



TEC

Tecnológico de Costa Rica

ESCUELA DE INGENIERÍA EN COMPUTACIÓN

PROGRAMA DE MAESTRÍA EN COMPUTACIÓN

---

# Evaluación de Métodos Agregados de Aprendizaje de Máquina: Aplicación sobre datos de la misión Kepler

---

Tesis

Presentada en cumplimiento parcial de los requisitos para el grado de  
*Magister Scientiæ* en Ingeniería en Computación con énfasis en  
Ciencias de la Computación

Autor

Sergio Morales Esquivel

Asesor

Francisco J. Torres Rojas, Ph.D.

diciembre 2016



*Esta tesis está dedicada a todas las personas que me han hecho llegar hasta acá; mis padres Luis y Nuria, mi hermano Andrés, mi compañera y mejor amiga Andrea y un sinnúmero de profesores, mentores y amigos.*

# Agradecimientos

Quiero agradecer a mi profesor asesor, Francisco Torres, por el apoyo que me ha mostrado a lo largo de mis estudios, no sólo en calidad de supervisor para la elaboración de esta investigación, si no también como profesor y amigo a lo largo de la mayor parte de mi carrera universitaria.

Agradezco además a mis lectores y el resto del comité de tesis por sus invaluable comentarios y retroalimentación durante todo el proceso de realización de esta investigación, a Jeudy Blanco por su guía y aportes en las etapas de planeación, a Szymon Wojciechowski por la implementación del algoritmo MODLEM para Weka utilizada en la fase experimental y al Colaboratorio Nacional de Computación Avanzada (CNCA) por el uso de su cluster para la ejecución del experimento.

Finalmente, deseo agradecerle a toda mi familia y amigos el apoyo y paciencia que me han ofrecido a lo largo de este proyecto.

## **Abstract**

Numerous tools and techniques have been used to find and study extrasolar planets, but none has been more successful than NASA's Kepler Space Telescope, which has discovered the majority of known exoplanets. Not only that, but the mission's data has been made available in an open format, allowing for independent efforts to not only find new exoplanet candidates.

Data analysis on the gathered samples is still an ongoing process, and the use of specialized techniques that may improve the confidence level for both as-of-now unconfirmed and new findings is of high interest to the scientific community. In particular, the data gathered by the Kepler mission is known to have significant levels of noise, both intrinsic to the stars and arising from limitations such as instrument noise. Classic machine learning algorithms are limited in the sense that the existence of noise requires more robust preprocessing stages in the candidate identification pipeline.

It is the purpose of this research to apply ensemble machine learning methods based on rule-induction classifiers on data retrieved by the Kepler project, and measure their success in comparison with more conventional machine learning methods. Experimentation on various factors related to said machine learning methods revealed significant differences between both classic classification models and rule-induction classifiers as well as effects of ensemble machine learning on the classification accuracy of said data.

## **Resumen**

Numerosas herramientas y técnicas han sido utilizadas para encontrar y estudiar planetas extrasolares, pero ninguna ha sido más exitosa que el proyecto Kepler de NASA, el cual ha descubierto la mayoría de los planetas extrasolares conocidos. No solo eso, la misión ofrece de forma abierta sus datos, lo que permite que esfuerzos independientes contribuyan a la búsqueda de candidatos.

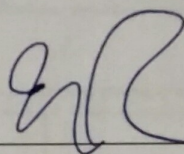
El análisis de datos sigue en progreso, y técnicas que puedan mejorar el nivel de confianza de candidatos son de gran interés para la comunidad científica. En particular, los datos son conocidos por sus considerables niveles de ruido, intrínsecos a la naturaleza de las estrellas, y derivados de limitantes fundamentales como el ruido propio de los instrumentos. A pesar del éxito de enfoques clásicos de aprendizaje de máquina para la identificación de candidatos, se encuentran limitados por la existencia de dicho ruido, requiriendo un mayor esfuerzo en las etapas de pre-procesamiento para minimizar su impacto.

El presente proyecto propone la aplicación de enfoques agregados de aprendizaje de máquina basados en clasificadores de inducción de reglas sobre datos recolectados por la misión Kepler, y su análisis en comparación con métodos tradicionales de aprendizaje de máquina supervisada. El proceso de experimentación sobre varios factores relacionados tanto con dichos enfoques de aprendizaje así como el uso de métodos agregados revelaron diferencias significativas en cuanto a la efectividad de clasificación de los datos estudiados.

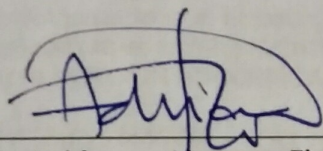
## APROBACIÓN DE LA TESIS

### "Evaluación de Métodos Agregados de Aprendizaje de Máquina: Aplicación sobre datos de la misión Kepler"

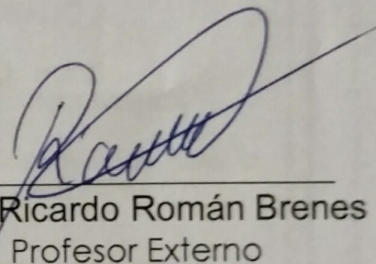
#### TRIBUNAL EXAMINADOR



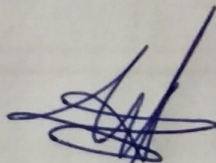
Dr. Francisco Torres Rojas  
Profesor Asesor



MGP. Adriana Álvarez Figueroa  
Profesor Lector



M.Sc. Ricardo Román Brenes  
Profesor Externo



Dr. Roberto Cortés Morales  
Coordinador del Programa  
de Maestría en Computación

Agosto, 2016

---

## Contenido

---

<b>1</b>	<b>Introducción y Antecedentes</b>	<b>1</b>
1.1	Introducción . . . . .	2
1.2	Métodos de aprendizaje de máquina agregados . . . . .	3
1.3	Algoritmos de Inducción de Reglas . . . . .	8
1.4	La misión Kepler . . . . .	11
1.4.1	Pipeline de Datos . . . . .	13
1.5	Planteamiento del Problema . . . . .	17
1.6	Propuesta de Investigación . . . . .	19
1.7	Hipótesis . . . . .	19
1.8	Trabajo Relacionado . . . . .	20
<b>2</b>	<b>Objetivos y Aportes</b>	<b>21</b>
2.1	Objetivo general . . . . .	22
2.2	Objetivos específicos . . . . .	22
2.3	Alcances y Limitaciones . . . . .	23
<b>3</b>	<b>Experimentación</b>	<b>25</b>
3.1	Factores y niveles . . . . .	27
3.1.1	Métodos de Clasificación . . . . .	28
3.1.2	Volumen de Datos . . . . .	28
3.1.3	Atributos . . . . .	29
3.1.4	Proporción entrenamiento-prueba . . . . .	29
3.2	Otras Consideraciones . . . . .	29
3.3	Conjunto de datos . . . . .	30
3.3.1	Transformaciones sobre el conjunto de datos . . . . .	32
3.4	Variables de respuesta . . . . .	33
3.5	Análisis de varianza . . . . .	34
3.6	Implementación de Random MODLEM . . . . .	35
3.7	Herramientas y Ejecución . . . . .	37

<b>4</b>	<b>Resultados y Análisis</b>	<b>40</b>
4.1	Verificación de supuestos . . . . .	42
4.2	Análisis de resultados . . . . .	44
4.3	Modelo de Clasificación . . . . .	51
<b>5</b>	<b>Conclusiones y Trabajo Futuro</b>	<b>53</b>
5.1	Aportes . . . . .	56
5.2	Trabajo futuro . . . . .	57
	<b>Referencias</b>	<b>58</b>
<b>A</b>	<b>Código Auxiliar</b>	<b>64</b>
<b>B</b>	<b>Código MODLEM</b>	<b>76</b>
<b>C</b>	<b>Muestra de Datos</b>	<b>83</b>
<b>D</b>	<b>Muestra de Modelo de Clasificación MODLEM</b>	<b>87</b>
<b>E</b>	<b>Artículo AAAI-17</b>	<b>96</b>



---

## Lista de figuras

---

1.1	Los métodos de aprendizaje de máquina agregados consisten en la combinación de varios clasificadores base para la generación de un único algoritmo de aprendizaje fuerte.[12] . . . . .	3
1.2	Representación gráfica del espacio de búsqueda en problemas de clasificación, en el que el nodo a buscar es una función de clasificación perfecta.[14] .	6
1.3	Dado el ejemplo mostrado anteriormente, un árbol de decisión presentaría subárboles redundantes para tres valores posibles en cada una de las variables de clasificación.[44] . . . . .	9
1.4	Pseudocódigo para la generación de reglas por medio de un algoritmo de inducción . . . . .	10
1.5	El volumen de búsqueda de Kepler, en el contexto de la Vía Láctea. [54]	13
1.6	Diagrama resumen del pipeline de datos de la información recolectada por Kepler, desde su recopilación por medio del instrumento hasta su análisis. [47] . . . . .	14
1.7	Curva de luz perteneciente al planeta extrasolar identificado ahora como Kepler-5b. Se puede notar su periodicidad en los niveles de luz decrecientes de manera periódica en esta representación.[27] . . . . .	16
1.8	Histograma representando la cantidad por año de planetas extrasolares descubiertos hasta Septiembre 2014.[35] . . . . .	17
3.1	Fragmento de la salida de un clasificador en Weka. Se pueden identificar datos importantes como el número y porcentaje de instancias clasificadas correcta e incorrectamente, la matriz de confusión y un bloque detallando la precisión por categoría. . . . .	39
4.1	Exploración visual de la normalidad y homogeneidad de varianza en los datos de tiempo sin transformaciones. . . . .	43

4.2	Invocación de la función <code>aov</code> para Análisis de Varianza en R. El parámetro recibido muestra las relaciones entre las variables: <code>accu(t)</code> (precisión de clasificación) la variable de respuesta se debe analizar en función de los factores y todas sus combinaciones. . . . .	44
4.3	Efecto de los factores principales del experimento. . . . .	46
4.4	Efecto de las combinaciones de los factores de segundo nivel en el experimento, Clasificadores-Volumen. . . . .	47
4.5	Efecto de las combinaciones de los factores de segundo nivel en el experimento, Clasificadores-Atributos. . . . .	48
4.6	Efecto de las combinaciones de los factores de segundo nivel en el experimento, Volumen-Atributos. . . . .	48
4.7	Efecto de las combinaciones de los factores de segundo nivel en el experimento, Clasificadores-Proporción. . . . .	49
4.8	Efecto de las combinaciones de los factores de segundo nivel en el experimento, Volumen-Proporción. . . . .	49
4.9	Efecto de las combinaciones de los factores de segundo nivel en el experimento, Atributos-Proporción. . . . .	50
4.10	Modelo de clasificación producido por MODLEM durante una corrida experimental . . . . .	52

---

## Introducción y Antecedentes

---

‘Our contemplations of the cosmos stir us. There is a tingling in the spine, a catch in the voice, a faint sensation, as if a distant memory of falling from a great height.

We know we are approaching the grandest of mysteries’.

Carl Sagan.

# 1.1 Introducción

El número de esfuerzos actuales para el descubrimiento de planetas extrasolares comprende proyectos tanto de agencias espaciales al rededor del mundo como de investigaciones independientes [53], sin embargo ninguno ha resultado tan exitoso como el proyecto Kepler de la NASA. Durante 4 años, el telescopio Kepler registró curvas de luz provenientes de más de 145,000 estrellas en una región cercana de nuestra galaxia [54]; observaciones temporales de niveles de luz, esperando encontrar indicios de la existencia de planetas en dichas observaciones. Desde entonces, esfuerzos tanto de la NASA como independientes en algoritmos de clasificación y aprendizaje de máquina han llevado al descubrimiento de más de 900 exoplanetas, con más de 3000 candidatos aún sin confirmar [11].

El proceso de manejo de la información recolectada es complejo, y debido a la naturaleza de las observaciones, propenso a anomalías, lo que dificulta el proceso de confirmación de candidatos [21]. Entre dichas anomalías se encuentra el ruido existente en las curvas de luz recolectadas por la misión. El comportamiento de las estrellas, así como el equipo en sí poseen grados esperados de ruido, y es actualmente de gran interés elaborar procesos que lo minimicen tal que el proceso de clasificación pueda considerarse más confiable, y se reduzca el número de potenciales falsos positivos [29].

La actual investigación muestra el resultado de la aplicación de una familia de algoritmos de aprendizaje de máquina conocidos como de “inducción de reglas” [44] para la clasificación de datos multidimensionales y ruidosos, ambas características asociadas con las muestras de la misión Kepler, y la medición de su influencia sobre su capacidad de clasificación, siendo aplicados como algoritmos base para la comparación de métodos de agregación.

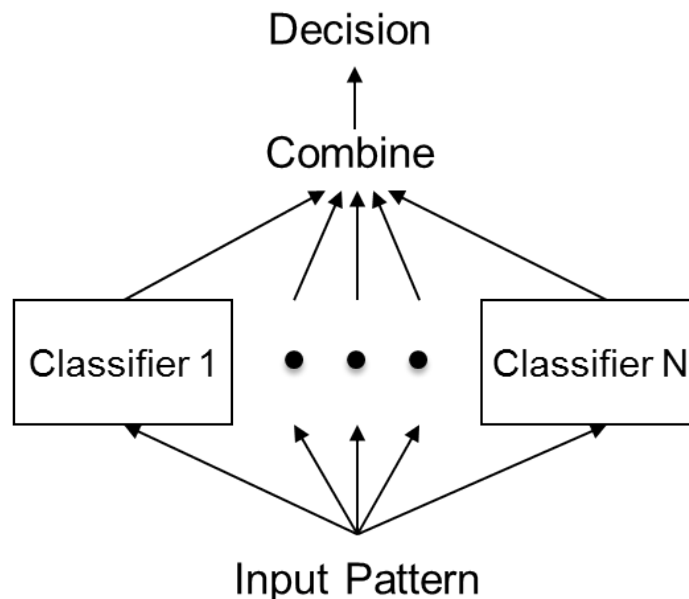
El presente documento se divide de la siguiente forma: Primeramente, se provee un resumen de los conceptos relevantes al proyecto experimental realizado, la hipótesis del mismo y sus objetivos. Posterior a esto, se discute la metodología experimental empleada, antes de continuar con un análisis detallado de los resultados obtenidos y

conclusiones asociadas.

## 1.2 Métodos de aprendizaje de máquina agregados

Uno de los enfoques que más ha ganado popularidad de manera reciente en el área del aprendizaje de máquina es el de “*ensemble machine learning*”, o aprendizaje agregado, siendo usado de manera reciente en proyectos de investigación en áreas como la bioinformática [44], o productos comerciales como motores de recomendación de películas [19].

Dicho enfoque consiste en la obtención de mejores resultados de clasificación en escenarios donde el aprendizaje supervisado es normalmente aplicado, por medio de la aplicación de diversas instancias de un mismo método de aprendizaje, o múltiples métodos de aprendizaje, para generar un clasificador que agrupa los parámetros y métodos de toma de decisión empleados por dichos diversos enfoques [52].



**Figura 1.1:** Los métodos de aprendizaje de máquina agregados consisten en la combinación de varios clasificadores base para la generación de un único algoritmo de aprendizaje fuerte.[12]

Existen diversos motivos detrás del enfoque múltiple o agregado de dichas técnicas. Conjuntos de datos multidimensionales contienen muchas veces cantidades significativas de parámetros que pueden ser tomados en cuenta por un clasificador para catalogar elementos en el conjunto de prueba luego de ser entrenado. Dados dichos parámetros, un único clasificador puede en este sentido formular una hipótesis de clasificación, como es el caso de un simple árbol de decisión, basado en una secuencia lineal de decisiones, sin embargo esto puede llevarlo a obviar otros parámetros o ignorar condiciones de clasificación más complejas [48]. Al crear diversos clasificadores, los métodos agregados pretenden crear un “clasificador fuerte” a partir de varios “clasificadores débiles” [52], intentando obviar dichas limitaciones. Un ejemplo de esto es el método conocido como “*Random Forest*”.

Un efecto secundario de este enfoque consiste en el alcance de un nivel significativamente mayor de resiliencia al ruido en los datos de entrada con respecto al encontrado en algoritmos de aprendizaje supervisado tradicionales, al existir diversos clasificadores actuando sobre los mismos conjuntos de datos. Esto es cierto en especial cuando los datos son multi-dimensionales y poseen grados de ruido significativos [44].

Thomas G. Diettrich [14] propone 3 razones principales por las cuales esta clasificación cumple las propiedades anteriormente mencionadas de mayores niveles de acierto y resiliencia al ruido con respecto a métodos tradicionales:

1. Un algoritmo de aprendizaje puede ser visto como un problema de búsqueda en un espacio de hipótesis que busca encontrar la mejor para clasificar, lo que lleva a problemas cuando no hay suficientes datos para elegir un buen clasificador. Emplear diversos métodos por lo tanto puede crear diferentes búsquedas dentro del espacio de hipótesis, y al promediar sus resultados, es posible reducir el riesgo de elegir finalmente un clasificador deficiente [6].

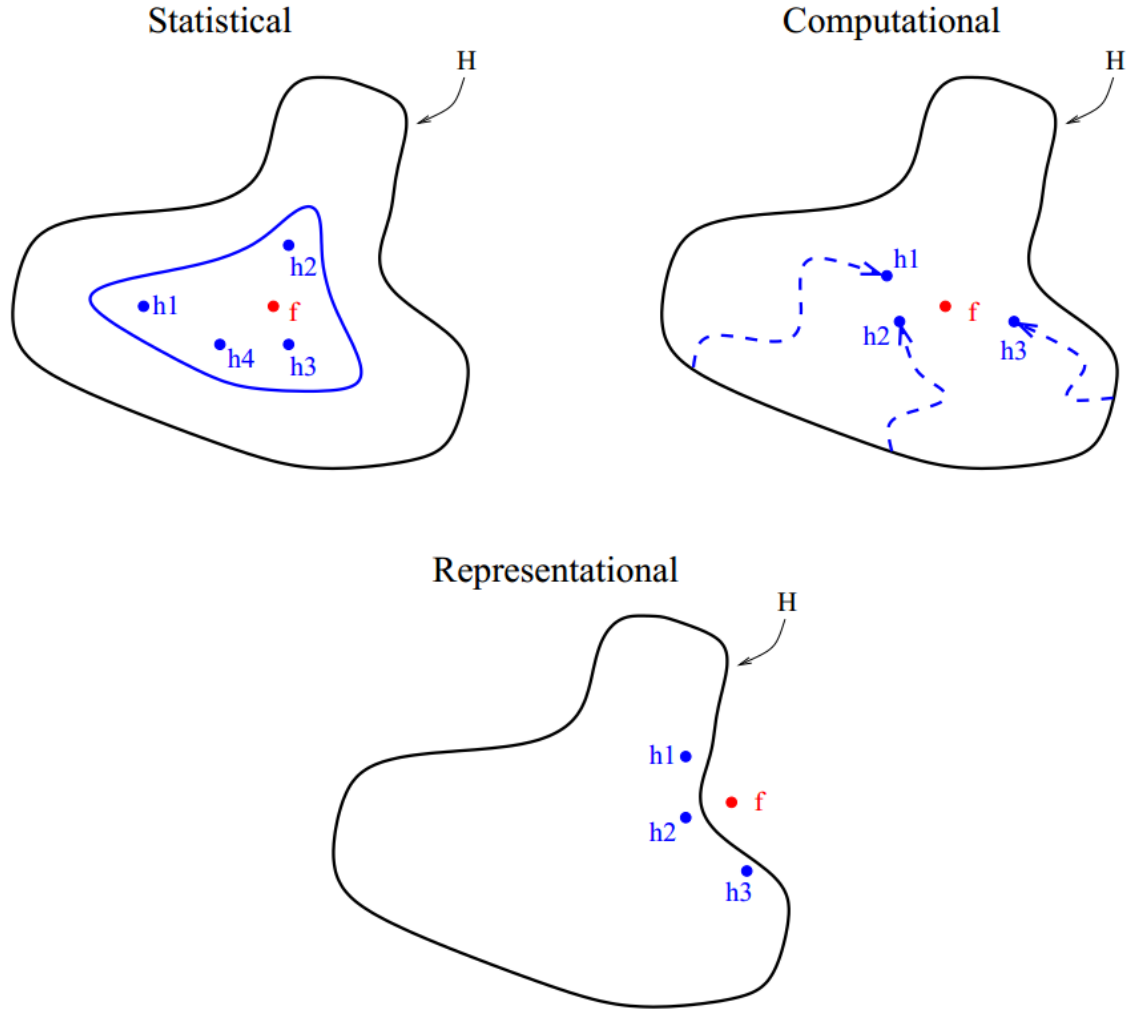
2. Al existir una búsqueda dentro de un espacio de hipótesis, se corre el riesgo de que un solo algoritmo de aprendizaje se atore en un óptimo local. Múltiples clasificadores pueden buscar a partir de distintos puntos dentro de dicho espacio, minimizando dicha posibilidad.
3. De manera empírica, es común que la función de clasificación que el clasificador busca aproximar no pueda ser representada de manera exacta por el mismo, pero pueda ser mejor aproximada por un conjunto de clasificadores en vez de sólo uno [14].

