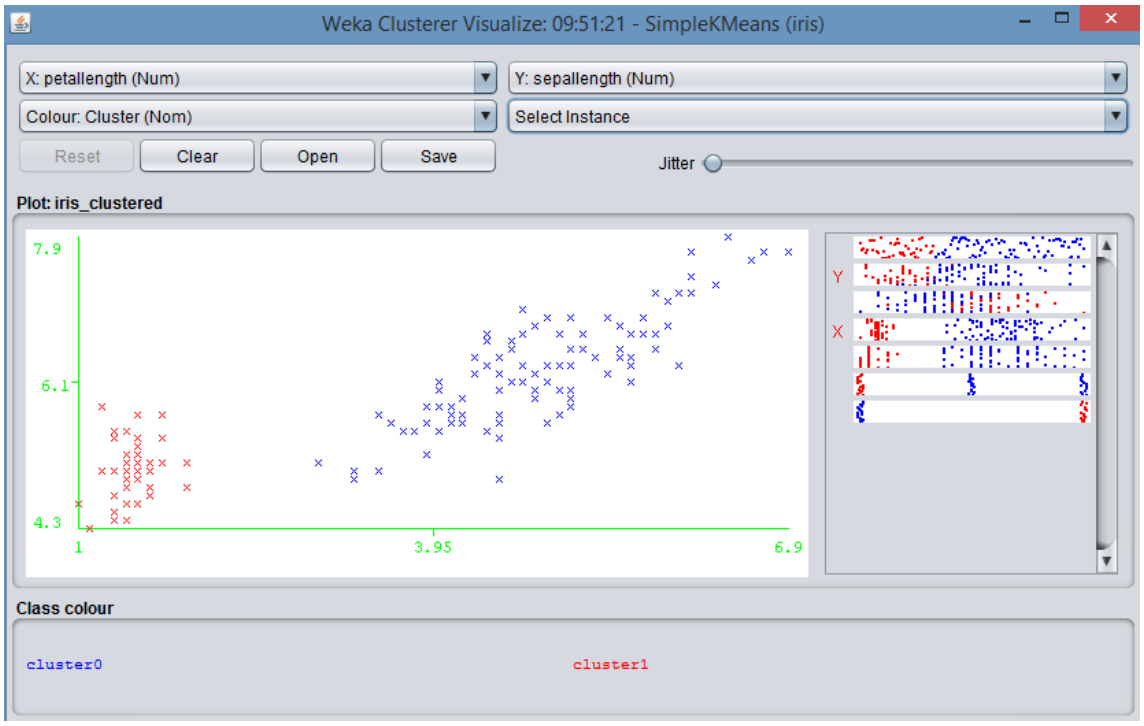



*Ilustración 102: Anexo I: Visualice de clasificador.*

## 2.6.Asociación de los datos

Otra de las pestañas, más sencilla de manejar ya que apenas tiene opciones, que podemos usar en WEKA es la de *Associate.* Con esta ventana se aplican los métodos y algoritmos para buscar asociaciones entre los datos. La forma de utilizarla es la siguiente: se selecciona el método, se configura y lo ejecutamos con el botón *Start.*
Pero es importante saber que para poder hacer uso de unos de los algoritmos más populares, este es el algoritmo *Apriori*, previamente se tiene que preprocesar los atributos para poderlos discretizar, ya que en caso contrario el programa no te permite seleccionar esta opción.
Esto último se hace de la siguiente manera, primero nos vamos a la pestaña *Preprocess* y elegimos en los filtros el siguiente *unsupervised/attribute/Discretize,* con esto conseguimos discretizar los datos y nos da entonces la opción de aplicar la asociación dándonos el siguiente resultado.

Apriori

=======

Minimum support: 0.1 (15 instances)

Minimum metric <confidence>: 0.9

Number of cycles performed: 18

Generated sets of large itemsets:

Size of set of large itemsets L(1): 20

Size of set of large itemsets L(2): 15

Size of set of large itemsets L(3): 3

Best rules found:

1. petalwidth='(-inf-0.34]' 41 ==> class=Iris-setosa 41 <conf:(1)> lift:(3) lev:(0.18) [27] conv:(27.33)

2. petallength='(-inf-1.59]' 37 ==> class=Iris-setosa 37 <conf:(1)> lift:(3) lev:(0.16) [24] conv:(24.67)

3. petallength='(-inf-1.59]' petalwidth='(-inf-0.34]' 33 ==> class=Iris-setosa 33 <conf:(1)> lift:(3) lev:(0.15) [22] conv:(22)

4. petalwidth='(1.06-1.3]' 21 ==> class=Iris-versicolor 21 <conf:(1)> lift:(3) lev:(0.09) [14] conv:(14)

5. petallength='(5.13-5.72]' 18 ==> class=Iris-virginica 18 <conf:(1)> lift:(3) lev:(0.08) [12] conv:(12)

6. sepallength='(4.66-5.02]' petalwidth='(-inf-0.34]' 17 ==> class=Iris-setosa 17 <conf:(1)> lift:(3) lev:(0.08) [11] conv:(11.33)

7. sepalwidth='(2.96-3.2]' class=Iris-setosa 16 ==> petalwidth='(-inf-0.34]' 16 <conf:(1)> lift:(3.66) lev:(0.08) [11] conv:(11.63)

8. sepalwidth='(2.96-3.2]' petalwidth='(-inf-0.34]' 16 ==> class=Iris-setosa 16 <conf:(1)> lift:(3) lev:(0.07) [10] conv:(10.67)

9. petallength='(3.95-4.54]' 26 ==> class=Iris-versicolor 25 <conf:(0.96)> lift:(2.88) lev:(0.11) [16] conv:(8.67)

10. petalwidth='(1.78-2.02]' 23 ==> class=Iris-virginica 22 <conf:(0.96)> lift:(2.87) lev:(0.1) [14] conv:(7.67)