A continuación se presenta un breve resumen de varios enfoques generales de métodos de “ensemble machine learning”, los cuales buscan capturar las propiedades previamente expuestas por medio de la aplicación de varios algoritmos de aprendizaje, a través de distintos métodos generales.

### **Boosting**

*Boosting* consiste en un enfoque de aprendizaje de máquina que busca reducir los máximos locales durante la construcción de un clasificador para un conjunto de datos [9]. El meta-algoritmo se encuentra basado en una pregunta planteada por el investigador Michael Kearns en 1988, la cual cuestionaba la posibilidad de emplear un conjunto de clasificadores débiles para crear un único clasificador fuerte [24].

Este efecto es logrado por medio del “mejoramiento” iterativo de un clasificador inicial que es utilizado como base. Este clasificador es entrenado utilizando los datos de entrenamiento y luego, basándose en su posterior capacidad de clasificación sobre datos de prueba, un segundo clasificador se le es agregado, el cual se enfoca en aquellas instancias en las que el primero haya fallado. Este proceso es repetido hasta que se llegue a un límite impuesto por una medida de profundidad o de certeza en la capacidad de clasificación [9].



**Figura 1.2:** Representación gráfica del espacio de búsqueda en problemas de clasificación, en el que el nodo a buscar es una función de clasificación perfecta.[14]

Entre diferentes algoritmos que emplean este enfoque, su principal punto de variación es la manera en la que combinan los pesos asignados a cada punto de entrenamiento en cada sucesión. *AdaBoost* es significativo históricamente por ser el primero en dicha clase en tener propiedades adaptativas, sin embargo existen algoritmos más recientes como *LPBoost*, *TotalBoost*, *BrownBoost* y otros que se enfocan en diferentes especializaciones del algoritmo base [43].



### Bagging

El enfoque de *Bootstrap Aggregating*, también referido como *Bagging*, consiste en un método que separa diferentes muestras del conjunto de datos provisto para el entrenamiento del clasificador final, y crea un algoritmo individual para cada una de estas muestras. Los resultados de estos múltiples sub-clasificadores son posteriormente combinados, de tal forma que se calcule un nuevo valor promediado en cada punto de decisión, o bien se siga una regla de mayoría sobre los clasificadores que lo consideren. Este enfoque busca que cada clasificador provea una perspectiva diferente del problema, al enfocarse en una muestra única y por lo tanto distinta de todas las demás[9].

A diferencia del enfoque visto anteriormente, en el enfoque de *bagging*, cada uno de los clasificadores posee un voto de igual peso; no es un proceso iterativo en el cual los clasificadores posean un orden inherente que modifique el peso de decisión que tienen sobre el clasificador final[52].

El algoritmo de *Random Forest* es un ejemplo de un enfoque de este tipo. Propuesto por Leo Breiman y Adele Cutler, opera por medio de la construcción de múltiples árboles de decisión basados en sub-segmentos derivados a partir de los datos de entrenamiento provistos para el clasificador, de tal forma que para cada ejecución del mismo, se calcule la moda estadística de cada una de las clases reportadas por cada uno de estos árboles individuales. Los parámetros elegidos por estos árboles de decisión son aleatorios, con ajustes para controlar la varianza [7].

### Stacking

El método de *stacking*, también conocido como *blending*, involucra de manera principal el entrenar un algoritmo de aprendizaje para combinar el resultado de múltiples algoritmos de clasificación entrenados sobre el mismo conjunto de entrenamiento. De manera intuitiva, su mayor diferencia con respecto a los otros métodos consiste en que el meta-clasificador construido como producto final de este enfoque es generado a partir

no de diferentes instancias del mismo método como en los casos vistos anteriormente, sino a partir de métodos que pueden variar de manera significativa entre ellos [9].

Este método es particularmente volátil en cuanto a la mejora que puede llegar a presentar sobre los métodos que componen el clasificador final, sin embargo en la mayoría de los casos un clasificador que lo implemente puede garantizar ser mejor que cualquiera de los métodos de clasificación que lo integran de manera individual [55].

Al involucrar métodos de aprendizaje de todo tipo, no existen ejemplos concretos de implementaciones de este enfoque que no varíen con respecto a los métodos empleados tanto para generar el meta-clasificador final, así como para que este le asigne valor a los resultados de cada uno de los resultados y los combine, sin embargo con respecto a la segunda tarea, es común el empleo de métodos de regresión logística, un modelo de clasificación estadística probabilístico [5].

A continuación, se presenta una descripción del comportamiento general de algoritmos de inducción de reglas, sobre los cuales la presente investigación se enfoca.

## 1.3 Algoritmos de Inducción de Reglas

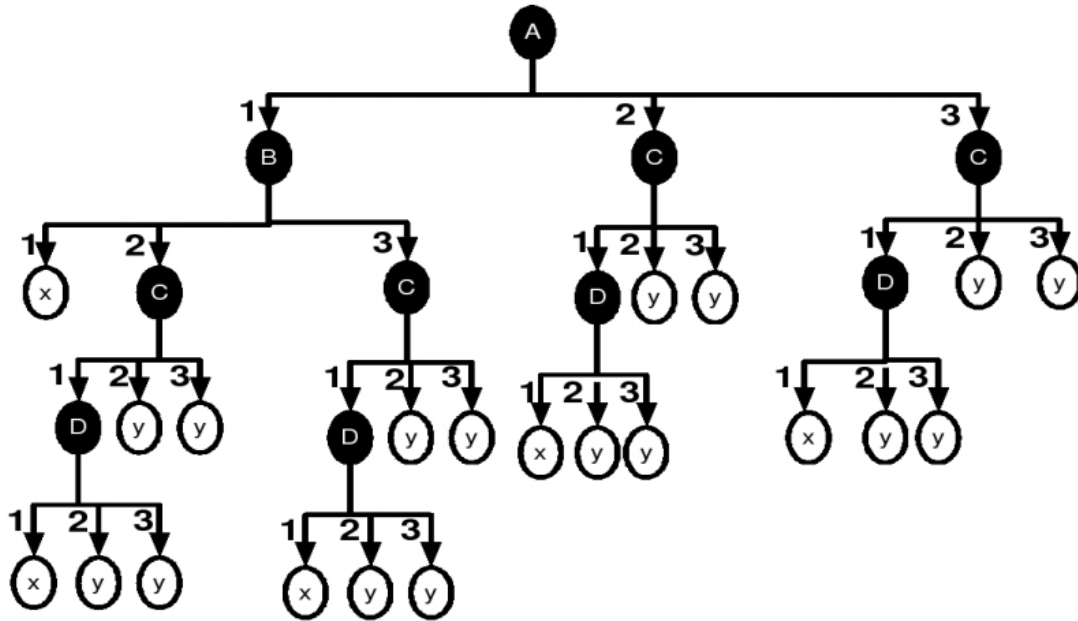
La inducción de reglas se basa en la extracción de patrones y reglas generales a partir de observaciones de características específicas. Existen dos enfoques generales para la inducción de reglas de clasificación: la de “divide y vencerás”, y la de “separa y vencerás”. La primera induce reglas de clasificación en la representación de un árbol de decisión, mientras que la segunda intuye un conjunto de reglas de formato *IF..THEN* en vez de un árbol de decisión. La familia de algoritmos de inducción de reglas pertenece a este segundo enfoque. Dichas reglas mejoran el acierto de clasificación para datos con altos niveles de ruido. El trabajo de Bramer, M. y Stahl, F. [44] busca crear un clasificador *ensemble* empleando como clasificadores base aquellos inspirados en esta variedad de algoritmo, inspirándose en la implementación clásica de *Random Forest*.

La representación de reglas en el enfoque de “separa y vencerás” son modulares, y pueden ser representadas de la siguiente forma:

*IF A = 1 AND B = 1 THEN class = x*

*IF C = 1 AND D = 1 THEN class = x*

El forzar reglas como estas en una estructura de árbol sería redundante y confusa debido a que generaría sub-árboles duplicados para tan pocos como 3 posibles valores [44].



**Figura 1.3:** Dado el ejemplo mostrado anteriormente, un árbol de decisión presentaría subárboles redundantes para tres valores posibles en cada una de las variables de clasificación.[44]

Los algoritmos de inducción evitan este problema usando conjuntos modulares de reglas, ignorando la estructura de árbol clásica. En este enfoque un objeto “conjunto de reglas” es construido generando de manera iterativa las mejores heurísticas de clasificación sobre un subconjunto de los datos de entrenamiento:

```
Rule_set rules = new Rule_set()
while Stopping_Criterion not satisfied:
    Rule = Learn_Rule()
    Remove all data instances covered from Rule
    rules.add(Rule)
```

**Figura 1.4:** Pseudocódigo para la generación de reglas por medio de un algoritmo de inducción

Luego de introducir cada regla, todas las instancias cubiertas por la misma son eliminadas, y una nueva regla es creada a partir del conjunto restante de casos no cubiertos. Un algoritmo de inducción básico intenta cubrir en cada paso la clase que contenga menos instancias dentro del conjunto de entrenamiento.

El método de *Random Prism* propuesto por Bramer [44] integra dentro de este algoritmo básico las características de aleatorizar la selección de características en el espacio de búsqueda mencionadas anteriormente en la descripción del algoritmo de Random Forest. A diferencia de este, sin embargo, no se emplea un sistema de votación [8] para el establecimiento del resultado de clasificación, sino que se añade un sistema de pesos en el que cada voto es influenciado por el peso de cada clasificador participante, el cual a su vez es generado a partir del nivel de aciertos del mismo sobre datos de prueba.

Dicho método posee sin embargo significativas limitaciones en cuanto al tipo de atributos soportados por el clasificador; al solo admitir atributos nominales. Existen, sin embargo, clasificadores similares como MODLEM, propuesto por Stefanowski [45], el cual de igual manera genera una lista de reglas de clasificación, ordenadas en orden de prioridad.

MODLEM se encuentra basado en la idea de recubrimiento secuencial, e intenta generar un conjunto mínimo de reglas para cada clase, el cual cubre todos los ejemplos positivos de dicha clase, sin cubrir ningún ejemplo negativo [16]. Este realiza una búsqueda de todos los atributos elementales que se encuentran en las muestras, y una vez creada una regla que los describa, remueve dichos ejemplos del conjunto total, de manera muy similar a los esfuerzos de Brahm descritos anteriormente.

Los atributos numéricos son manejados durante el proceso de inducción de reglas, y son representados por relaciones de igualdad o desigualdad ( $a < b$ ) o ( $a \geq b$ ) entre los valores encontrados en las muestras que componen el conjunto de entrenamiento durante el proceso de aprendizaje supervisado [45][16].

## 1.4 La misión Kepler

Kepler es un observatorio espacial lanzado por NASA el 7 de Marzo del 2009, luego de haber sido seleccionado como el Discovery Mission #10 en Diciembre del 2001. Recolectó información en forma de curvas de luz de más de 100,000 estrellas de manera continua hasta Mayo del 2013, cuando desperfectos mecánicos frenaron la recolección de datos. Las principales metas científicas del proyecto son [3]:

- Determinar la frecuencia de planetas con un radio igual o mayor a un 80 % al de la Tierra, dentro o cerca de la zona habitable de diversos tipos de estrellas.
- Determinar la distribución de tamaños y ejes principales de órbita de dichos planetas.
- Estimar la frecuencia de planetas orbitando sistemas con múltiples estrellas.
- Estimar la distribución de ejes, excentricidad<sup>1</sup>, albedo<sup>2</sup>, tamaño, masa y densidad

---

<sup>1</sup>Métrica que describe que tan elíptica es una órbita

<sup>2</sup>Proporción de luz que refleja un cuerpo celeste

de planetas gigantes de periodo corto<sup>3</sup>.

- Identificar miembros adicionales de cada sistema planetario descubierto usando técnicas complementarias.
- Determinar las propiedades de las estrellas que alberguen sistemas planetarios.

El instrumento de recolección de datos que alberga el observatorio espacial consiste en un fotómetro diferencial de campo de visión amplio, con un FOV de 100 grados cuadrados<sup>4</sup> que de manera continua y simultánea monitorea el brillo de 100,000 estrellas con suficiente precisión como para detectar el tránsito de planetas de tamaño similar a la Tierra orbitando estrellas enanas de tipo G2. El rango de brillo para las estrellas monitoreadas comprende desde una magnitud visual de 9 hasta 15 [26]. Este se encuentra enclaustrado por un vehículo espacial que orbita la Tierra y soporta diversos sistemas de poder, comunicación y navegación.

El instrumento transmitió de manera exitosa sus primeros resultados en Junio del 2009. Para Diciembre del 2011, el equipo Kepler había anunciado el descubrimiento de 2,326 candidatos planetarios, de los cuales 207 eran de tamaño similar a la Tierra [4]. El 26 de Febrero del 2014, el equipo publicó el catálogo más grande de nuevos planetas extrasolares confirmados hasta la fecha, con más de 700 nuevos planetas confirmados [35].

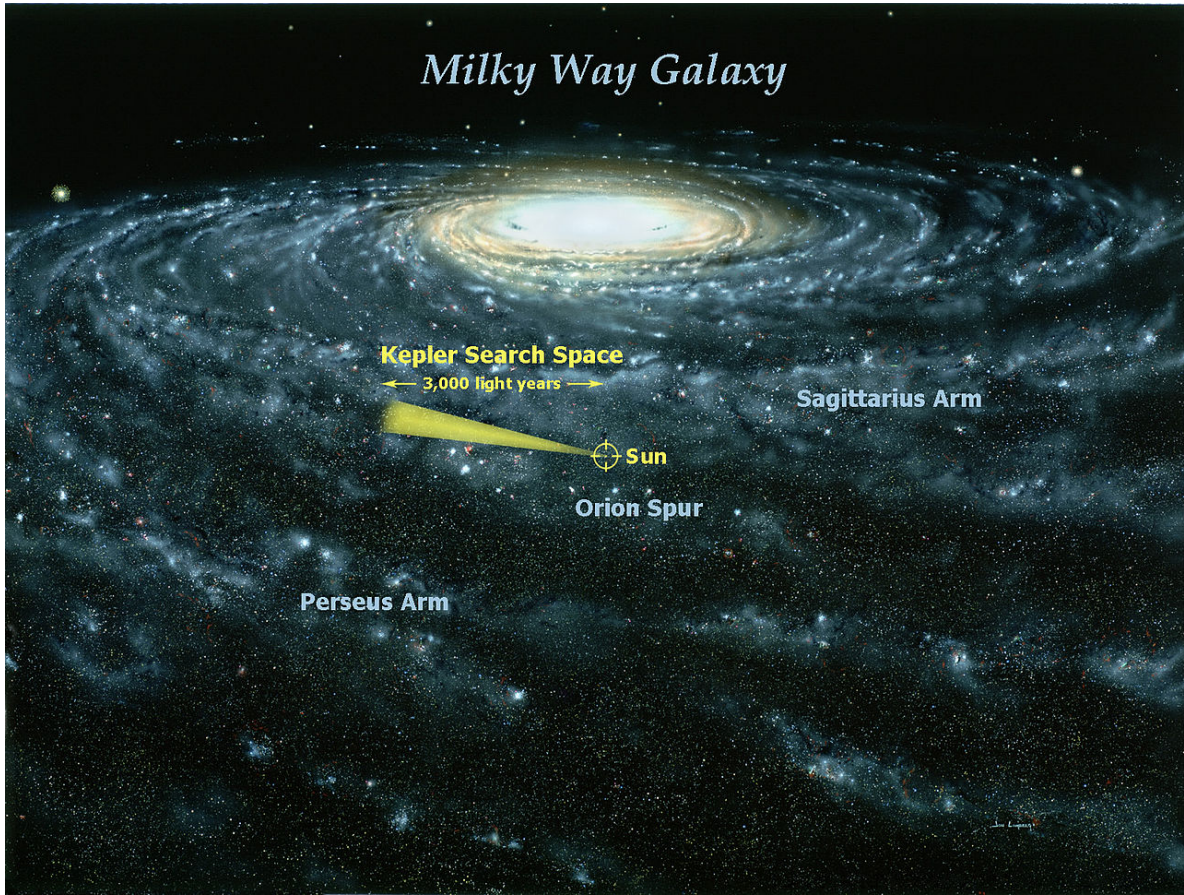
Las observaciones de Kepler se encuentran organizadas en intervalos de 3 meses llamados cuartos, definidos por las maniobras de navegación del instrumento en órbita, realizados para mantener los paneles solares apuntando al sol. Cada mes, los datos acumulados son enviados al centro de operaciones [22].

A continuación se presenta una descripción del proceso por el cual las muestras son recolectadas por el fotómetro y transportadas para su procesamiento y análisis, así como del estado actual del proyecto.

---

<sup>3</sup>Tiempo que le toma a un planeta recorrer su órbita

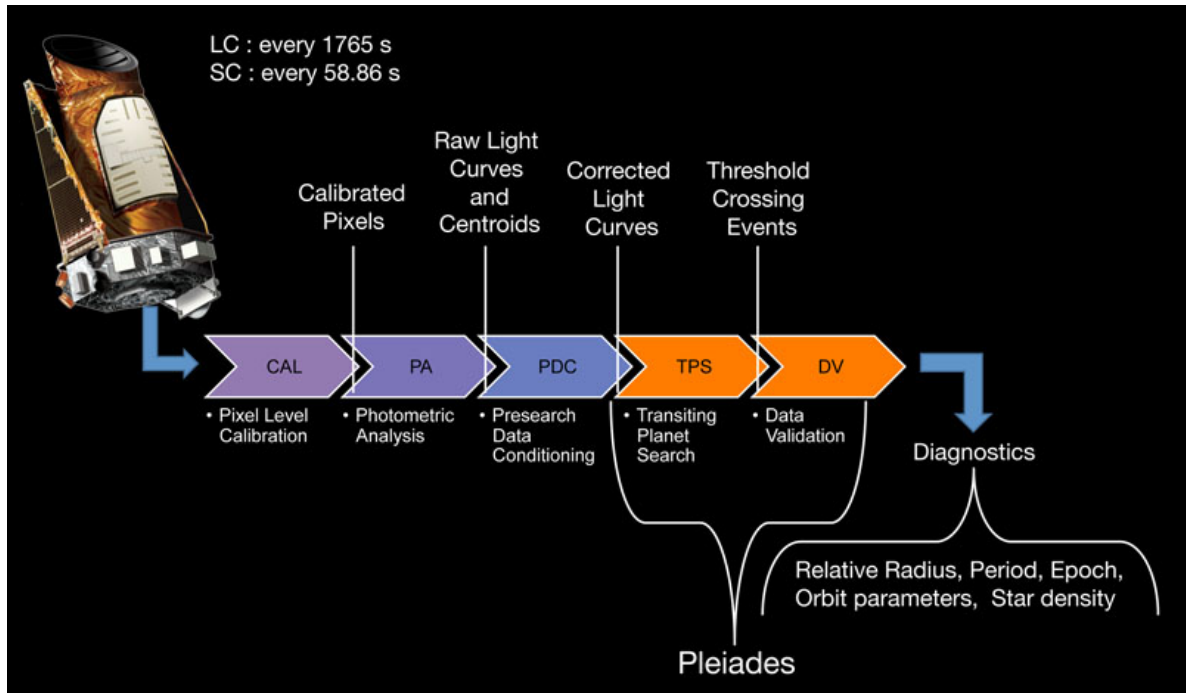
<sup>4</sup>Unidad de medida que describe una sección de una esfera



**Figura 1.5:** El volumen de búsqueda de Kepler, en el contexto de la Vía Láctea. [54]

### 1.4.1 Pipeline de Datos

El proceso por el cual los datos pasan hasta que potenciales candidatos planetarios puedan ser identificados es extenso y consta de varias fases, llevadas a cabo en el Kepler Mission Science Operations Center. Los datos recolectados por el equipo en órbita anteriormente descrito son enviados sin ningún tipo de procesamiento directamente a dicho centro de operaciones[35], encargado de administrar la selección de estrellas observadas por el instrumento, administrar parámetros del mismo, reportar sobre su estado, aplicar tareas de preprocesamiento sobre los datos recolectados, archivar dichos datos y aplicar pruebas estadísticas para la detección de candidatos planetarios, entre otros.



**Figura 1.6:** Diagrama resumen del pipeline de datos de la información recolectada por Kepler, desde su recopilación por medio del instrumento hasta su análisis. [47]

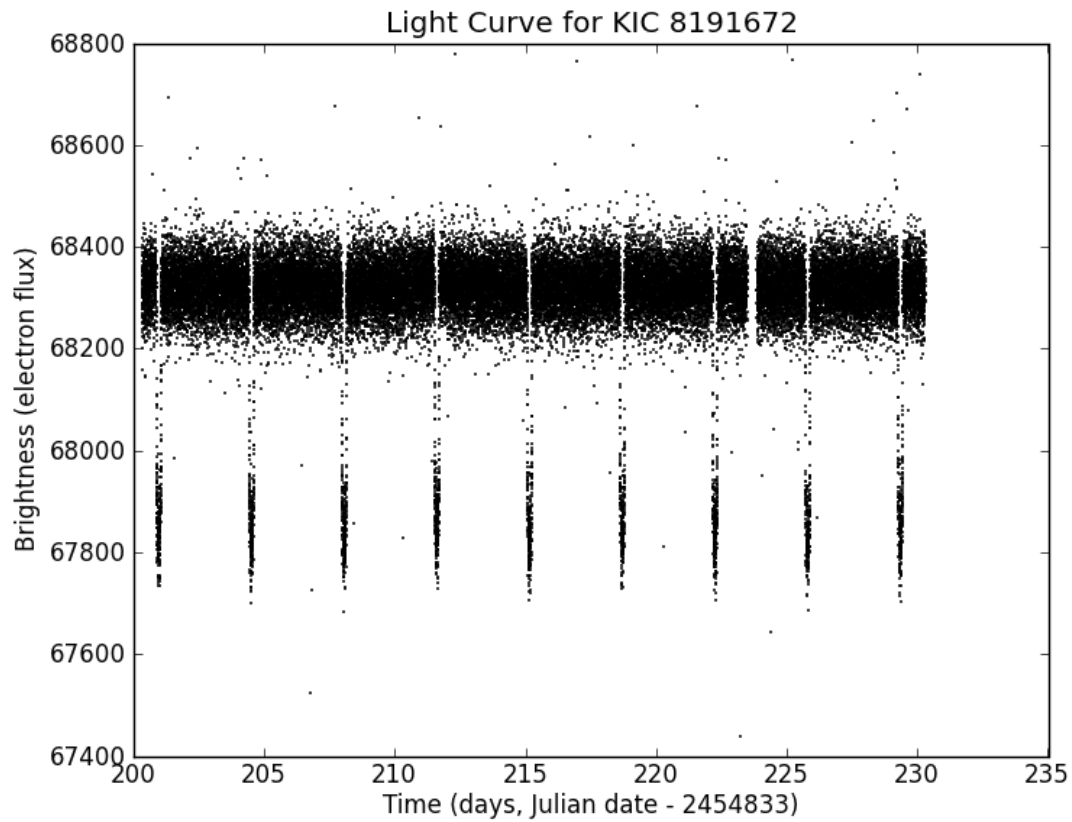
Los datos sin procesar son primero calibrados a nivel de píxeles para corregir diversas anomalías electrónicas inherentes al comportamiento del sensor y otros componentes del equipo que genera las imágenes en el telescopio. A esto le siguen otros procesos de corrección y diagnóstico que remueven errores sistemáticos de los datos recolectados, así como potenciales falsos positivos. Este proceso de preprocesamiento tiene como objetivo incrementar la integridad de los datos, de tal forma que un filtro adaptativo pueda ser aplicado para determinar candidatos de manera confiable [22]. Este proceso puede dividirse en las siguientes etapas:

1. **Calibraciones a nivel de pixel:** Se realizan varias operaciones de limpieza sobre los píxeles de las imágenes recuperadas por el telescopio, principalmente aquellas que involucran parámetros de ruido conocidos provocado por las especificaciones del equipo fotométrico.



2. **Análisis Fotométrico:** Antes de que datos fotométricos y astrométricos puedan ser extraídos, se detectan anomalías en el “*background*” de los pixeles recolectados producidas por el reflejo de la luz del Sol en partículas de polvo en el lente del telescopio, las cuales son corregidas por medio de estimaciones de valores sobre cada pixel.
3. **Búsqueda de Planetas en Órbita:** Se aplica un filtro adaptativo que estima el espectro de poder del ruido observado como una función del tiempo, un proceso diseñado de manera específica para estrellas similares al Sol. Este aumenta la capacidad y precisión de búsqueda de candidatos en este tipo de estrellas. Si por medio de este proceso se detecta un candidato potencial, este es enviado al siguiente proceso del pipeline.
4. **Validación de Datos:** Finalmente, se aplica una serie de pruebas y diagnósticos diseñadas para determinar niveles de confianza de cada candidato detectado por el proceso anterior, de tal manera que se elimine la mayor cantidad posible de falsos positivos, incluyendo eclipses mutuos en sistemas con estrellas binarias, un fenómeno cuyas curvas de luz pueden resultar muy similares a las que comunmente son reconocidas como candidatos planetarios.

Aquellas curvas de luz que sean detectadas por el proceso de filtro como potenciales candidatos son denominadas Objetos de Interés Kepler, o KOI por sus siglas en inglés [1], las cuales son archivadas para su análisis posterior, que determinará si son confirmados como candidatos planetarios. El archivo KOI no es estático, y un objeto catalogado como candidato Kepler podría ser juzgado como un falso positivo después de más rondas de inspección. Actualmente, cada candidato planetario debe ser verificado por medio de métodos secundarios para incrementar el nivel de confianza de su existencia como planeta, como la espectroscopía Doppler, la cual es llevada a cabo por observatorios terrestres para la verificación de la existencia de planetas masivos [54].



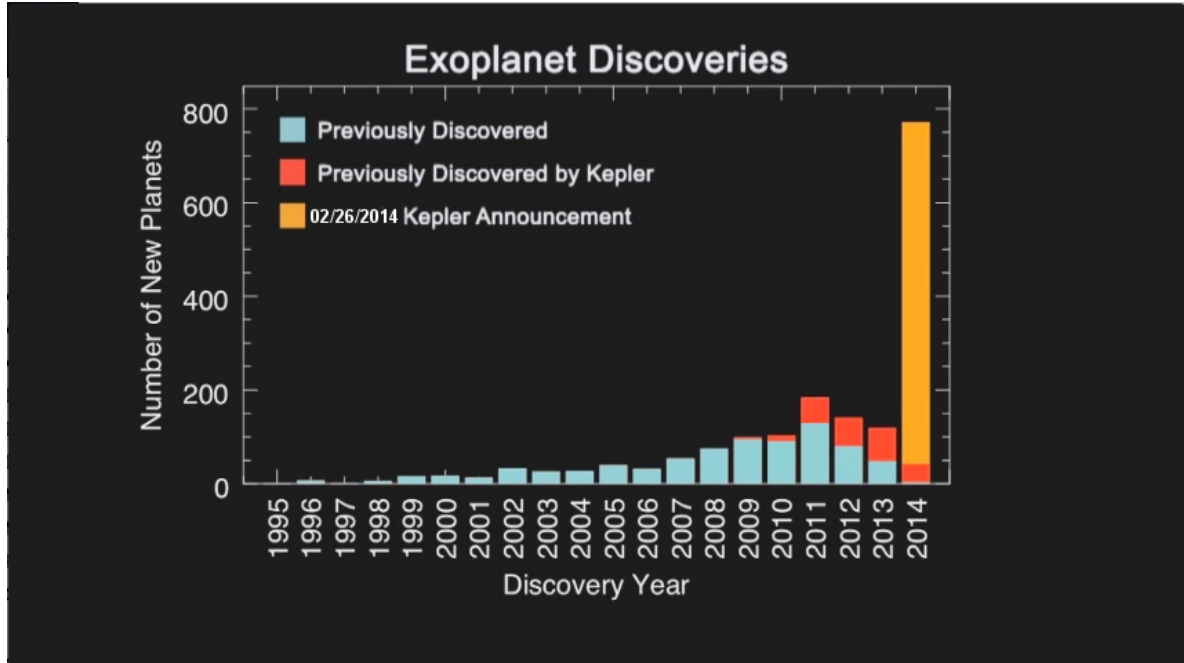
**Figura 1.7:** Curva de luz perteneciente al planeta extrasolar identificado ahora como Kepler-5b. Se puede notar su periodicidad en los niveles de luz decrecientes de manera periódica en esta representación.[27]

Finalmente, no todos los candidatos planetarios pasan por el proceso anteriormente mencionados. Planetas circumbinarios no muestran tránsitos estrictamente periódicos y deben ser estudiados independientemente [38]. De igual forma, estudios independientes pueden omitir o reemplazar partes del pipeline descrito anteriormente.

Hasta la fecha, el proyecto Kepler ha logrado incrementar el número de planetas extrasolares catalogados de manera significativa; un paso de gran importancia para lograr determinar los valores de proporción de planetas extrasolares en términos de tamaño, período orbital, tipos de estrellas que orbitan, entre otros. Más de 3500 planetas extrasolares han sido identificados a partir del análisis de los primeros 3 años de información, incluyendo 100 planetas ubicados en las áreas habitables de sus respectivas

## 1.5. Planteamiento del Problema

estrellas [35]. Esto ha arrojado información de gran valor científico, poniendo en evidencia una relativa abundancia de planetas en nuestra galaxia, incluyendo aquellos de tamaño pequeño [2] [29].



**Figura 1.8:** Histograma representando la cantidad por año de planetas extrasolares descubiertos hasta Septiembre 2014.[35]

## 1.5 Planteamiento del Problema

El proceso de clasificación de planetas extrasolares es un proceso complejo, que requiere de múltiples pasos de preprocesamiento para que los datos que son recolectados directamente por el telescopio Kepler puedan ser tratados. Como se mencionó, sólo una pequeña parte de los candidatos actualmente detectados han sido confirmados debido a que la misión depende de validaciones cruzadas con otros métodos de observación, y parte de esta incertidumbre se debe a los niveles de ruido que dichas muestras contienen. Es necesaria la elaboración y optimización de métodos que no solo puedan identificar candidatos planetarios y descartar falsos positivos de manera efectiva, aportando información acerca de esta clasificación en el proceso.

El ruido se define de manera general como modificaciones no deseadas que una señal puede sufrir durante su captura, almacenamiento, transmisión, procesamiento o conversión [49]. En el caso del proyecto Kepler, el ruido en la señales (en este caso, curvas de luz) que recolecta proviene de diversas fuentes, entre ellas el ruido proveniente del equipo fotográfico del telescopio, y el proveniente de la variabilidad en el comportamiento de las estrellas, así como ruido estelar debido a las distancias involucradas en la recolección de datos, el segundo de los cuales presenta un reto mayor [21]. No sólo esto, sino que durante el desarrollo de la misión, se determinó que dicho ruido excedía en más de un 50 % los niveles esperados previos a la puesta en órbita del telescopio Kepler [15], siendo este una causa importante de falsos positivos.

Para Septiembre del 2014, el manejo del ruido en las muestras de la misión había sido considerado de gran interés para la detección de planetas pequeños y con periodos más largos [29], así como para el mejoramiento en la calidad de datos recolectados sobre aquellos que ya han sido confirmados.

## 1.6 Propuesta de Investigación

Considerando el problema expuesto anteriormente, la presente investigación propuso mejorar la certeza de los resultados de clasificación, limitada por la presencia de dicho ruido en las muestras recolectadas, por medio de la aplicación de algoritmos de clasificación “ensemble” o agregados, basados en la familia de algoritmos de inducción de reglas como clasificadores base [44].

## 1.7 Hipótesis

El uso de algoritmos de aprendizaje de máquina agregados usando algoritmos de inducción de reglas como base para la clasificación de conjuntos de datos multidimensionales y ruidosos, tales como las muestras de la misión Kepler, produce más aciertos que los generados por otros enfoques de clasificación.

### Otras Métricas de Interés

En adición al número de aciertos en el conjunto de prueba del experimento, se estudiaron dos métricas adicionales:

- **Tiempo de Ejecución:** El tiempo en el que cada uno de los algoritmos a estudiar generan resultados. El volumen de datos por analizar en misiones subsecuentes puede ser considerablemente mayor al que el proyecto Kepler ha logrado recopilar, llevando a la necesidad de optimizar el tiempo de ejecución del *pipeline* de datos.
- **Número de Falsos Positivos:** Un algoritmo que de manera conservadora asegure un número bajo de falsos positivos, aún si en otra de las métricas estudiadas fuese deficiente en comparación con otros enfoques, sería de interés para el proyecto Kepler.

## 1.8 Trabajo Relacionado

La mayor parte de los esfuerzos orientados a la clasificación de curvas de luz utilizando técnicas de aprendizaje supervisado vienen de parte de proyectos independientes, los cuales hacen uso de los datos publicados de manera libre por el proyecto Kepler en la forma de archivos de imagen en formato FITS (Flexible Image Transport System), los cuales contienen no sólo la información lumínica sobre las series temporales de observación, sino también parámetros tangenciales a esta información tales como información de orientación de la estrella observada.

Algunos de estos esfuerzos se han concentrado en el minado del catálogo de curvas de luz en busca de curvas anómalas, con el objetivo de crear un catálogo de anomalías disponible públicamente, así como la identificación cuantitativa de variabilidad en dichas curvas anómalas [50].

En cuanto a esfuerzos de identificación de potenciales candidatos a exoplanetas, los resultados de diversos esfuerzos han sido publicados en años recientes. Debray & Wu [13] reportaron un promedio de 80 % en resultados de clasificación de exoplanetas usando como base el catálogo del proyecto Kepler para datos de entrenamiento y prueba empleando varios métodos de clasificación, siendo el más exitoso el de regresión logística.

Un estudio enfocado en la clasificación de eventos de cruce en curvas de luz de Kepler mucho más cercano al propuesto por el presente proyecto es el presentado por McCauliff et al [31], en el que se emplea el método de Random Forest clásico para producir una lista de planetas extrasolares candidatos con moderado éxito y un relativamente bajo nivel de error. Sin embargo, el mismo estudio menciona algunas limitaciones introducidas por muestras con periodos orbitales extensos, así como la existencia de ruido inherente en las mismas, y la necesidad de identificar atributos adicionales que ayuden a descartar potenciales falsos positivos.

---

## Objetivos y Aportes

---

‘Two possibilities exist: Either we are alone in the Universe or we are not. Both are equally terrifying’.

Arthur C. Clarke

Con miras a derivar el máximo beneficio de la presente investigación, así como comunicar de manera acertada los parámetros bajo los cuales esta será llevada a cabo, es necesario el establecimiento claro acerca de qué se espera poder demostrar una vez concluida y cómo medir el aporte que se genera a la ciencia y la comunidad en general.

En adición a lo anterior, debe considerarse que el tema que la presente investigación busca desarrollar es sumamente amplio, e incluyen muchas consideraciones que podrían hacer la ejecución experimental virtualmente interminable. Por lo anterior, junto a los objetivos del presente proyecto a continuación se presentan los alcances y limitaciones del mismo.

## 2.1 Objetivo general

Estudiar la influencia del uso de algoritmos de aprendizaje de máquina agregados usando métodos de inducción de reglas como clasificadores base, sobre el número de aciertos en la clasificación de conjuntos de datos multidimensionales con niveles de ruido significativos, tales como las curvas de luz recuperadas por el proyecto Kepler de NASA.

## 2.2 Objetivos específicos

1. Desarrollar un algoritmo de clasificación agregado para la clasificación de muestras usando aprendizaje de inducción de reglas.
2. Desarrollar un ambiente de pruebas para el establecimiento de medidas de acierto que sea reproducible.
3. Determinar la complejidad espacial y temporal de los algoritmos de clasificación implementados.
4. Medir experimentalmente el tiempo que tome la ejecución del algoritmo a desarrollar durante el proceso de experimentación.



5. Medir el desempeño en términos de clasificación de los algoritmos de inducción de reglas en comparación a los convencionales cuando son aplicados en una metodología de tipo *ensemble*.
6. Llevar a cabo un análisis estadístico que permita confiablemente establecer la influencia del algoritmo desarrollado sobre los resultados de la clasificación.
7. Brindar a la comunidad científica un aporte en relación con la búsqueda de posibles optimizaciones relacionadas al proceso de clasificación de exoplanetas con base en curvas de luz como las provistas por la misión Kepler.

## 2.3 Alcances y Limitaciones

Dentro del alcance de esta propuesta se define el siguiente conjunto de entregables:

- Prototipo de un programa para la clasificación de muestras usando un algoritmo de aprendizaje de máquina de inducción de reglas.
- Programas auxiliares para la ejecución y el control del entregable anterior.
- Análisis estadístico para contrastar los resultados de los experimentos.
- Un artículo científico que se entregará al comité editorial de alguna revista o conferencia, con miras a su publicación.

Es necesario delimitar esta investigación por motivos de tiempo y extensión. Se plantean entonces las siguientes limitaciones; no se tomará en cuenta:

- Ningún algoritmo de clasificación no mencionado en este documento.
- Otros formatos de muestras, o muestras no recolectadas por la misión Kepler de la NASA.

- Consideraciones o la implementación del algoritmo para uso comercial.
- Patente alguna que prohíba el uso de cualquier aporte generado en este proyecto.
- Cualquier otro resultado, documento, *software* o producción que no se encuentren contemplados en los entregables.

En general, debe considerarse que el presente proyecto cubre únicamente la etapa de clasificación de muestras una vez que han sido preprocesadas y publicadas por el proyecto Kepler; y excluye cualquier pre o postprocesamiento que pueda ser aplicado a las mismas.

---

## Experimentación

---

“One of the things Ford Prefect had always found hardest to understand about humans was their habit of continually stating and repeating the very very obvious”.

Douglas Adams

---

Con el motivo de probar la veracidad de la hipótesis plateada en el capítulo 2, se ha elegido el modelo de diseño estadístico de experimentos explicado en [34] y [51, p 561]. La principal razón para esta escogencia es la gran aceptación y uso extendido del mismo en una cantidad considerable de proyectos científicos.

El diseño de experimentos (DoE, por sus siglas en inglés) es un enfoque sistemático y riguroso para la resolución de problemas ingenieriles que aplica principios y técnicas a una colección de datos para asegurar una conclusión válida, defendible y soportable [25]. Además de lo anterior, este modelo es llevado a cabo bajo la condición de minimizar el gasto de recurso, corridas de los experimentos, tiempo y dinero [37].

Existen cuatro tipos de problemas generales en los cuales es provechoso aplicar diseño de experimentos:

1. **Comparativa:** el investigador está interesado en saber si el cambio de un factor genera mejora en un proceso.
2. **Caracterización:** se desea saber cuáles son los factores que afectan un un proceso para poder ordenarlos según su nivel de impacto.
3. **Modelado:** es de interés modelar un proceso cuya salida sea una función matemática con alto poder predictivo y tener un buen estimado de los coeficientes de dicha función.
4. **Optimización:** el ingeniero desea determinar los parámetros óptimos de los factores de un proceso. Esto es, determinar para cada factor el nivel que optimiza la respuesta del proceso.

Se diseñó una serie de experimentos que permitieron determinar el efecto sobre la efectividad de clasificación de muestras, de los algoritmos de clasificación agregados usando clasificadores base de distintos tipos, así como otros atributos del proceso de aprendizaje de máquina supervisado, sobre datos multidimensionales y con altos niveles de ruido, tales como las curvas de luz recolectadas por el proyecto Kepler.

## 3.1 Factores y niveles

En el diseño de experimentos, un *factor* es aquel componente que tiene cierta influencia en las variables de respuesta [34]. El objetivo de un experimento es determinar esta influencia. A su vez, cada factor cuenta con varios *niveles* posibles con los cuales experimentar.

Para efectos del experimento realizado, fue escogido un conjunto de factores relativos al proceso de aprendizaje de máquina supervisado, en el cual un clasificador es entrenado usando un conjunto de muestras de entrenamiento, y subsecuentemente su efectividad es analizada por un conjunto aparte de prueba. Los factores y niveles elegidos fueron:

#### 1. Método de Clasificación

- (a) MODLEM
- (b) Random MODLEM
- (c) Bagging MODLEM
- (d) Boosting MODLEM
- (e) Random Forest

#### 2. Volumen de Datos

- (a) Conjunto completo (100 %)
- (b) Conjunto reducido (50 %)

#### 3. Atributos

- (a) Todos los atributos
- (b) Sin preprocesamiento de imagen

#### 4. Proporción entrenamiento-prueba

(a) 25-75

(b) 50-50

(c) 75-25

#### 3.1.1 Métodos de Clasificación

Como se mencionó anteriormente, la investigación se centra sobre la influencia de el uso de algoritmos de aprendizaje de inducción de reglas, y la aplicación de diversos métodos de aprendizaje agregado, sobre la clasificación de las muestras. Con este fin se empleó el algoritmo MODLEM, descrito con anterioridad, como clasificador base para diversos métodos de agregación, así como sin ningún método de agregación, de tal forma que funcionara de línea base.

En adición a lo anterior, se desarrolló una variación de MODLEM inspirada en el método de muestreo aleatorio de atributos empleado en la construcción de árboles durante la construcción de un clasificador Random Forest, a la que se le denominó Random MODLEM. Esto con el motivo de contrastar la efectividad de este método de muestreo con el de Random Forest.

#### 3.1.2 Volumen de Datos

Debido a que el volumen de datos utilizado durante el proceso de construcción de un clasificador ejerce una influencia importante sobre la efectividad del mismo [17], se decidió investigar la influencia de dicho atributo sobre los resultados arrojados por los diferentes algoritmos de clasificación. El conjunto completo comprende de 7623 muestras, mientras que el reducido consiste en la mitad de dichas muestras, conservando la proporción de clase de las mismas. Más información acerca del conjunto de datos es ofrecida a continuación.

### 3.1.3 Atributos

Existe una gran cantidad de atributos disponibles en cada una de las muestras recolectadas para el conjunto de datos, descritos a continuación. Debido a que cada uno de estos atributos es generado a partir de diferentes elementos en el pipeline de procesamiento del proyecto Kepler, se decidió estudiar la influencia de uno de esos subconjuntos, correspondiente a los atributos basados en información de análisis a nivel de pixel de imágenes recolectadas.

### 3.1.4 Proporción entrenamiento-prueba

Finalmente, se estableció la proporción de datos empleada para dividir los conjuntos de muestras de entrenamiento y prueba para la construcción de los modelos de clasificación, de tal forma que fuera posible determinar si variaciones en este atributo causan cambios en la variable de respuesta.

## 3.2 Otras Consideraciones

Potencialmente, la cantidad de factores que influyen en las variables de respuesta es infinita; sin embargo, por razones de tiempo y presupuesto fueron seleccionado las que se consideraron tenían la mayor posibilidad de influir en las variables de respuesta, tal y como se resume en el cuadro 3.1. Algunos factores que no se consideraron en esta investigación pero que también pueden ejercer influencia son:

- Plataforma de implementación (*hardware* y *software*)
- Propiedades temporales de la recolección de las muestras.

### 3.3. Conjunto de datos

---

	Factores			
	Clasificador	Volumen de Datos	Atributos	Entrenamiento-Prueba
Niveles	MODLEM	Completo	Todos	25-75
	Random MODLEM	Reducido	Sin preproceso	50-50
	Boosting MODLEM			75-25
	Bagging MODLEM			
	Random Forest			

**Cuadro 3.1:** Los factores empleados en la fase de experimentación, así como sus respectivos niveles.

Para realizar al menos una corrida experimental por cada una de las combinaciones de factores descritos anteriormente, un total de  $5 \times 2 \times 2 \times 3 = 60$  ejecuciones serían necesarias, debido a la cantidad de niveles seleccionados para cada factor. Es además necesario, para garantizar una nivel de confianza razonable en los resultados del experimento, y evitar la influencia no deseada de factores externos adicionales sobre el experimento, la ejecución repetida de los experimentos, de tal forma que cada combinación es repetida 5 veces, para un total de 300 ejecuciones experimentales o réplicas. Para asegurar que el orden de la ejecución no introdujera ningún sesgo adicional sobre las variables de respuesta, el orden de estos experimentos fue aleatorizado.

## 3.3 Conjunto de datos

El conjunto de datos utilizado en la investigación corresponde al Catálogo de Candidatos Planetarios para los cuartos Q1 a Q17, Data Release 24 del proyecto Kepler [36]. La principal motivación detrás de elegir este subconjunto de datos se encuentra en que es el mismo conjunto de datos con el cual NASA ha entrenado uno de sus componentes de clasificación, conocido como Autovetter[23].

Todo clasificador de aprendizaje de máquina supervisado requiere de un conjunto de datos conocido como el conjunto de entrenamiento, con ejemplos muestrales que cuenten con una etiqueta o atributo de clase, el cual será utilizado como salida al



proceso de clasificación de cada muestra, y determinará la clasificación final de aquellas en el conjunto de prueba posterior a su construcción. En el caso del conjunto de datos elegidos, este cuenta con tres clases que clasifican los eventos de cruce que representan cada muestra: PC, AFP y NTP. Por sus siglas en inglés estos representan Candidatos Planetarios, Falsos Positivos y Fenómenos “No de Tránsito”. En su estado inicial, el conjunto de entrenamiento contiene un total de 3600 PCs, 9596 AFPs y 2541 NTPs[23]. Una muestra de este conjunto de datos se encuentra en el apéndice de este documento.

En cuanto a atributos, a continuación se presenta una breve descripción de los distintos tipos de atributos presentes en cada muestra del conjunto de datos:

1. **Etiquetas y Banderas de Objetivo:** Atributos básicos de identificación de la muestra, tales como el número de identificación del evento de cruce, el número de planeta asignado al mismo, el conjunto de datos al que cada muestra pertenece, y la fecha en la que fue detectado.
2. **Parámetros de Ajuste de Tránsito:** Se refieren a parámetros de tránsito producidos por uno de los componentes de preprocesamiento del *pipeline* de Kepler, ajustado por un modelo Mandel-Agol [30]. Contienen datos tales como el periodo orbital, la proporción de radio planeta-estrella y su separación.
3. **Parametros de Escala Planetaria:** Consiste en una combinación de los atributos anteriores con propiedades conocidas de la estrella objetivo para determinar métricas en unidades físicas.
4. **Parámetros Estelares:** Subconjunto de atributos que se refieren a la estrella objetivo a partir de la cual el evento de cruce fue encontrado, tales como su temperatura, gravedad, metalicidad, radio y masa.
5. **Estadísticas de Curvas de Luz:** Consiste en atributos avanzados derivados de análisis sobre las curvas de luz producidas por Kepler. Este módulo aplica un filtro

adaptativo para crear caracterizaciones de atributos que han sido determinados como importantes en el proceso de clasificación [20].

6. **Estadísticas de Análisis de Píxeles:** Atributos producidos por medio de un componente de análisis a nivel de *pixel* que busca encontrar indicios de contaminación lumínica en imágenes provistas por el proceso de observación sobre el evento de cruce [10].
7. **Parámetros de *Autovetter*:** Atributos asociados al proceso de entrenamiento y clasificación del módulo de *autovetter* descrito anteriormente. Indica la clase de entrenamiento, así como el resultado para la muestra en conjunto de prueba, porcentajes de acierto y otros.

#### 3.3.1 Transformaciones sobre el conjunto de datos

Se aplicó una serie de transformaciones sobre el conjunto de datos recolectado de tal forma que los experimentos factoriales propuestos por el presente documento pudieran ser efectuados de manera válida y efectiva[51, p 561]. En primer lugar, el orden de muestras en el conjunto de datos fue aleatorizado de tal forma que no existiera sesgo alguno en el orden en el cual se contruirían los clasificadores.

La proporción entre clases fue balanceada de tal forma que existiera igual cantidad de muestras representando las 3 clases disponibles en el conjunto de datos, pasando a un total de 7623 muestras, con 2541 muestras para cada clase. La selección de dichas muestras fue también aleatoria.

Se removieron varios de los atributos disponibles en el conjunto de datos original; en particular todos aquellos pertenecientes al primer grupo descrito, correspondiente a datos de identificación de la muestra, y aquellos relacionados al proceso de clasificación del *autovetter* que no fueran la clase de cada muestra, a ser utilizada por el proceso de entrenamiento.

Asimismo, con el motivo de generar los conjuntos de datos requeridos para abarcar todos los factores y niveles descritos anteriormente como parte del experimento, se duplicó el conjunto de datos, removiendo la mitad de las muestras para cada clase de manera aleatoria. Estos dos conjuntos fueron duplicados una vez más, removiendo de cada una de estas réplicas el subconjunto de datos referidos anteriormente como de Estadísticas de Análisis de Píxeles, y finalmente cada uno de estos conjuntos resultantes fue triplicado y separado aleatoriamente siguiendo las proporciones de división para entrenamiento y prueba establecidas anteriormente: 25-75, 50-50 y 75-25. El resultado fueron 12 conjuntos de datos de entrenamiento y un mismo número de conjuntos para prueba, todos con una representación de clases muestrales iguales, de manera que cumplieran con los requisitos del experimento factorial descrito.

## 3.4 Variables de respuesta

Dado que la hipótesis afirma minimizar los falsos positivos identificados por el método de clasificación, fueron seleccionadas las siguientes como variables de respuesta en cada uno de los experimentos a efectuar:

1. Desempeño final del algoritmo de clasificación, representado por el porcentaje de aciertos en el proceso de clasificación.
2. Cantidad de aciertos durante el proceso de clasificación del subconjunto de muestras de prueba.
3. Tiempo de ejecución tanto de la construcción del modelo de clasificación, como de la clasificación de muestras en el conjunto de prueba.

## 3.5 Análisis de varianza

El análisis de varianza (ANOVA, por sus siglas en inglés) permite asegurar que la variación en los resultados de un experimento no es mayor a la suma de variaciones de los factores y un cierto grado de error en sus medidas. Esto permite aceptar o rechazar la hipótesis con una probabilidad de error (de preferencia muy baja) con base en evidencia estadística [51, p 507].

El ANOVA tiene como objetivo analizar la relación entre una variable cuantitativa  $X$  y una variable cualitativa  $Y$  de  $k$  atributos. Cada atributo  $i$  define una población dada por la variable cuantitativa [42, p 247].

$X_i$  : variable  $X$  restringida al atributo  $i$ .

Así, se tienen  $k$  poblaciones  $X_1, X_2, \dots, X_n$  (llamadas tratamientos) que se suponen normales, independientes, con variancias similares y con medias poblacionales  $\mu_1, \mu_2, \dots, \mu_n$ . Se desea determinar si  $X$  no varía según el atributo de  $Y$ , es decir, si las poblaciones son equivalentes y entonces los tratamientos son igualmente efectivos. Para ello se plantean y contrastan las hipótesis:

$H_0$  :  $X$  no varía según el atributo de  $Y$  (poblaciones equivalentes, es decir  $\mu_1 = \mu_2 = \dots = \mu_k$ ).

$H_1$  :  $X$  varía según el atributo de  $Y$  (poblaciones no equivalentes), al menos dos de las medias no son iguales.

Una gran cualidad de ANOVA es que permite analizar varias poblaciones, mientras que otros métodos no permiten más de dos.

La herramienta de *software* seleccionada para ejecutar el proceso de análisis de varianza y presentar los datos de manera gráfica consiste en el paquete provisto por el proyecto R [39]. Más información acerca de los datos de entrada y salida y las

representaciones gráficas obtenidas a partir de esta herramienta es provista en la siguiente sección del presente documento.

## 3.6 Implementación de Random MODLEM

Como se mencionó anteriormente, el método de *Random Prism* propuesto por Bramer [44] integra dentro de este algoritmo básico las características de aleatorizar la selección de características en el espacio de búsqueda, una característica clave del modelo de clasificación de Random Forest. Inicialmente era de interés el empleo de este método de clasificación como parte de la investigación, sin embargo el mismo impone limitantes importantes dentro del proceso de transformación de datos no compatibles con los datos experimentales; en particular posee una restricción a atributos únicamente nominales y no numéricos, lo cual lo hace poco apto para interpretar la información con la que se cuenta.

Por esta razón, el algoritmo MODLEM [45] fue empleado en su lugar, tanto como algoritmo único y como base para los clasificadores de agregación, por medio de *boosting* y *bagging*. El mismo cuenta con las mismas características atribuibles a *Random Prism* en términos de inducción de reglas, mientras que admite atributos numéricos en muestras. Con el motivo de además medir la influencia de la selección de características aleatorias en el espacio de búsqueda, se desarrolló una versión modificada de MODLEM, a la cual se le denominó Random MODLEM, que introduce esta característica en su algoritmo de inducción de reglas.

El algoritmo, descrito anteriormente, construye una lista de reglas a partir de la prioridad encontrada para cada atributo a partir de la efectividad de clasificación sobre ese atributo en el conjunto de entrenamiento [45]. La modificación realizada consistió en la aplicación de un paso anterior a este ciclo, en el cual antes de seleccionar cada regla, el conjunto de atributos es aleatoriamente modificado de tal forma que sólo se contempla un subconjunto de atributos en cada paso, asegurando que múltiples iteraciones del algoritmo produzcan diferentes conjuntos de reglas, evitando sesgos producidos por overfitting, un fenómeno común en el aprendizaje agregado [44].

Procedure RANDOM\_MODLEM

(input B - a set of positive examples from a given decision concept;

    C - a set of features from B;

    criterion - an evaluation measure;

output T - single local covering of B, treated here as rule condition pts)

begin

    G := B; {A temporary set of rules covered by generated rules}

    T :=  $\emptyset$ ;

    while G  $\neq \emptyset$  do {look for rules until some examples remain uncovered}

    begin

        T :=  $\emptyset$ ; {a candidate for a rule condition part}

        S := U; {a set of objects currently covered by T}

        while (T =  $\emptyset$ ) or (not([T]  $\subseteq$  B)) do {stop condition for a rule}

        begin

            t :=  $\emptyset$ ; {a candidate for an elementary condition}

            D := subsample of C

            for each attribute q  $\in$  D do {looking for best elem. condition}

            begin

                new\_t := Find\_best\_condition(q,S);

```

        if Better(new_t, t,criterion) then t := new_t;
        {evaluate if a new condition is better than previous one
        according to the chosen evaluation measure}
    end;
    T := T  $\cup$  {t}; {add the best condition to the candidate rule}
    S := S  $\cap$  [t]; {focus on examples covered by the candidate}
end; { while not([T]  $\subseteq$  B }
for each elementary condition t  $\in$  T do
    if [T - t]  $\subseteq$  B then T := T - t; {test a rule minimality}
    T := T  $\cup$  {T}; {store a rule}
    G := B - T  $\in$  T [T] ; {remove already covered examples}
end; { while G =  $\emptyset$  }
for each T  $\in$  T do
    if T  $\in$  T - T [T] = B then T := T - T {test minimality}
end procedure

```

## 3.7 Herramientas y Ejecución

Se emplearon varios paquetes de *software* en la preparación y ejecución del experimento. En primer lugar, las transformaciones descritas anteriormente fueron modeladas por medio de *scripts* para generar los pares de datos para la construcción y evaluación de modelos de clasificación. Posterior a esto, se desarrolló un *pipeline* en la forma de un segundo *script* para aleatorizar, preparar y ejecutar los experimentos, utilizando el lenguaje de programación Java y usando como base la plataforma de aprendizaje de máquina Weka 3 [18], la cual se usó para evaluar los clasificadores. Se escogió Weka por varias razones:

- Es una herramienta aceptada por la comunidad académica y cuya funcionalidad

está demostrada [40][41][32].

- La estandarización de entrada y salida de los clasificadores.
- La forma de particionar el conjunto de datos en “entrenamiento” y “evaluación”.
- La capacidad de guardar el resultado de la clasificación, para poder explorar más de cerca los datos de ser necesario.
- El detalle con el cual se muestra la salida de cada clasificación. En la figura 3.1 se muestra un ejemplo de salida para una clasificador.

Los archivos de muestras siguen el formato ARFF<sup>1</sup>, utilizado por Weka para la interpretación de los mismos.

---

<sup>1</sup><http://www.cs.waikato.ac.nz/ml/weka/arff.html>



```

=== Evaluation on test split ===

Time taken to test model on training split: 0.06 seconds

=== Summary ===

Correctly Classified Instances      1172           90.5019 %
Incorrectly Classified Instances    123           9.4981 %
Kappa statistic                    0.8569
Mean absolute error                 0.1361
Root mean squared error             0.2293
Relative absolute error             30.5921 %
Root relative squared error         48.5675 %
Coverage of cases (0.95 level)     99.8456 %
Mean rel. region size (0.95 level) 60.0257 %
Total Number of Instances          1295

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0.967   0.015   0.971     0.967   0.969     0.952   0.997    0.993    PC
                0.919   0.096   0.837     0.919   0.876     0.806   0.966    0.922    AFP
                0.819   0.032   0.917     0.819   0.865     0.814   0.970    0.948    NTP
Weighted Avg.   0.905   0.049   0.908     0.905   0.905     0.859   0.978    0.955

=== Confusion Matrix ===

  a  b  c  <-- classified as
434 11  4 | a = PC
 12 417 25 | b = AFP
  1  70 321 | c = NTP

```

**Figura 3.1:** Fragmento de la salida de un clasificador en Weka. Se pueden identificar datos importantes como el número y porcentaje de instancias clasificadas correcta e incorrectamente, la matriz de confusión y un bloque detallando la precisión por categoría.

El experimento se ejecutó en un equipo de tipo *cluster* en el Centro Nacional de Alta Tecnología durante el transcurso de aproximadamente 30 horas. Este cluster posee máquinas homogéneas, lo cual permite hacer ejecuciones simultaneas sin afectar las variables no controladas en el software, acelerando el proceso de experimentación. El orden de los experimentos fue aleatorio, garantizando evitar sesgos en el análisis de resultados generados a raíz del orden en los mismos. Para cada uno de los experimentos, se construyó un modelo de clasificación compatible con cada una de las combinaciones disponibles de factores, se evaluó a partir de su conjunto de pruebas correspondiente, y se almacenaron tanto las variables de respuesta asociadas al experimento, como la salida completa de cada experimento provista por Weka, tal como se observa en la figura 3.1. Este proceso fue llevado a cabo por medio de los *scripts* auxiliares desarrollados.

---

## Resultados y Análisis

---

“Men must fumble awhile with error  
to separate it from truth, I think- as  
long as they don’t seize the error  
hungrily because it has a pleasanter  
taste”.

Walter M. Miller Jr.

Los datos experimentales se analizaron por medio de ANOVA, donde los grupos estudiados para el experimento fueron la combinación de cada uno de los factores descritos durante la sección anterior, como se ilustra en el cuadro 4.1.

	Factores			
	Clasificador	Volumen de Datos	Atributos	Entrenamiento-Prueba
Niveles	MODLEM	Completo	Todos	25-75
	Random MODLEM	Reducido	Sin preproceso	50-50
	Boosting MODLEM			75-25
	Bagging MODLEM			
	Random Forest			

**Cuadro 4.1:** Los factores empleados en la fase de experimentación, así como sus respectivos niveles.

Para utilizar ANOVA es necesario validar una serie de supuestos [33], que en caso de no hacerlo, no se puede asegurar la validez del resultado. Estos supuestos son:

1. **Independencia y aleatoriedad** de las observaciones: esto se valida con el diseño experimental, donde se muestra que cada corrida del experimento es independiente de cualquier otra. Este supuesto se aseguró al ejecutar todas las instancias del experimento en un sólo equipo, de manera aleatoria con respecto a los factores involucrados.
2. **Normalidad** de los residuos: Esto se puede verificar con la prueba de Lilliefors (Kolmogorov-Smirnov) [46] o de manera visual con un gráfico de cuantiles contra cuantiles (Q-Q en inglés). Para la prueba, el estadístico  $D$  debe ser menor que el valor-p para confirmar normalidad. En el gráfico Q-Q, se muestran los residuos estandarizados de la variable medida por cuantiles contra los cuantiles teóricos de la variable de respuesta y una línea de tendencia, mientras más se alejen los puntos de esta línea, menos normales son los datos.
3. **Homogeneidad** de varianzas (homocedasticidad): Esto se puede verificar con la

prueba de Levene [28] o de manera visual con un gráfico de residuos contra valores ajustados (RvVA) de ANOVA. Para la prueba, el estadístico  $W$  debe ser menor que el valor-p para confirmar homogeneidad. En el gráfico, los puntos deben estar dispersos y no seguir algún patrón fácilmente reconocible, en caso contrario, los datos no son homocedásticos.

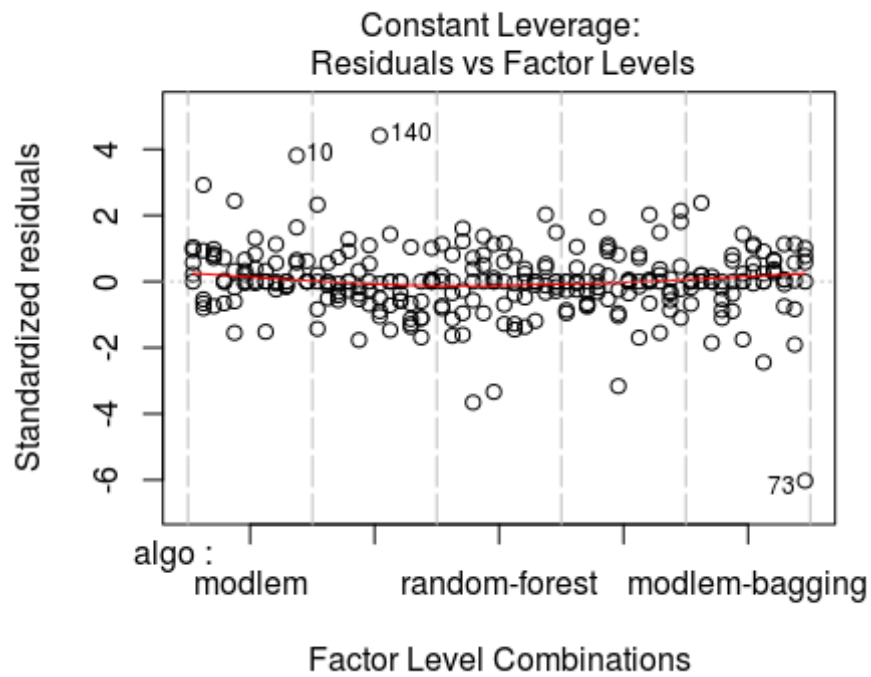
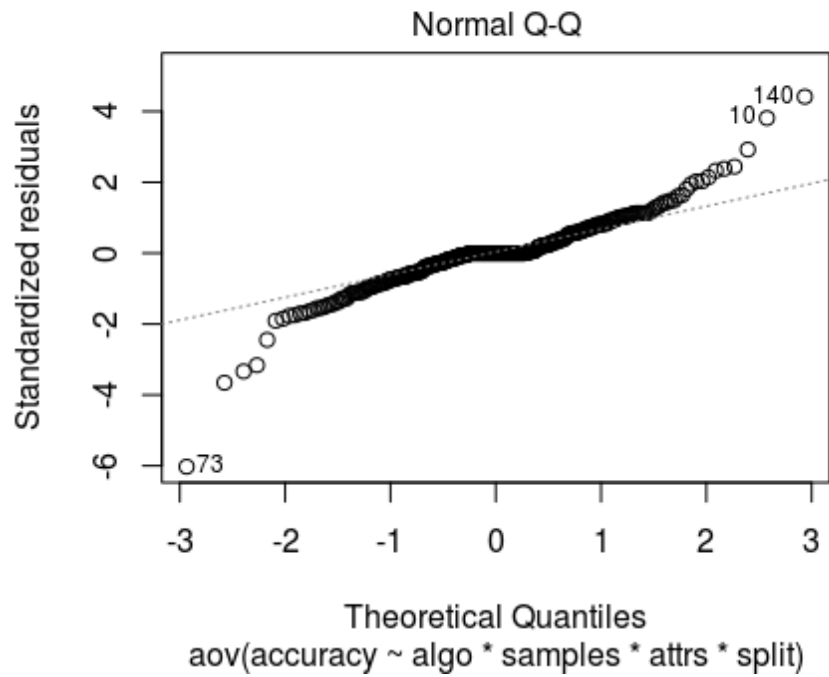
El paquete estadístico R<sup>1</sup> se utilizó para todo el análisis siguiente.

## 4.1 Verificación de supuestos

Los supuestos descritos anteriormente fueron verificados de manera visual empleando la herramienta mencionada anteriormente. En la figura 4.1 se muestra el gráfico Q-Q y el gráfico RvVA que corresponden a los datos recolectados durante la fase de experimentación. Como se puede notar, ambos presentan las características deseables descritas anteriormente.

---

<sup>1</sup><http://www.r-project.org/>



**Figura 4.1:** Exploración visual de la normalidad y homogeneidad de varianza en los datos de tiempo sin transformaciones.

Tanto la exploración visual del gráfico de Cuantiles contra Cuantiles como la del de Residuos contra Valores Ajustados se ajustan a los patrones esperados para la validación de los supuestos asociados a la ejecución correcta de ANOVA, demostrando de manera inequívoca la validez de los resultados arrojados por el mismo.

## 4.2 Análisis de resultados

Una vez superado los supuestos de ANOVA, se procedió a hacer el análisis de los tiempos transformados a rangos.

El procedimiento `aov` en R es el utilizado para hacer el análisis de varianza. Esta función reporta dos salidas: una tabla de ANOVA y el cuadro resumen, de los cuales la tabla es la que nos interesa.

La figura 4.2 muestra la invocación de la función `aov` y la tabla 4.2 muestra la relevancia de cada combinación de factores.

```
> aov(accur(t) ~ algo * samples * attr * split)
```

**Figura 4.2:** Invocación de la función `aov` para Análisis de Varianza en R. El parámetro recibido muestra las relaciones entre las variables:  $accur(t)$  (precisión de clasificación) la variable de respuesta se debe analizar en función de los factores y todas sus combinaciones.

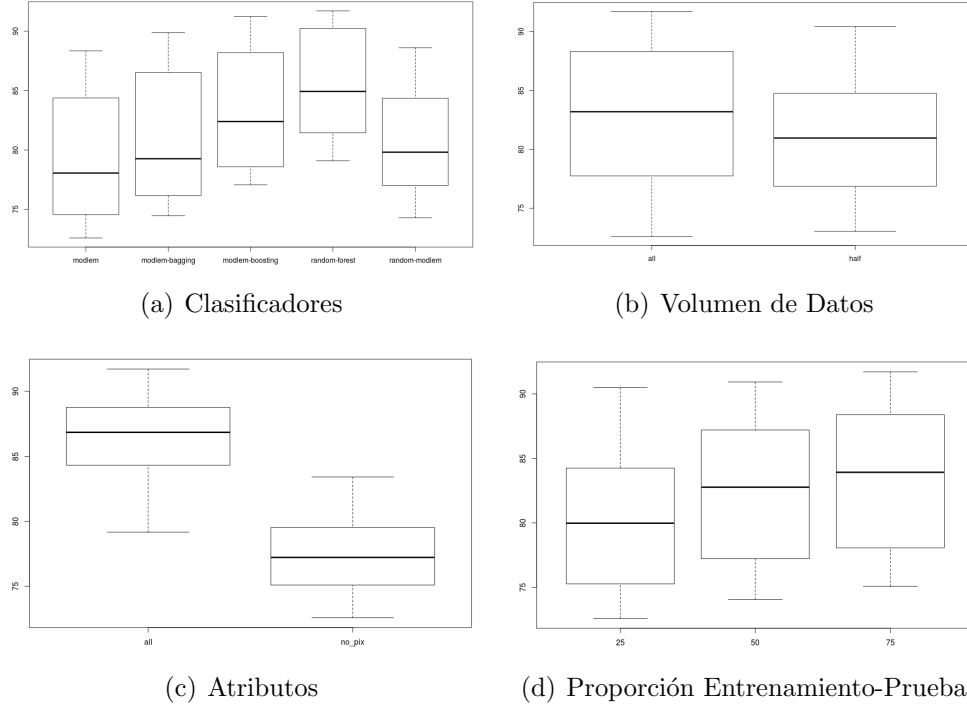
La tabla indica que casi todas las combinaciones de factores influyen sobre la variable de respuesta. Estos son los cuatro factores principales, la mayor parte de las interacciones de segundo nivel representadas por la combinación de 2 factores, y todas las de tercer y cuarto nivel.

El efecto de los factores principales se evidencia en la figura 4.3. El efecto de estos es esperable: a más información disponible para la ejecución del proceso de entrenamiento del clasificador, más exacta se vuelve la clasificación y por lo tanto el porcentaje de

Factor	Df	Sum Sq	Mean Sq	F-value	Pr(>F)	Sig.
algo	4	1389	347	2235.337	<2e-16	3
samples	1	328	328	2108.853	<2e-16	3
attrs	1	6064	6064	39025.787	<2e-16	3
split	2	524	262	1687.281	<2e-16	3
algo:samples	4	7	2	11.671	1.12e-08	3
algo:attrs	4	54	13	86.844	<2e-16	3
samples:attrs	1	106	106	684.674	<2e-16	3
algo:split	8	25	3	20.118	<2e-16	3
samples:split	2	1	1	4.566	0.0113	1
attrs:split	2	17	9	55.003	<2e-16	3
algo:samples:attrs	4	9	2	15.055	5.39e-09	3
algo:samples:split	8	10	1	7.661	3.95e-09	3
algo:attrs:split	8	10	1	7.859	2.23e-09	3
samples:attrs:split	2	19	10	62.647	<2e-16	3
algo:samples:attrs:split	8	9	1	6.925	3.34e-08	3
Residuos	240	37	0			

**Cuadro 4.2:** Tabla ANOVA para los rangos de tiempo del experimento A.

aciertos en dicho proceso. Es además apreciable el impacto que los algoritmos estudiados poseen sobre el proceso de clasificación, evidenciando la importancia de una selección apropiada de los mismos. La combinación entre la proporción de entrenamiento y prueba junto con la cantidad de muestras mostro un nivel bajo de importancia, indicando una menor relevancia en la certeza del proceso que la influencia que ejercen los otros dos factores.



**Figura 4.3:** Efecto de los factores principales del experimento.

El factor del clasificador empleado presenta un efecto marcado en cuanto a la variable de respuesta de aciertos en el proceso de clasificación. Como se plantea en la hipótesis, los métodos que utilizan algún tipo de clasificación agregada son marcadamente superiores al método que utiliza una única instancia para la clasificación de todas las muestras proporcionadas. Los 3 clasificadores con mayor cantidad de aciertos son además aquellos que emplean *boosting* como método de agregación: *Modlem-boosting*, *Random Modlem* y *Random Forest*. Estos resultados corresponden a los esperados durante el proceso de planeamiento del experimento.

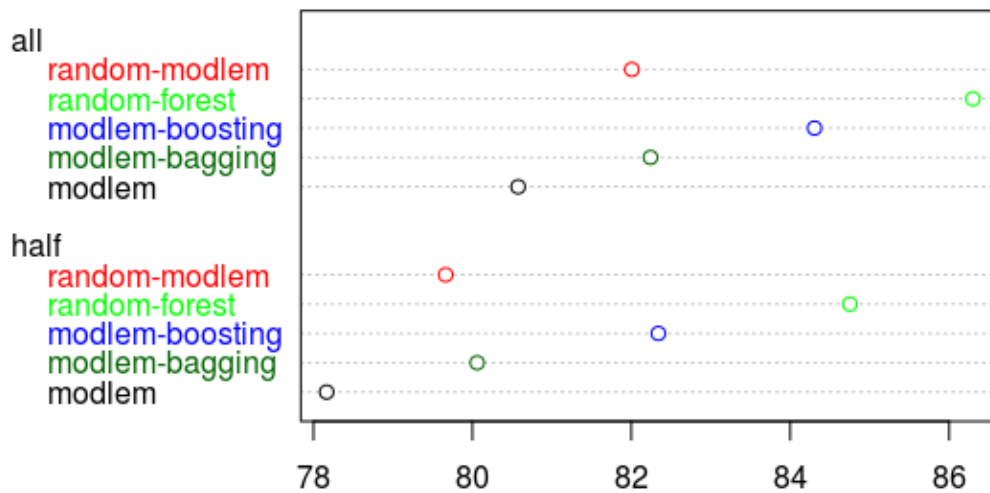
El factor de *Volumen de Datos* tuvo un efecto menos importante, más no insignificante, sobre la certeza del proceso de clasificación. Como es de esperar, el efecto del volumen de datos involucrado en el proceso de aprendizaje es significativo [17], sin embargo es importante señalar que el efecto no es proporcional, indicando que si bien un volumen de datos mayor parece resultar en una cantidad de aciertos mayor, el proceso de entrenamiento y clasificación de datos es resiliente a conjuntos de muestras reducidos,



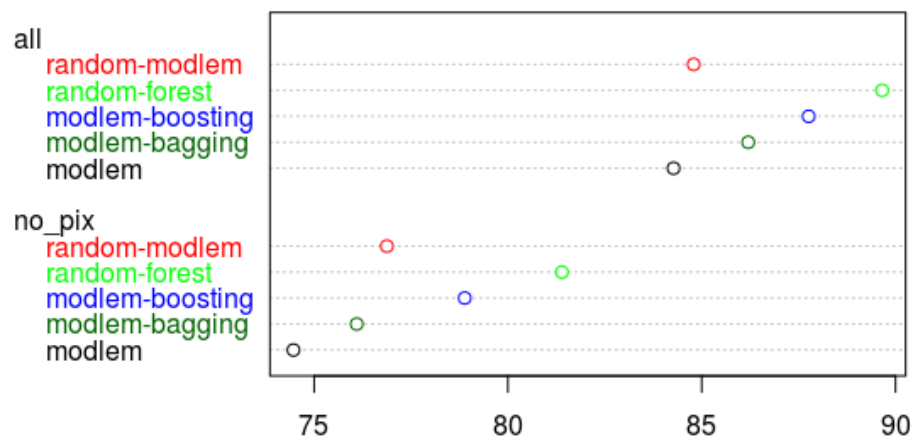
y el incremento de muestras de manera arbitraria no va a necesariamente resultar en un incremento significativo en el número de aciertos obtenido.

El factor correspondiente al de *Conjunto de Atributos* tuvo una influencia particularmente fuerte en el número de aciertos. Si bien coincide con las expectativas asociadas al experimento el que a mayor cantidad de atributos, el porcentaje de aciertos en la clasificación subiera, el impacto de los datos provenientes de la información visual pone en evidencia la importancia supuesta por la presente investigación de elegir algoritmos de clasificación resilientes al ruido muestral, presente en datos tales como los provenientes de este conjunto de atributos.

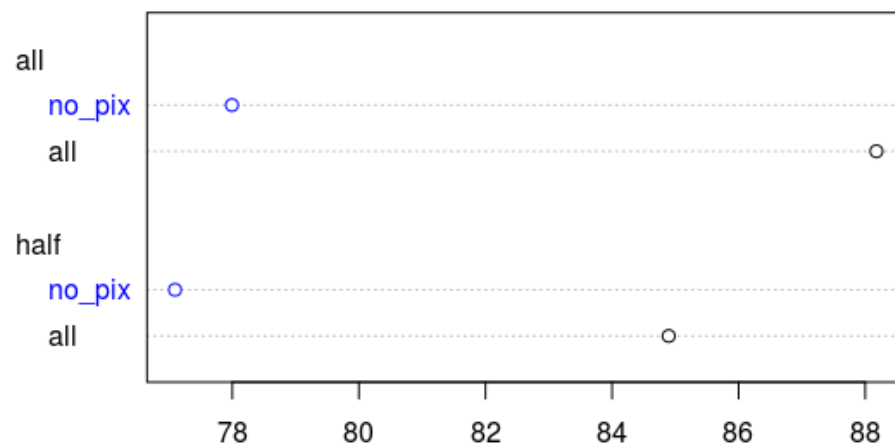
Finalmente, el factor de *Proporción* entre el conjunto de muestras de entrenamiento y prueba influyó de manera esperada, con mejores resultados en aciertos de clasificación al haber un conjunto mayor de datos de entrenamiento.



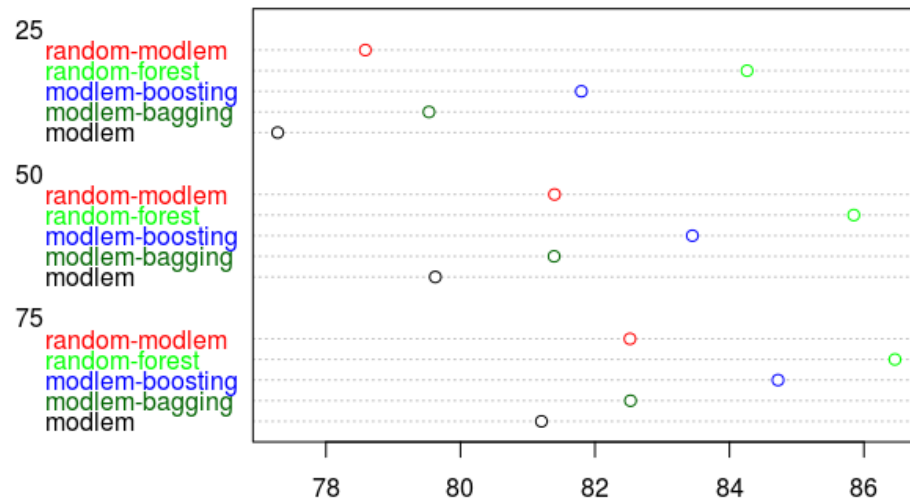
**Figura 4.4:** Efecto de las combinaciones de los factores de segundo nivel en el experimento, Clasificadores-Volumen.



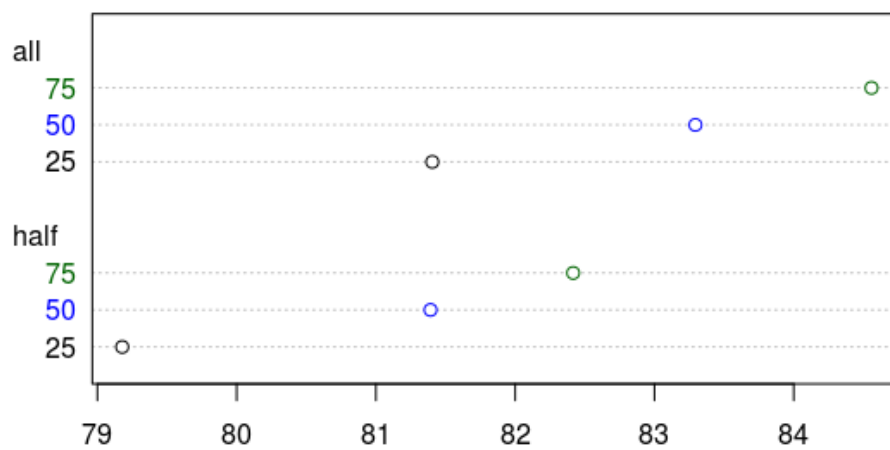
**Figura 4.5:** Efecto de las combinaciones de los factores de segundo nivel en el experimento, Clasificadores-Atributos.



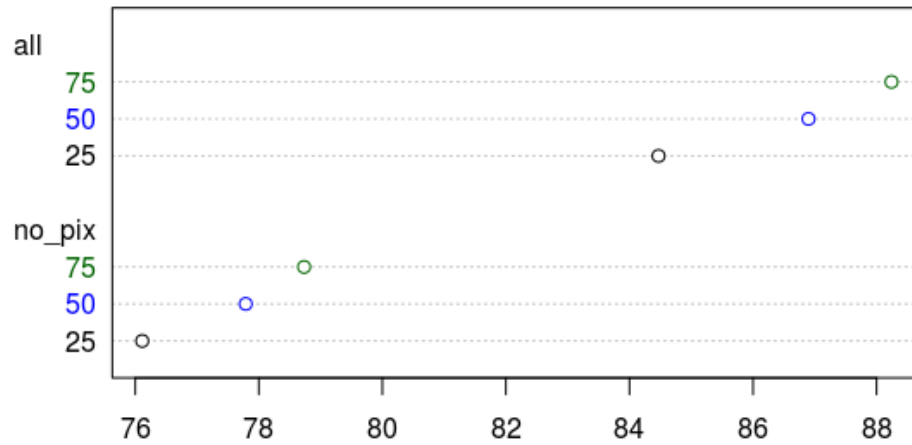
**Figura 4.6:** Efecto de las combinaciones de los factores de segundo nivel en el experimento, Volumen-Atributos.



**Figura 4.7:** Efecto de las combinaciones de los factores de segundo nivel en el experimento, Clasificadores-Proporción.



**Figura 4.8:** Efecto de las combinaciones de los factores de segundo nivel en el experimento, Volumen-Proporción.



**Figura 4.9:** Efecto de las combinaciones de los factores de segundo nivel en el experimento, Atributos-Proporción.

En las figuras 4.4, 4.5, 4.6, 4.7, 4.8 y 4.9 puede observarse la interacción producida entre factores de primer nivel para formar aquellos de segundo nivel. Ninguna de estas interacciones se desvía de las observaciones anteriormente mencionadas, ya que se adecúan a las expectativas del experimento: A mayor cantidad de información disponible ya sea representada por medio de un mayor conjunto de entrenamiento, atributos o volumen muestral, mayor la cantidad de aciertos generados. La oscilación más pronunciada se produce en aquellas interacciones que involucran el particionamiento de los atributos en el conjunto de datos, señalando la relevancia de atributos que provienen de las etapas posteriores en el proceso de refinamiento de datos del proyecto Kepler, llegando a generar una diferencia de hasta 15 % cuando la influencia de este factor y el algoritmo de clasificación empleado es estudiada (4.5). En contraste, los gráficos de interacción compuestos parcialmente por el factor de volumen de datos y la proporción de entrenamiento y prueba presentan una oscilación en cuanto a la certeza de clasificación menor. Cuando la influencia de la interacción entre estos dos se observa, la variabilidad es la menor; un

5 % entre la combinación que produce más y menos aciertos<sup>4.8</sup>.

En general, en cuanto a aquellos factores de segundo nivel involucrando el factor de método de clasificación, las observaciones anteriores se mantienen; aquellos clasificadores que corresponden a metodologías agregadas, en particular aquellos implementando alguna variación de *boosting*, presentaron porcentajes de aciertos mayores a las otras alternativas exploradas.

Estas observaciones y supuestos se mantienen al explorar las interacciones en factores de tercer y cuarto nivel. Vale la pena destacar que si bien de manera independiente la influencia de los factores sobre el porcentaje de aciertos es variable, los resultados producidos por el proceso de ANOVA indican que la mayor parte de estos contribuyen de manera significativa a dicha variable de respuesta, sin uno que en particular domine al resto.

## 4.3 Modelo de Clasificación

Como se mencionó, una ventaja de los algoritmos de clasificación de inducción de reglas consiste en que el modelo de clasificación generado durante el aprendizaje supervisado de los mismos puede ser representado como una lista ordenada de reglas. Esto permite que el modelo sea fácilmente representable y por consiguiente legible por investigadores humanos. Esto puede llevar a reconocer reglas y atributos relevantes para la clasificación de muestras, llevando a optimizaciones en cuanto a los algoritmos empleados y a los datos que se recolectan acerca de cada evento de cruce. La figura 4.10 muestra un subconjunto de las reglas derivadas por uno de los procesos de clasificación ejecutados en el experimento. El conjunto completo de reglas de esta misma iteración se encuentra disponible en la sección de apéndice.

En la figura pueden distinguirse varios de los atributos que el modelo encontró más relevantes para la clasificación de eventos de cruce. En primer lugar, el denominado

```

R1. (tce_nkoi >= 2.5)&(tce_smet >= -0.25)
    => (av_training_set = PC)    (229/229, 18.03%)
R2. (tce_nkoi >= 1.5)&(tce_fwm_sdec >= 40.68)&(tce_ingress >= 0.02)
    => (av_training_set = PC)    (401/401, 31.57%)
R3. (tce_nkoi >= 1.5)&(tce_mesmad < 0.67)&(tce_robstat < 93.67)
    => (av_training_set = PC)    (129/129, 10.16%)
R4. (tce_nkoi >= 1.5)&(tce_dicco_msky < 0.46)&(tce_steff < 6340.5)
    => (av_training_set = PC)    (346/346, 27.24%)
R5. (tce_sradius < 0.51)&(tce_rminmes < 0.2)&(tce_ror < 0.11)
    => (av_training_set = PC)    (36/36, 2.83%)
R6. (tce_rminmes < 0.19)&(tce_fwm_sdec >= 43)&(tce_chisq2 < 1632)
    => (av_training_set = PC)    (336/336, 26.46%)
R7. (tce_minmes >= -3.55)&(tce_smet >= 0.23)&(tce_model_snr >= 17.28)
    => (av_training_set = PC)    (48/48, 3.78%)
R8. (tce_minmes >= -3.55)&(tce_fwm_sra >= 20.03)
    => (av_training_set = PC)    (13/13, 1.02%)
R9. (tce_chisq2 < 258.55)&(tce_minmes >= -2.7)&(tce_dicco_msky < 3.38)
    => (av_training_set = PC)    (88/88, 6.93%)

```

**Figura 4.10:** Modelo de clasificación producido por MODLEM durante una corrida experimental

*tce\_nkoi* corresponde a la cantidad de observaciones de eventos de cruce que ocurrieron en la estrella objetivo a la que la muestra pertenece. Esto apunta a que mientras más eventos de cruce fueron reconocidos durante el periodo de observación sobre una estrella, más posibilidades hay de que en efecto exista un planeta transitándola. Otro atributo de interés es *tce\_smet*, que corresponde a la metalicidad de la estrella; definida como la proporción entre el hierro e hidrógeno detectada en la superficie de la misma. Una proporción importante (31%) de la clase de *Planetary Candidates* es caracterizada a base de estos atributos, y *tce\_ingress*, la diferencia temporal entre el primer y segundo contacto de tránsito.

---

## Conclusiones y Trabajo Futuro

---

“Oh, hell! I can’t sleep!” “Neither can I! But I might as well try—as a matter of principle.” Twelve hours later, sleep was still just that—a matter of principle, unattainable in practice”

Isaac Asimov

---

A lo largo del presente trabajo se ha evaluado el desempeño de diversos métodos de clasificación para datos con ruido muestral, tal como lo son los provenientes del proyecto de búsqueda de cuerpos extrasolares Kepler de la NASA, bajo diversos factores y en términos de precisión de clasificación, como medio a determinar la efectividad de dichos clasificadores y los principios subyacentes de su diseño.

El proceso experimental aquí descrito pretende aportar evidencia estadística a favor de la hipótesis de investigación:

*El uso de algoritmos de aprendizaje de máquina agregados usando algoritmos de inducción de reglas como base para la clasificación de conjuntos de datos multidimensionales y ruidosos, tales como las muestras de la misión Kepler, produce más aciertos que los generados por otros enfoques de clasificación.*

A partir de los datos expuestos por dicho proceso experimental, se llega a las siguientes conclusiones:

1. Pueden distinguirse diversos grados de aprobación del enunciado anterior; en efecto los métodos agregados de aprendizaje de máquina son los que resultan en un porcentaje mayor de aciertos en los resultados de clasificación. Sin embargo, se encontró que métodos basados en árboles de decisión (empleando métodos agregados en su diseño) producen más aciertos que aquellos basados en inducción de reglas.
2. La mayor parte de los factores estudiados; consistentes en el método de clasificación, volumen de datos, número de atributos y proporción de entrenamiento y prueba contribuyen de manera significativa al número de aciertos, y de manera subsecuente, también lo hacen las interacciones de segundo, tercer y cuarto nivel entre ellos.
3. En cuanto a la combinación de factores que produce el mejor porcentaje de aciertos en el proceso de clasificación, esta corresponde a los niveles esperables en cuanto a tamaño de datos disponible para el proceso de entrenamiento (en términos de



---

volumen y conjunto de atributos) y el método de clasificación **Random Forest**, basado en árboles de decisión y métodos agregados de clasificación.

4. Los métodos agregados de clasificación presentaron de manera estricta un mayor porcentaje de aciertos que el clasificador evaluado que no empleaba ningún tipo de agregación en su diseño, siendo el enfoque de *Boosting* el más exitoso entre estos.
5. El análisis en cuanto a la efectividad de los clasificadores sustenta trabajo previo acerca de las mejoras presentadas por algoritmos de inducción de reglas al ser complementadas por enfoque de agregación.
6. Los resultados indican que entre los factores estudiados, el volumen de datos es el que parece influir menos dentro del proceso de clasificación, sugiriendo la existencia de un crecimiento que indica pocos beneficios a la arbitrariedad en cuanto al tamaño del conjunto de datos de entrada, en contraste significativo al factor contemplando los subconjuntos de atributos empleados para el entrenamiento y prueba.

## 5.1 Aportes

El trabajo realizado en esta investigación cumplió con los entregables establecidos:

1. Una implementación básica de un algoritmo de inducción de reglas, complemento dentro del ambiente de desarrollo scikit-learn. Esta implementación se encuentra de manera completa en <https://github.com/Fireblend/scikit-learn>. Un extracto se presenta en el apéndice.
2. Una implementación de uso de los clasificadores, utilizando el sistema Weka [18] para la ejecución del experimento.
3. Los programas auxiliares para la ejecución y el control de los experimentos. Estos están disponibles en el repositorio <https://github.com/Fireblend/thesis-auxiliary>. Un extracto de los mismos se presenta en el apéndice.
4. El análisis estadístico de los experimentos.
5. Un artículo científico, que ha sido sometido a revisión para participación en AAAI-17, la trigésimo-primer conferencia de AAAI de Inteligencia Artificial. Este puede encontrarse adjunto al presente documento en el apéndice.

## 5.2 Trabajo futuro

Posterior a la realización del presente esfuerzo investigativo, surgen nuevas preguntas y quedan abiertos varios caminos para plantear nuevos proyectos. Se proponen a continuación varias ideas al rededor de las cuales trabajos futuros podrían desarrollarse:

1. Evaluar de manera más exhaustiva la lista de reglas generadas por el modelo de clasificación de MODLEM, de tal forma que puedan generarse conclusiones valiosas para la comunidad científica que busca refinar el proceso de clasificación de eventos de cruce detectados por el proyecto Kepler.
2. Construir un clasificador basado en inducción de reglas implementando métodos de agregación aleatorios con selección de sub-atributos de manera más compleja que el elaborado durante el desarrollo de esta investigación, buscando igualar o mejorar el desempeño mostrado por el enfoque de agregación basada en *Boosting* sobre el clasificador de inducción de reglas base.
3. Explorar alternativas de optimización en cuanto a recursos tanto computacionales como temporales para los algoritmos basados en inducción de reglas.
4. La elaboración de métodos de preprocesamiento u optimización de atributos previo al proceso de clasificación de muestras provenientes del proyecto Kepler podrían incrementar la cantidad de aciertos en el proceso de clasificación.
5. Puede también ser de interés la caracterización del subconjunto de atributos opcionales considerados dentro de este proyecto, de tal forma que se puedan determinar las causas para su significancia en los resultados del proceso de clasificación.

---

## Referencias

---

- [1] N. M. Batalha, W. J. Borucki, D. G. Koch, S. T. Bryson, M. R. Haas, T. M. Brown, D. A. Caldwell, J. R. Hall, R. L. Gilliland, D. W. Latham, S. Meibom, and D. G. Monet. Selection, Prioritization, and Characteristics of Kepler Target Stars. *apjl*, 713:L109–L114, April 2010. doi: 10.1088/2041-8205/713/2/L109.
- [2] Natalie M. Batalha. Exploring exoplanet populations with nasa’s kepler mission. *Proceedings of the National Academy of Sciences*, 111(35):12647–12654, 2014. doi: 10.1073/pnas.1304196111. URL <http://dx.doi.org/10.1073/pnas.1304196111>.
- [3] W. J. Borucki, D. Koch, G. Basri, T. Brown, D. Caldwell, E. Devore, E. Dunham, T. Gautier, J. Geary, R. Gilliland, A. Gould, S. Howell, and J. Jenkins. The Kepler Mission: A Transit-Photometry Mission to Discover Terrestrial Planets. *ISSI Scientific Reports Series*, 6:207–220, 2006.
- [4] W. J. Borucki, D. Koch, G. Basri, N. Batalha, T. Brown, D. Caldwell, J. Caldwell, J. Christensen-Dalsgaard, W. D. Cochran, E. DeVore, E. W. Dunham, A. K. Dupree, T. N. Gautier, J. C. Geary, R. Gilliland, A. Gould, S. B. Howell, J. M. Jenkins, Y. Kondo, D. W. Latham, G. W. Marcy, S. Meibom, H. Kjeldsen, J. J. Lissauer, D. G. Monet, D. Morrison, D. Sasselov, J. Tarter, A. Boss, D. Brownlee, T. Owen, D. Buzasi, D. Charbonneau, L. Doyle, J. Fortney, E. B. Ford, M. J. Holman, S. Seager, J. H. Steffen, W. F. Welsh, J. Rowe, H. Anderson, L. Buchhave, D. Ciardi, L. Walkowicz, W. Sherry, E. Horch, H. Isaacson, M. E. Everett, D. Fischer, G. Torres, J. A. Johnson, M. Endl, P. MacQueen, S. T. Bryson, J. Dotson, M. Haas, J. Kolodziejczak, J. Van Cleve, H. Chandrasekaran, J. D. Twicken, E. V. Quintana, B. D. Clarke, C. Allen, J. Li, H. Wu, P. Tenenbaum, E. Verner, F. Bruhweiler, J. Barnes, and A. Prsa. Kepler Planet-Detection Mission: Introduction and First Results. *Science*, 327:977–, February 2010. doi: 10.1126/science.1185402.
- [5] Leo Breiman. Stacked regressions. *Machine Learning*, 24(1):49–64, 1996. ISSN

- 1573-0565. doi: 10.1007/BF00117832. URL <http://dx.doi.org/10.1007/BF00117832>.
- [6] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL <http://dx.doi.org/10.1023/A%3A1010933404324>.
- [7] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL <http://dx.doi.org/10.1023/A%3A1010933404324>.
- [8] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL <http://dx.doi.org/10.1023/A%3A1010933404324>.
- [9] Jason Brownlee. Improve machine learning results with boosting, bagging and blending ensemble methods in weka, 2014. URL <http://machinelearningmastery.com/improve-machine-learning-results-with-boosting-bagging-and-blending-ensemble-methods-in-weka/>. [Internet; consultado 14-octubre-2014].
- [10] S. T. Bryson, P. Tenenbaum, J. M. Jenkins, H. Chandrasekaran, T. Klaus, D. A. Caldwell, R. L. Gilliland, M. R. Haas, J. L. Dotson, D. G. Koch, and W. J. Borucki. The Kepler Pixel Response Function. , 713:L97–L102, April 2010. doi: 10.1088/2041-8205/713/2/L97.
- [11] Kepler Science Center. Astrophysics publications, 2014. URL <http://keplerscience.arc.nasa.gov/PublicationsAstrophysics.shtml>. [Internet; consultado 15-octubre-2014].
- [12] Sogang University Data Mining Research Lab. Ensemble learning, 2014. URL [http://mllab.sogang.ac.kr/?mid=research\\_subj\\_el](http://mllab.sogang.ac.kr/?mid=research_subj_el). [Internet; consultado 15-octubre-2014].
- [13] Arun Debray and Raymond Wu. Astronomical implications of machine learning, 2013.
- [14] Thomas G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, pages 1–15, London, UK, UK, 2000. Springer-Verlag. ISBN 3-540-67704-6. URL <http://dl.acm.org/citation.cfm?id=648054.743935>.
- [15] Ronald L. Gilliland, William J. Chaplin, Edward W. Dunham, Vic S. Argabright, William J. Borucki, et al.
- [16] Jerzy W Grzymala-Busse and Jerzy Stefanowski. Three discretization methods for rule induction. *International Journal of Intelligent Systems*, 16(1):29–38, 2001.

- [17] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009. ISSN 1541-1672. doi: <http://doi.ieeecomputersociety.org/10.1109/MIS.2009.36>.
- [18] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009. ISSN 1931-0145. doi: 10.1145/1656274.1656278. URL <http://doi.acm.org/10.1145/1656274.1656278>.
- [19] Todd Holloway. Ensemble machine learning tutorial, 2014. URL <http://abeautifulwww.com/EnsembleLearning.pdf>. [Internet; consultado 14-octubre-2014].
- [20] J. M. Jenkins. The Impact of Solar-like Variability on the Detectability of Transiting Terrestrial Planets. , 575:493–505, August 2002. doi: 10.1086/341136.
- [21] Jon M. Jenkins. The impact of solar-like variability on the detectability of transiting terrestrial planets. *The Astrophysical Journal*, 575(1):493, 2002. URL <http://stacks.iop.org/0004-637X/575/i=1/a=493>.
- [22] Jon M. Jenkins, Douglas A. Caldwell, Hema Chandrasekaran, Joseph D. Twicken, Stephen T. Bryson, Elisa V. Quintana, Bruce D. Clarke, Jie Li, Christopher Allen, Peter Tenenbaum, Hayley Wu, Todd C. Klaus, Christopher K. Middour, Miles T. Cote, Sean McCauliff, Forrest R. Girouard, Jay P. Gunter, Bill Wohler, Jeneen Sommers, Jennifer R. Hall, AKM K. Uddin, Michael S. Wu, Paresh A. Bhavsar, Jeffrey Van Cleve, David L. Pletcher, Jessie A. Dotson, Michael R. Haas, Ronald L. Gilliland, David G. Koch, and William J. Borucki. Overview of the kepler science processing pipeline. *The Astrophysical Journal Letters*, 713(2):L87, 2010. URL <http://stacks.iop.org/2041-8205/713/i=2/a=L87>.
- [23] Joseph H. Catanzarite. Autovetter planet candidate catalog for q1-q17 data release 24. July 2015. URL <http://exoplanetarchive.ipac.caltech.edu/docs/KSCI-19091-001.pdf>.
- [24] Michael Kearns. Thoughts on hypothesis boosting. *Unpublished manuscript*, 1988.
- [25] Roger E Kirk. *Experimental design*. Wiley Online Library, 1982.
- [26] D. G. Koch, W. J. Borucki, G. Basri, N. M. Batalha, T. M. Brown, D. Caldwell, J. Christensen-Dalsgaard, W. D. Cochran, E. DeVore, E. W. Dunham, T. N. Gautier, III, J. C. Geary, R. L. Gilliland, A. Gould, J. Jenkins, Y. Kondo, D. W. Latham, J. J. Lissauer, G. Marcy, D. Monet, D. Sasselov, A. Boss, D. Brownlee, J. Caldwell, A. K. Dupree, S. B. Howell, H. Kjeldsen, S. Meibom, D. Morrison, T. Owen, H. Reitsema, J. Tarter, S. T. Bryson, J. L. Dotson, P. Gazis, M. R. Haas, J. Kolodziejczak, J. F. Rowe, J. E. Van Cleve, C. Allen, H. Chandrasekaran, B. D. Clarke, J. Li, E. V. Quintana, P. Tenenbaum, J. D. Twicken, and H. Wu. Kepler Mission Design, Realized Photometric Performance, and Early Science. 713:L79, April 2010. doi: 10.1088/2041-8205/713/2/L79.

- [27] David G. Koch, William J. Borucki, Jason F. Rowe, Natalie M. Batalha, Timothy M. Brown, Douglas A. Caldwell, John Caldwell, William D. Cochran, Edna DeVore, Edward W. Dunham, Andrea K. Dupree, Thomas N. Gautier III, John C. Geary, Ron L. Gilliland, Steve B. Howell, Jon M. Jenkins, David W. Latham, Jack J. Lissauer, Geoff W. Marcy, David Morrison, and Jill Tarter. Discovery of the transiting planet kepler-5b. *The Astrophysical Journal Letters*, 713(2):L131, 2010. URL <http://stacks.iop.org/2041-8205/713/i=2/a=L131>.
- [28] H. Levene. Robust tests for equality of variances. In I. Olkin, editor, *Contributions to Probability and Statistics*. Stanford University Press, 1960.
- [29] Dawson Rebekah I. Lissauer, Jack J. and Scott Tremaine. Advances in exoplanet science from kepler. *Nature Insight*, 513(7518):336–344, 2014. doi: 10.1038/nature13781. URL <http://dx.doi.org/10.1038/nature13781>.
- [30] K. Mandel and E. Agol. Analytic Light Curves for Planetary Transit Searches. , 580:L171–L175, December 2002. doi: 10.1086/345520.
- [31] Sean McCauliff, Jon M. Jenkins, Joseph Catanzarite, Christopher J. Burke, Jeffrey L. Coughlin, Joseph D. Twicken, Peter Tenenbaum, Shawn Seader, Jie Li, and Miles Cote. Automatic classification of kepler threshold crossing events, 2014.
- [32] Olena Medelyan, Steve Manion, Jeen Broekstra, Anna Divoli, Anna-Lan Huang, and Ian H. Witten. Constructing a focused taxonomy from a document collection. In *Proc 10th European Semantic Web Conference*, Montpellier, France, pages 367–381. Springer, 2013.
- [33] Douglas Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, 2001. ISBN 0471316490.
- [34] Douglas C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, 2001. ISBN 0471316490.
- [35] NASA. Nasa’s kepler mission announces a planet bonanza, 715 new worlds, 2014. URL [http://www.nasa.gov/ames/kepler/nasas-kepler-mission-announces-a-planet-bonanza/#.VDNcM\\_ldV8E](http://www.nasa.gov/ames/kepler/nasas-kepler-mission-announces-a-planet-bonanza/#.VDNcM_ldV8E). [Internet; consultado 6-octubre-2014].
- [36] NASA. Q1-q17 data release 24, 2015.
- [37] Mary Natrella. *NIST/SEMATECH e-Handbook of Statistical Methods*. NIST/SEMATECH, July 2010. URL <http://www.itl.nist.gov/div898/handbook/>.
- [38] J. A. Orosz, W. F. Welsh, J. A. Carter, D. C. Fabrycky, W. D. Cochran, M. Endl, E. B. Ford, N. Haghighipour, P. J. MacQueen, T. Mazeh, R. Sanchis-Ojeda, D. R. Short, G. Torres, E. Agol, L. A. Buchhave, L. R. Doyle, H. Isaacson, J. J. Lissauer, G. W. Marcy, A. Shporer, G. Windmiller, T. Barclay, A. P. Boss, B. D. Clarke,

- J. Fortney, J. C. Geary, M. J. Holman, D. Huber, J. M. Jenkins, K. Kinemuchi, E. Kruse, D. Ragozzine, D. Sassellov, M. Still, P. Tenenbaum, K. Uddin, J. N. Winn, D. G. Koch, and W. J. Borucki. Kepler-47: A Transiting Circumbinary Multiplanet System. *Science*, 337:1511–, September 2012. doi: 10.1126/science.1228380.
- [39] R Project. What is r, 2013. URL <http://www.r-project.org/about.html>. [Online; accessed 25-May-2013].
- [40] Antti Puurula. Cumulative progress in language models for information retrieval. In *Proc 11th Australasian Language Technology Workshop*, Brisbane, Australia, pages 96–100. ACL, 2013.
- [41] Jeremy Rajanayagam, Eibe Frank, Ross W. Shepherd, and Peter J. Lewindon. Artificial neural network is highly predictive of outcome in paediatric acute liver failure. *Pediatric Transplantation*, 2013.
- [42] Giovanni Sanabria Brenes. *Comprendiendo la estadística inferencial*. Editorial Tecnológica de Costa Rica, 2011. ISBN 78-9977-66-238-1.
- [43] Robert E. Schapire. A brief introduction to boosting. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI’99, pages 1401–1406, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. URL <http://dl.acm.org/citation.cfm?id=1624312.1624417>.
- [44] F. Stahl and Max Bramer. Random prism: an alternative to random forests. In *Thirty-first SGAI International Conference on Artificial Intelligence*, 2012. URL <http://eprints.port.ac.uk/4901/>.
- [45] Jerzy Stefanowski. *Transactions on Rough Sets VI: Commemorating the Life and Work of Zdzisław Pawlak, Part I*, chapter On Combined Classifiers, Rule Induction and Rough Sets, pages 329–350. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-71200-8. doi: 10.1007/978-3-540-71200-8\_18. URL [http://dx.doi.org/10.1007/978-3-540-71200-8\\_18](http://dx.doi.org/10.1007/978-3-540-71200-8_18).
- [46] H.C. Thode Jr. *Testing for Normality*. Marcel Dekker, 2002.
- [47] Joseph D. Twicken Todd C. Klaus. Kepler: Nasa’s search for habitable earth-size planets, 2011. URL <http://www.nas.nasa.gov/SC11/demos/demo17.html>. [Internet; consultado 15-octubre-2014].
- [48] KAGAN TUMER and JOYDEEP GHOSH. Error correlation and error reduction in ensemble classifiers. *Connection Science*, 8(3-4):385–404, 1996. doi: 10.1080/095400996116839. URL <http://dx.doi.org/10.1080/095400996116839>.
- [49] V. P. Tuzlukov. *Signal processing noise*. CRC Press, Boca Raton, 2002. ISBN 9781420041118.



- [50] L. Walkowicz, A. R. Howe, R. Nayar, E. L. Turner, J. Scargle, V. Meadows, and A. Zee. Mining the Kepler Data using Machine Learning. In *American Astronomical Society Meeting Abstracts #223*, volume 223 of *American Astronomical Society Meeting Abstracts*, page #146.04, January 2014.
- [51] R.E. Walpole, R.H. Myers, and S.L. Myers. *Probability and Statistics for Engineers and Scientists*. Pearson Education, Limited, 2010. ISBN 9780321629111. URL <http://books.google.co.cr/books?id=tzZxRQAACAAJ>.
- [52] Wikipedia. Ensemble machine learning, 2003. URL [http://www.wikiwand.com/en/Ensemble\\_learning](http://www.wikiwand.com/en/Ensemble_learning). [Internet; consultado 14-octubre-2014].
- [53] Wikipedia. Exoplanet detection methods, 2014. URL [http://www.wikiwand.com/en/Exoplanet#Detection\\_methods](http://www.wikiwand.com/en/Exoplanet#Detection_methods). [Internet; consultado 5-octubre-2014].
- [54] Wikipedia. Kepler (spacecraft), 2014. URL [http://en.wikipedia.org/wiki/Kepler\\_\(spacecraft\)](http://en.wikipedia.org/wiki/Kepler_(spacecraft)). [Internet; consultado 5-octubre-2014].
- [55] David H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

## Apéndice A

---

Código Auxiliar

---

```
import java.io.*;
import java.nio.charset.Charset;
import java.util.HashMap;
import java.util.Random;

/**
 * Created by sergio on 16/02/16.
 */
public class Splitter {
    static Random random = new Random();
    static String[] files all = {"all", "all no pix"};
    static String[] files half = {"half", "half no pix"};
    static String root = "/home/sergio/DATASETS/";

    public static void main(String[] args){
        for(int i=0; i< files all.length; i++) {
            String file = files all[i];
            split(5717/3, 75, 1906/3, 25, file);
            split(3811/3, 50, 3811/3, 50, file);
            split(1906/3, 25, 5717/3, 75, file);
        }
        for(int i=0; i< files half.length; i++) {
            String file = files half[i];
            split(2857/3, 75, 953/3, 25, file);
            split(1905/3, 50, 1905/3, 50, file);
            split(953/3, 25, 2857/3, 75, file);
        }
    }
    //Big samples = 7623 5717/1906 == 3811/3811
    //Low samples = 3810 2857/953 == 1905/1905

    public static void split(int train, int trainp, int test, int testp, String filename){
        try {
            PrintWriter trainWriter = new PrintWriter(root+"train/"+filename+" "+trainp+" train.arff", "UTF-8");
            PrintWriter testWriter = new PrintWriter(root+"test/"+filename+" "+testp+" test.arff", "UTF-8");

            InputStream fis = new FileInputStream(root+filename+".arff");
            InputStreamReader isr = new InputStreamReader(fis, Charset.forName("UTF-8"));
            BufferedReader br = new BufferedReader(isr);

            String line;
            String PC = "PC";
            String AFP = "AFP";
            String NTP = "NTP";

            HashMap<String, Integer> trainBins = new HashMap<>();
            trainBins.put(PC, train);
            trainBins.put(AFP, train);
            trainBins.put(NTP, train);
            HashMap<String, Integer> testBins = new HashMap<>();
            testBins.put(PC, test);
            testBins.put(AFP, test);
            testBins.put(NTP, test);

            while ((line = br.readLine()) != null) {
                if(line.endsWith(PC) || line.endsWith(AFP) || line.endsWith(NTP)){
                    if(line.endsWith(PC)){
                        if(trainBins.get(PC) != 0 && testBins.get(PC) != 0){
                            if(random.nextInt(1) == 1){
                                trainBins.put(PC, trainBins.get(PC)-1);
                                trainWriter.println(line);
                            }
                            else {
                                testBins.put(PC, testBins.get(PC)-1);
                                testWriter.println(line);
                            }
                        }
                    }
                    else if(trainBins.get(PC) != 0){
                        trainBins.put(PC, trainBins.get(PC)-1);
                        trainWriter.println(line);
                    }
                    else if(testBins.get(PC) != 0){
                        testBins.put(PC, testBins.get(PC)-1);
                        testWriter.println(line);
                    }
                }
                if(line.endsWith(AFP)){
                    if(trainBins.get(AFP) != 0 && testBins.get(AFP) != 0){
                        if(random.nextInt(1) == 1){
                            trainBins.put(AFP, trainBins.get(AFP)-1);
                            trainWriter.println(line);
                        }
                    }
                }
            }
        }
    }
}
```

```
        else {
            testBins.put(AFP, testBins.get(AFP)-1);
            testWriter.println(line);
        }
    }
    else if(trainBins.get(AFP) != 0){
        trainBins.put(AFP, trainBins.get(AFP)-1);
        trainWriter.println(line);
    }
    else if(testBins.get(AFP) != 0){
        testBins.put(AFP, testBins.get(AFP)-1);
        testWriter.println(line);
    }
}

if(line.endsWith(NTP)){
    if(trainBins.get(NTP) != 0 && testBins.get(NTP) != 0){
        if(random.nextInt(1) == 1){
            trainBins.put(NTP, trainBins.get(NTP)-1);
            trainWriter.println(line);
        }
        else {
            testBins.put(NTP, testBins.get(NTP)-1);
            testWriter.println(line);
        }
    }
    else if(trainBins.get(NTP) != 0){
        trainBins.put(NTP, trainBins.get(NTP)-1);
        trainWriter.println(line);
    }
    else if(testBins.get(NTP) != 0){
        testBins.put(NTP, testBins.get(NTP)-1);
        testWriter.println(line);
    }
}

}
else {
    trainWriter.println(line);
    testWriter.println(line);
}
}

trainWriter.close();
testWriter.close();

} catch (Exception e) {

}

}
```

```
/**
 * Created by sergio on 13/02/16.
 */
import weka.classifiers.Classifier;
import weka.classifiers.Evaluation;
import weka.classifiers.bayes.NaiveBayes;
import weka.core.Attribute;
import weka.core.FastVector;
import weka.core.Instance;
import weka.core.Instances;

public class Experiment {

    public static void main(String[] args) throws Exception{

        // Declare two numeric attributes
        Attribute Attribute1 = new Attribute("firstNumeric");
        Attribute Attribute2 = new Attribute("secondNumeric");

        // Declare a nominal attribute along with its values
        FastVector fvNominalVal = new FastVector(3);
        fvNominalVal.addElement("blue");
        fvNominalVal.addElement("gray");
        fvNominalVal.addElement("black");
        Attribute Attribute3 = new Attribute("aNominal", fvNominalVal);

        // Declare the class attribute along with its values
        FastVector fvClassVal = new FastVector(2);
        fvClassVal.addElement("positive");
        fvClassVal.addElement("negative");
        Attribute ClassAttribute = new Attribute("theClass", fvClassVal);

        // Declare the feature vector
        FastVector fvWekaAttributes = new FastVector(4);
        fvWekaAttributes.addElement(Attribute1);
        fvWekaAttributes.addElement(Attribute2);
        fvWekaAttributes.addElement(Attribute3);
        fvWekaAttributes.addElement(ClassAttribute);

        // Create an empty training set
        Instances isTrainingSet = new Instances("Rel", fvWekaAttributes, 10);

        // Set class index
        isTrainingSet.setClassIndex(3);

        // Create the instance
        Instance iExample = new Instance(4);
        iExample.setValue((Attribute) fvWekaAttributes.elementAt(0), 1.0);
        iExample.setValue((Attribute) fvWekaAttributes.elementAt(1), 0.5);
        iExample.setValue((Attribute) fvWekaAttributes.elementAt(2), "gray");
        iExample.setValue((Attribute) fvWekaAttributes.elementAt(3), "positive");

        // add the instance
        isTrainingSet.add(iExample);
        Classifier cModel = (Classifier) new NaiveBayes();
        cModel.buildClassifier(isTrainingSet);

        // Test the model
        Evaluation eTest = new Evaluation(isTrainingSet);
        eTest.evaluateModel(cModel, isTrainingSet);

        // Print the result à la Weka explorer:
        String strSummary = eTest.toSummaryString();
        System.out.println(strSummary);

        // Get the confusion matrix
        double[][] cmMatrix = eTest.confusionMatrix();
        for(int row_i=0; row_i<cmMatrix.length; row_i++){
            for(int col_i=0; col_i<cmMatrix.length; col_i++){
                System.out.print(cmMatrix[row_i][col_i]);
                System.out.print("|");
            }
            System.out.println();
        }
    }
}
```

```
import modules.ModlemModule;
import modules.Module;
import modules.RandomModule;
import weka.classifiers.Evaluation;
import weka.core.Instances;
import weka.core.converters.ConverterUtils;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Random;

/**
 * Created by sergio on 14/02/16.
 */
public class ExperimentMain {

    public static void main(String[] args){

        String algorithm = args[0];
        String size = args[1];
        String attrs = args[2];
        String split = args[3];
        String jobid = args[4];

        String trainfile = "train/"+size;
        String testfile = "test/"+size;

        if(attrs.equals("no pix")){
            trainfile += " no pix ";
            testfile += " no pix ";
        }
        if(split.equals("25")){
            trainfile += "25 train.arff";
            testfile += "75 test.arff";
        }
        else if(split.equals("50")){
            trainfile += "50 train.arff";
            testfile += "50 test.arff";
        }
        else if(split.equals("25")){
            trainfile += "75 train.arff";
            testfile += "25_test.arff";
        }
        }

        Instances trainSet;
        Instances testSet;

        try {
            ConverterUtils.DataSource trainSource = new ConverterUtils.DataSource(trainfile);
            ConverterUtils.DataSource testSource = new ConverterUtils.DataSource(testfile);

            trainSet = trainSource.getDataSet();
            testSet = testSource.getDataSet();

            trainSet.setClassIndex(trainSet.numAttributes() - 1);
            testSet.setClassIndex(testSet.numAttributes() - 1);

        } catch (Exception e) {

        }

        }

        long startTime = System.currentTimeMillis();
        Evaluation eval = algo.executeExperiment(datasets);
        long stopTime = System.currentTimeMillis();
        long runtime = stopTime - startTime;

    }
}
```

```
#!/usr/bin/python
import time
import os
import commands
import re
```

```
def waitForJob(outputFile):
    while(not (os.path.exists(outputFile))):
        print("Durmiendo por 60s: " + outputFile + " no existe\n")
        time.sleep(10)
```

```
def initQsubs():
    qsubs = []
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="random-forest",size="half",attrs="no_pix",split=25,JOB=1000)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-bagging",size="all",attrs="all",split=75,JOB=1001)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="random-modlem",size="all",attrs="no_pix",split=75,JOB=1002)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-bagging",size="half",attrs="no_pix",split=75,JOB=1003)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-bagging",size="half",attrs="all",split=75,JOB=1004)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem",size="all",attrs="no_pix",split=50,JOB=1005)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-bagging",size="half",attrs="all",split=25,JOB=1006)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="random-forest",size="all",attrs="no_pix",split=25,JOB=1007)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="random-forest",size="half",attrs="no_pix",split=75,JOB=1008)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-bagging",size="half",attrs="no_pix",split=50,JOB=1009)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="random-modlem",size="all",attrs="all",split=25,JOB=1010)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="random-forest",size="half",attrs="all",split=50,JOB=1011)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="random-modlem",size="half",attrs="no_pix",split=25,JOB=1012)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="random-forest",size="all",attrs="all",split=25,JOB=1013)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="random-modlem",size="half",attrs="no_pix",split=75,JOB=1014)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-boosting",size="half",attrs="no_pix",split=50,JOB=1015)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-bagging",size="all",attrs="no_pix",split=25,JOB=1016)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-bagging",size="all",attrs="no_pix",split=50,JOB=1017)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem",size="half",attrs="all",split=50,JOB=1018)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-boosting",size="all",attrs="all",split=25,JOB=1019)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem",size="half",attrs="no_pix",split=25,JOB=1020)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="random-modlem",size="half",attrs="all",split=25,JOB=1021)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem",size="half",attrs="all",split=25,JOB=1022)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-boosting",size="all",attrs="no_pix",split=50,JOB=1023)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-boosting",size="all",attrs="no_pix",split=75,JOB=1024)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-bagging",size="all",attrs="no_pix",split=75,JOB=1025)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-boosting",size="all",attrs="all",split=50,JOB=1026)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="random-modlem",size="all",attrs="all",split=75,JOB=1027)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem",size="all",attrs="all",split=50,JOB=1028)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-boosting",size="half",attrs="all",split=25,JOB=1029)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-bagging",size="half",attrs="no_pix",split=25,JOB=1030)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-boosting",size="all",attrs="all",split=75,JOB=1031)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem",size="all",attrs="no_pix",split=25,JOB=1032)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem",size="all",attrs="no_pix",split=75,JOB=1033)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="random-modlem",size="half",attrs="all",split=75,JOB=1034)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="random-modlem",size="all",attrs="no_pix",split=50,JOB=1035)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="random-forest",size="half",attrs="no_pix",split=50,JOB=1036)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-bagging",size="all",attrs="all",split=50,JOB=1037)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-boosting",size="half",attrs="all",split=50,JOB=1038)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-bagging",size="half",attrs="all",split=50,JOB=1039)
    qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-boosting",size="half",attrs="all",split=75,JOB=1040)
```

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

```
=1289)
qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="random-modlem",size="half",attrs="no pix",split=50,JOB=
=1290)
qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="random-forest",size="all",attrs="no pix",split=50,JOB=
1291)
qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem",size="all",attrs="no pix",split=50,JOB=1292)
qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem",size="all",attrs="all",split=25,JOB=1293)
qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-bagging",size="half",attrs="all",split=25,JOB=
1294)
qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-bagging",size="half",attrs="no pix",split=75,
JOB=1295)
qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-boosting",size="all",attrs="all",split=75,JOB=
1296)
qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-bagging",size="all",attrs="all",split=50,JOB=
1297)
qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-boosting",size="half",attrs="all",split=75,JOB=
1298)
qsubs.append(qsub -q xeon -lnodes=1:ppn=8 -lwalltime=24:00:00 ./kepler.torque -v algo="modlem-bagging",size="all",attrs="all",split=75,JOB=
1299)

return qsubs

# main #
qsubs = initQsubs()
total = len(qsubs)
i = 0
while(i < total):
    print i
    status1, output1 = commands.getstatusoutput(qsubs[i+0])
    status2, output2 = commands.getstatusoutput(qsubs[i+1])
    status3, output3 = commands.getstatusoutput(qsubs[i+2])
    status3, output3 = commands.getstatusoutput(qsubs[i+3])
    status3, output3 = commands.getstatusoutput(qsubs[i+4])
    waitForJob((re.split("=", qsubs[i+0]))[-1] + ".output")
    waitForJob((re.split("=", qsubs[i+1]))[-1] + ".output")
    waitForJob((re.split("=", qsubs[i+2]))[-1] + ".output")
    waitForJob((re.split("=", qsubs[i+3]))[-1] + ".output")
    waitForJob((re.split("=", qsubs[i+4]))[-1] + ".output")
    i += 5
```

---

Código MODLEM

---



```
1 import numpy as np
2 import copy
3 from sklearn.base import BaseEstimator, ClassifierMixin
4 from selection criterion import LaplaceEstimator
5 import selection criterion as sc
6 import classification strategies
7 import classification strategies.m estimate strategy as m estimate strategy
8 import classification strategies.strategy as strategy
9 from dataset mapper import DataSetMapper
10 import rule
11 import numerical class border
12 import condition
13
14 global LAPLACE ESTIMATOR
15
16 class Modlem(BaseEstimator, ClassifierMixin):
17     """Predicts the majority class of its training data."""
18
19     def init (self, dataset attr type=None, nominal categories=None):
20         LAPLACE ESTIMATOR = 1
21         pass
22
23     def fit(self, X, y):
24         self.m data = copy.deepcopy(X)
25         self.m data classes = copy.deepcopy(y)
26         self.m num classes = len(np.unique(self.m data classes))
27
28         #TODO/POLISH: remove instances without classes if any
29
30         #TODO/POLISH: add option to change criterion
31         self.m criterion = LaplaceEstimator(y.shape[0])
32
33         rules = []
34         self.m possible conditions = []
35
36         for instance index in range(0, len(self.m data classes)):
37             self.m data[instance index].append(m data classes[instance index])
38
39         self.m_dataset_mapper = DataSetMapper(self.m_data, self.m_num_classes, self.m_data_classes,
40 self.dataset_attr_type, self.nominal_categories)
41
42         approximate_classes()
43         generate_conditions()
44
45         self.m_current_rule_coverage = bitarray([False]*self.m_data.shape[0])
46         self.m_current_rule_positive_coverage = bitarray([False]*self.m_data.shape[0])
47         covered_from_concept = bitarray([False]*self.m_data.shape[0])
48
49         self.class_index = self.m_data.shape[1];
50
51         for consequent in range(0, self.m_num_classes):
52             self.m_positives = self.m_dataset_mapper.get_bit_set_and_check(self.class_index,
53 self.consequent)
54             covered_from_concept ^= covered_from_concept
55
56             while bitarray.count(covered_from_concept) < bitarray.count(self.m_positives):
57                 self.m_current_rule_coverage = bitarray([True]*self.m_data.shape[0])
58
59                 for x in range(0, len(self.m_current_rule_coverage)):
60                     if self.m_current_rule_coverage[x]:
61                         if covered_from_concept[x]:
62                             self.m_current_rule_coverage[x] = False
63
64                 self.m_current_rule_positive_coverage ^= self.m_current_rule_positive_coverage
65                 self.m_current_rule_positive_coverage |= self.m_positives
66
67                 for x in range(0, len(self.m_current_rule_positive_coverage)):
68                     if self.m_current_rule_positive_coverage[x]:
69                         if covered_from_concept[x]:
70                             self.m_current_rule_positive_coverage[x] = False
71
72                 clean_usage()
73
74                 rule = Rule(self.m_data, consequent)
75
76                 while True:
77                     self.m_criterion.set_worst_eval()
```

```

76         best = best condition()
77
78         if best.get_evaluation() == m.criterion.get_worst_eval():
79             break
80
81         rule.extend(copy.deepcopy(best))
82
83         self.m_current_rule_coverage &= best.get_coverage()
84         self.m_current_rule_positive_coverage &= best.get_coverage()
85
86         mark_as_used(best)
87         if bitarray.count(self.m_current_rule_coverage) <= bitarray.count(self.
            m_current_rule_positive_coverage):
88             break
89
90         rule.drop_redundant_conditions(self.m_positives)
91
92         if rule.get_size() == 0:
93             break
94
95         rule.merge_conditions()
96         rule.calculate_coverage(self.m_dataset_mapper)
97         rules.append(rule)
98
99         current_rule_positive_coverage = deepcopy(self.m_positives)
100         current_rule_positive_coverage &= rule.get_coverage()
101         covered_from_concept |= current_rule_positive_coverage
102
103         remove_redundant_rules(rules, consequent)
104
105         self.m_classification_strategy = Strategy.create()
106         generate_output(rules)
107
108         self.m_data = None
109         self.m_possible_conditions = None
110         self.m_dataset_mapper = None
111
112         return self
113
114     def predict(self, X):
115         return self.m_classification_strategy.classify_instance(X)
116
117     def get_params(self, deep=True):
118         # suppose this estimator has parameters "alpha" and "recursive"
119         return {}
120
121     def set_params(self, **parameters):
122         for parameter, value in parameters.items():
123             self.setattr(parameter, value)
124         return self
125
126     def approximate_classes(self):
127         if self.m_rules_type == NORMAL_APPROXIMATION:
128             return
129
130         for i in range(0, self.m_data.shape[0]):
131             consequent = self.m_data_classes[i]
132             curr_coverage = self.m_dataset_mapper.get_bit_set(self.class_index, consequent)
133
134             for j in range(i+1, self.m_data.shape[0]):
135
136                 if curr_coverage[j] or (not compare(i, j)):
137                     continue
138
139                 if self.m_rules_type == UPPER_APPROXIMATION:
140                     curr_coverage[j] = True
141                     self.m_dataset_mapper.get_bit_set(self.class_index, self.m_data_classes[j])[i] = True
142                 elif self.m_rules_type == LOWER_APPROXIMATION:
143                     curr_coverage[i] = False
144                     self.m_dataset_mapper.get_bit_set(self.class_index, self.m_data_classes[j])[j] = False
145
146     def best_condition(self):
147         best = Condition.create_nominal_condition(-1, [], bitarray(), "None")
148         best.set_evaluation(self.m_criterion.get_worst_eval())
149         best.set_positives_nr(0)

```



```

150         for attributeIndex in range(0, len(self.m_possible_conditions)):
151
152             if len(self.m_possible_conditions[attributeIndex]) == 0:
153                 continue
154
155             if isinstance(self.m_data[0][attributeIndex], float):
156                 best = best_numerical(attributeIndex, best)
157
158             elif isinstance(self.m_data[0][attributeIndex], int):
159                 best = best_nominal(attributeIndex, best)
160
161
162     def best_nominal(self, attribute, best):
163         candidate = Condition.create_nominal_condition(attribute, [], bytearray([True]*self.m_data.
164             shape[0]), attribute)
165         candidate.set_evaluation(self.m_criterion.get_worst_eval())
166
167         current = None
168
169         while True:
170             current = copy.deepcopy(candidate)
171
172             rule_conditions = self.m_possible_conditions[attribute]
173             for i in range(0, len(rule_conditions)):
174                 element = rule_conditions[i]
175                 if element.is_used():
176                     continue
177                 if candidate.contains(element):
178                     continue
179
180                 merged = copy.deepcopy(candidate)
181                 merged.extend(element)
182                 merged.or_op(element.get_coverage())
183
184                 self.m_criterion.evaluate(merged, self.m_current_rule_coverage, self.
185                     m_current_rule_positive_coverage)
186                 if(self.m_criterion.compare(merged, current)):
187                     current = merged
188
189                 if not self.m_criterion.compare(current, candidate):
190                     break
191
192             candidate = current
193
194         return select_better(candidate, best)
195
196     def best_numerical(self, attribute, best):
197         current1 = None
198         current2 = None
199
200         for i in range(0, len(self.m_possible_conditions[attribute]), 2):
201             current1 = self.m_possible_conditions.get(attribute).get(i)
202             current2 = self.m_possible_conditions.get(attribute).get(i+1)
203
204             is_lesser_cond_intersected = self.m_available_intervals_for_numerical[attribute].
205                 is_intersected(current1.get_interval())
206             is_greater_equal_cond_intersected = self.m_available_intervals_for_numerical[attribute].
207                 is_intersected(current2.get_interval())
208
209             if not (is_lesser_cond_intersected and is_greater_equal_cond_intersected):
210                 continue
211
212             candidate = self.m_criterion.evaluate(current1, current2, self.m_current_rule_coverage,
213                 self.m_current_rule_positive_coverage)
214             best = select_better(candidate, best)
215
216             if(best.get_evaluation() == self.m_criterion.best()):
217                 break
218
219         return best
220
221     def clean_usage(self):
222         for i in range(0, len(self.m_possible_conditions)):
223             if isinstance(self.m_data[0][i], int):
224                 for condition in self.m_possible_conditions[i]:

```

```

222         condition.set used(not bitarray.count(condition.get coverage() & self.
223                             m current rule positive coverage) > 0)
224
225     self.m available intervals for numerical = Interval(self.m data.shape[1])
226     for i in range(0, len(self.m available intervals for numerical)):
227         self.m available intervals for numerical[i] = Interval(-float("inf"), float("inf"
228             ), true)
229
230     def compare(self, left, right):
231         for attribute in range(0, self.m data.shape[1]):
232             if (self.m data[left][attribute] != self.m data[right][attribute]):
233                 return false
234             return true
235
236     def generate_conditions(self):
237         for attribute in range(0, self.m data.shape[1]):
238             if(isinstance(self.m data[0][attribute], float)):
239                 generate numerical(attribute)
240             elif(isinstance(self.m data[0][attribute], int)):
241                 generate nominal(attribute)
242
243     def generate_nominal(self, attribute):
244         conditions = []
245         condition = None
246
247         attribute values = set(self.m data[:,attribute])
248         for i in range(0, len(attribute values)):
249             condition = Condition.create nominal condition(attribute, [float(i)], copy.deepcopy(self
250                 .m dataset mapper.get bit set and check(attribute, i)), attribute)
251
252             if bitarray.count(condition.get coverage()) == 0:
253                 continue
254
255             conditions.append(condition)
256
257         self.m possible conditions.append += conditions
258
259     def generate_numerical(self, attribute):
260         values = self.m_data[:,attribute]
261         sorted_indexes = np.argsort(values)
262         values = np.sort(values)
263
264         points = []
265         temp = NumericalClassBorder(values[sorted_indexes[0]])
266         temp.add(self.m_data_classes[sorted_indexes[0]])
267
268         points.append(temp)
269
270         for i in range(1, self.m_data.shape[0]):
271             index = sorted_indexes[i]
272             value = values[index]
273             consequent = self.m_data_classes[index]
274
275             if(temp.m_value != value):
276                 temp = NumericalClassBorder(value)
277                 points.append(temp)
278
279             temp.add(consequent)
280
281         mask = bitarray([False] * len(self.m_data.shape[0]))
282         neg_mask = bitarray([False] * len(self.m_data.shape[0]))
283
284         conditions = []
285         attribute_keys = self.m_dataset_mapper.get_keys_for_attribute(attribute)
286
287         for i in range(1, len(points)):
288             less_key = float("inf")
289             current_key = float(points[i].m_value)
290             previous_key = float(points[i-1].m_value)
291
292             for key in attribute_keys:
293                 less_key = key
294                 if not (less_key < current_key):
295                     break
296
297             mask |= self.m_dataset_mapper.get_bit_set(attribute, less_key)

```

```

296
297         if points[i-1].m has ambiguous consequent or points[i].m has ambiguous consequent or
points[i-1].m consequent != points[i].m consequent:
298             border = (previous key + current key) / 2.0
299
300             neg mask.setAll(True)
301             neg mask ^= mask
302
303             interval1 = Interval(-float("inf"), true, border, false)
304             interval2 = Interval(border, true, float("inf"), true)
305             condition1 = Condition.create numeric condition(attribute, interval1, copy.deepcopy(
mask))
306             condition2 = Condition.create numeric condition(attribute, interval2, copy.deepcopy(
neg mask))
307
308             conditions.append(condition1)
309             conditions.append(condition2)
310
311             mask |= self.m dataset mapper(attribute, less key)
312
313             self.m possible conditions.append(conditions)
314
315     def generate_output(self, rules):
316         i = 1
317         for rule in rules:
318             covered = bitarray.count(rule.get coverage())
319             positive coverage = rule.get class coverage()[int(rule.get consequent)]
320
321             covered positives = float( float(positive coverage) / float(bitarray.count(
m dataset mapper.get bit set(self.m data.shape[1], rule.get consequent()))))
322
323             self.m string += "Rule " + i + ". " + rule.to string() + " (" \
324                 + positive coverage + "/" + covered + ", " \
325                 + '%.2f' % (coveredPositives * 100) + "%)\n"
326
327             i += 1
328             self.m string += "\nNumber of rules: " + len(rules) + "\n"
329
330     def mark_as_used(self, p_best):
331         attribute = p_best.get_attribute()
332
333         if isinstance(m_data[0][attribute], float):
334             condition = p_best
335             self.m available_intervals_for_numerical[attribute] = self.
m_available_intervals_for_numerical[attribute].intersect(condition.get_interval())
336
337         for attribute in range(0, len(self.m possible conditions)):
338             if isinstance(self.m_data[0][attribute], int):
339                 for condition in self.m possible_conditions[attribute]:
340                     if (not condition.is_used()) and (not bitarray.count(condition.get_coverage() &
self.m_current_rule_positive_coverage) > 0):
341                         condition.set_used(True)
342
343     def remove_redundant_rules(self, rules, p_consequent):
344         temporary_coverage = None
345
346         for i in range(0, len(rules)):
347             if rules[i].get_consequent == p_consequent:
348                 temporary_coverage = bitarray([False] * self.m_data.shape[0])
349                 for j in range(0, len(rules)):
350                     if rules[j].get_consequent() != p_consequent or i == j:
351                         continue
352
353                     temporary_coverage |= rules[j].get_coverage()
354
355                 temporary_coverage &= self.m_positives
356
357                 if bitarray.count(self.m_positives) == bitarray.count(temporary_coverage):
358                     del rules[i]
359                     continue
360
361     def select_better(self, candidate, best):
362         if candidate.get_evaluation() == best.get_evaluation():
363             if candidate.get_positives_nr() > best.get_positives_nr():
364                 best = candidate
365
366         elif self.m_criterion.compare(candidate, best):

```

```
367         best = candidate
368
369     return best
370
```

---

Muestra de Datos

---

@relation tce\_features-weka.filters.unsupervised.instance.Randomize-S42

@attribute tce\_period numeric  
@attribute tce\_timebkk numeric  
@attribute tce\_ror numeric  
@attribute tce\_dor numeric  
@attribute tce\_incl numeric  
@attribute tce\_impact numeric  
@attribute tce\_duration numeric  
@attribute tce\_ingress numeric  
@attribute tce\_depth numeric  
@attribute tce\_ldm\_coeff1 numeric  
@attribute tce\_ldm\_coeff2 numeric  
@attribute tce\_ldm\_coeff3 numeric  
@attribute tce\_ldm\_coeff4 numeric  
@attribute tce\_num\_transits numeric  
@attribute tce\_full\_conv numeric  
@attribute tce\_model\_snr numeric  
@attribute tce\_model\_chisq numeric  
@attribute tce\_model\_dof numeric  
@attribute tce\_robstat numeric  
@attribute tce\_dof1 numeric  
@attribute tce\_dof2 numeric  
@attribute tce\_chisq1 numeric  
@attribute tce\_chisq2 numeric  
@attribute tce\_prad numeric  
@attribute tce\_sma numeric  
@attribute tce\_eqt numeric  
@attribute tce\_nkoi numeric  
@attribute tce\_steff numeric  
@attribute tce\_slogg numeric  
@attribute tce\_smet numeric  
@attribute tce\_sradius numeric  
@attribute tce\_max\_sngle\_ev numeric  
@attribute tce\_max\_mult\_ev numeric  
@attribute tce\_minmes numeric  
@attribute tce\_mesmad numeric  
@attribute tce\_bin\_oeqp\_stat numeric  
@attribute tce\_mesmad\_numeric  
@attribute tce\_rsnmes numeric  
@attribute tce\_rminmes numeric  
@attribute tce\_fm\_stat numeric  
@attribute tce\_fm\_sra numeric  
@attribute tce\_fm\_sdec numeric  
@attribute tce\_fm\_srao numeric  
@attribute tce\_fm\_sdeco numeric  
@attribute tce\_fm\_prao numeric  
@attribute tce\_fm\_pdeco numeric  
@attribute tce\_dicco\_mira numeric  
@attribute tce\_dicco\_mdec numeric  
@attribute tce\_dikco\_msky numeric  
@attribute tce\_dikco\_mdec numeric  
@attribute av\_training\_set {PC,AFP,NTP}

@data  
30.433,131.979,0.018703,54.39536,89.11,0.8436,2.44,0.1489,329.6,0.377043,0.83333,-0.83167,0.292309,48,1,7.532,1032,1257,8.733,624,47,626.6,80.39,2.959,0.2139,809,0,7043,4.26,-0.02,1.45,7.233,8.235,-4.3  
851.1,1.262,0.06506,6.524,0.9146,0.5891,14.599,19.708212,42.578204,-0.3368,-1.066,0.000067,0.000327,-0.013169,0.054841,-0.12361,-0.077283,0.14578,NTP  
31.8249,144.549,0.007957,22.560774,87.88,0.8361,6.067,0.1563,59.96,0.135723,1.320108,-1.219789,0.412536,41,1,7.022,1886,2267,7.683,520,39,253,52.66,0.7354,0.191,589,2.6338,4.54,-0.8,0.847,4.078,7.686,3  
-2.683,0.6084,0.1492,12.63,0.9135,0.3491,0.91601,19.543377,41.729353,-0.5204,0.3711,0.00002,-0.000044,0.92971,-0.82798,1.245,0.75316,-1.0982,1.3317,PC  
1.41158,132.652,0.015382,1.007744,18,0.9584,3.587,0.5769,184.6,0.307518,1.031472,-1.068696,0.383446,0.919,1,14.52,16720,19300,20.95,12100,930,28150,7709,2.894,0.02783,2520,0,7261,4.12,-0.22,1.724,6.194,3  
15.96,-3.104,2.749,0,5.805,0.9099,0.1945,7.3703,19.505287,38.715528,-0.02583,0.173,0.000015,-0.000032,-0.016957,-0.04833,0.051218,0.05684,-0.06097,0.083355,AFP

[illegible]





---

## Muestra de Modelo de Clasificación MODLEM

---

--Full Classification Model--

Rule 1. (tce\_nkoi >= 2.5)&(tce\_smet >= -0.25) => (av\_training\_set = PC) (229/229, 18.03%)  
Rule 2. (tce\_nkoi >= 1.5)&(tce\_fwm\_sdec >= 40.68)&(tce\_ingress >= 0.02) => (av\_training\_set = PC) (401/401, 31.57%)  
Rule 3. (tce\_nkoi >= 1.5)&(tce\_mesmad < 0.67)&(tce\_robsat < 93.67) => (av\_training\_set = PC) (129/129, 10.16%)  
Rule 4. (tce\_nkoi >= 1.5)&(tce\_dicco\_msky < 0.46)&(tce\_steff < 6340.5) => (av\_training\_set = PC) (346/346, 27.24%)  
Rule 5. (tce\_sradius < 0.51)&(tce\_rminmes < 0.2)&(tce\_ror < 0.11) => (av\_training\_set = PC) (36/36, 2.83%)  
Rule 6. (tce\_rminmes < 0.19)&(tce\_fwm\_sdec >= 43)&(tce\_chisq2 < 1632) => (av\_training\_set = PC) (336/336, 26.46%)  
Rule 7. (tce\_minmes >= -3.55)&(tce\_smet >= 0.23)&(tce\_model\_snr >= 17.28) => (av\_training\_set = PC) (48/48, 3.78%)  
Rule 8. (tce\_minmes >= -3.55)&(tce\_fwm\_sra >= 20.03) => (av\_training\_set = PC) (13/13, 1.02%)  
Rule 9. (tce\_chisq2 < 258.55)&(tce\_minmes >= -2.7)&(tce\_dicco\_msky < 3.38) => (av\_training\_set = PC) (88/88, 6.93%)  
Rule 11. (tce\_nkoi >= 1.5)&(tce\_duration < 2.11)&(tce\_time0bk >= 131.7) => (av\_training\_set = PC) (85/85, 6.69%)  
Rule 12. (tce\_impact < 0)&(tce\_ror >= 0.01) => (av\_training\_set = PC) (3/3, 0.24%)  
Rule 13. (tce\_minmes >= -3.55)&(tce\_chisq2 < 619.65)&(tce\_num\_transits >= 186.5)&(tce\_period >= 5.08) => (av\_training\_set = PC) (83/83, 6.54%)  
Rule 14. (tce\_rminmes < 0.13)&(tce\_fwm\_sdec >= 41.13)&(tce\_chisq2 < 3594) => (av\_training\_set = PC) (411/411, 32.36%)  
Rule 15. (tce\_ldm\_coeff4 in [-0.57, -0.56])&(tce\_incl < 89.81) => (av\_training\_set = PC) (8/8, 0.63%)  
Rule 16. (tce\_chisq2 < 401.25)&(tce\_dof1 >= 3334.5) => (av\_training\_set = PC) (16/16, 1.26%)  
Rule 17. (tce\_chisq1 < 729.15)&(tce\_rminmes < 0.29)&(tce\_fwm\_stat < 12.06)&(tce\_prad < 3.76) => (av\_training\_set = PC) (219/219, 17.24%)  
Rule 18. (tce\_max\_sngle\_ev < 2.84)&(tce\_period >= 4.68) => (av\_training\_set = PC) (5/5, 0.39%)  
Rule 19. (tce\_nkoi >= 1.5)&(tce\_time0bk >= 142.7)&(tce\_period < 60.79) => (av\_training\_set = PC) (102/102, 8.03%)  
Rule 20. (tce\_sradius < 0.47)&(tce\_ldm\_coeff2 < 0.71) => (av\_training\_set = PC) (11/11, 0.87%)  
Rule 21. (tce\_minmes >= -3.55)&(tce\_chisq2 < 619.65)&(tce\_num\_transits >= 351.5) => (av\_training\_set = PC) (18/18, 1.42%)  
Rule 22. (tce\_minmes >= -3.55)&(tce\_chisq2 < 470.65)&(tce\_rmesmad >= 160)&(tce\_ingress < 1.35) => (av\_training\_set = PC) (34/34, 2.68%)  
Rule 23. (tce\_minmes in [-3.55, -3.52])&(tce\_ingress >= 0.03) => (av\_training\_set = PC) (16/16, 1.26%)  
Rule 24. (tce\_minmes >= -3.52)&(tce\_chisq2 < 258.55)&(tce\_model\_dof >= 3972.5)&(tce\_model\_snr >= 9.98) => (av\_training\_set = PC) (85/85, 6.69%)  
Rule 25. (tce\_duration >= 27.75)&(tce\_time0bk < 134.16) => (av\_training\_set = PC) (1/1, 0.08%)  
Rule 26. (tce\_nkoi >= 0.5)&(tce\_fwm\_sdec in [43.03, 43.81])&(tce\_model\_snr >= 7.48) => (av\_training\_set = PC) (122/122, 9.61%)  
Rule 27. (tce\_smet >= 0.47)&(tce\_duration < 3.17) => (av\_training\_set = PC) (10/10, 0.79%)  
Rule 28. (tce\_minmes >= -3.08)&(tce\_chisq1 < 247.15)&(tce\_period < 90.39) => (av\_training\_set = PC) (23/23, 1.81%)  
Rule 29. (tce\_minmes >= -2.91)&(tce\_time0bk >= 134.45)&(tce\_sma < 0.06) => (av\_training\_set = PC) (30/30, 2.36%)  
Rule 30. (tce\_nkoi >= 0.5)&(tce\_chisq1 < 166.35)&(tce\_rsnrmes >= 1.03) => (av\_training\_set = PC) (51/51, 4.02%)  
Rule 31. (tce\_minmes >= -2.42)&(tce\_dor >= 8.3)&(tce\_smet >= -0.2) => (av\_training\_set = PC) (53/53, 4.17%)  
Rule 32. (tce\_nkoi >= 0.5)&(tce\_fwm\_sdec >= 44.1)&(tce\_mesmad < 0.67) => (av\_training\_set = PC) (86/86, 6.77%)  
Rule 33. (tce\_rminmes < 0.21)&(tce\_fwm\_sdec in [44.56, 46.17]) => (av\_training\_set = PC) (167/167, 13.15%)  
Rule 34. (tce\_rminmes < 0.2)&(tce\_chisq2 < 209.85)&(tce\_dikco\_msky < 0.18)&(tce\_impact < 0.95) => (av\_training\_set = PC) (108/108, 8.5%)  
Rule 35. (tce\_nkoi >= 1.5)&(tce\_period >= 94.92)&(tce\_time0bk < 177.29) => (av\_training\_set = PC) (14/14, 1.1%)  
Rule 36. (tce\_duration < 0.72)&(tce\_period >= 0.93) => (av\_training\_set = PC) (3/3, 0.24%)  
Rule 37. (tce\_minmes in [-3.52, -3.5])&(tce\_steff < 5409.5) => (av\_training\_set = PC) (8/8, 0.63%)  
Rule 38. (tce\_minmes in [-3.5, -3.49])&(tce\_impact < 0.47) => (av\_training\_set = PC) (3/3, 0.24%)  
Rule 39. (tce\_minmes in [-3.42, -3.41])&(tce\_ror >= 0) => (av\_training\_set = PC) (9/9, 0.71%)  
Rule 40. (tce\_ingress >= 21.79)&(tce\_duration < 8.75) => (av\_training\_set = PC) (1/1, 0.08%)  
Rule 41. (tce\_minmes in [-2.91, -2.88])&(tce\_model\_snr >= 8.67) => (av\_training\_set = PC) (29/29, 2.28%)  
Rule 42. (tce\_minmes >= -2.76)&(tce\_time0bk >= 135.58)&(tce\_rsnrmes >= 1.02)&(tce\_impact >= 0.17) => (av\_training\_set = PC) (89/89, 7.01%)  
Rule 43. (tce\_nkoi >= 1.5)&(tce\_rminmes < 0.01) => (av\_training\_set = PC) (7/7, 0.55%)  
Rule 44. (tce\_fwm\_sdec >= 40.9)&(tce\_model\_snr >= 445.5) => (av\_training\_set = PC) (29/29, 2.28%)  
Rule 45. (tce\_minmes >= -2.38)&(tce\_chisq1 >= 7311500) => (av\_training\_set = PC) (1/1, 0.08%)  
Rule 46. (tce\_nkoi >= 0.5)&(tce\_fwm\_sdec >= 40.9)&(tce\_ldm\_coeff4 < -0.43)&(tce\_max\_sngle\_ev < 8.48) => (av\_training\_set = PC) (100/100, 7.87%)  
Rule 47. (tce\_minmes >= -2.38)&(tce\_smet >= 0.22) => (av\_training\_set = PC) (11/11, 0.87%)  
Rule 48. (tce\_nkoi >= 0.5)&(tce\_chisq1 < 236.7)&(tce\_sma < 0.26) => (av\_training\_set = PC) (40/40, 3.15%)  
Rule 49. (tce\_minmes in [-2.76, -2.75])&(tce\_period < 3.91) => (av\_training\_set = PC) (2/2, 0.16%)  
Rule 50. (tce\_minmes >= -2.67)&(tce\_ldm\_coeff1 >= 0.72) => (av\_training\_set = PC) (7/7, 0.55%)  
Rule 51. (tce\_nkoi >= 1.5)&(tce\_max\_sngle\_ev < 4.52)&(tce\_max\_mult\_ev >= 9.68) => (av\_training\_set = PC) (88/88, 6.93%)  
Rule 52. (tce\_minmes in [-2.38, -2.36]) => (av\_training\_set = PC) (11/11, 0.87%)  
Rule 53. (tce\_minmes in [-2.36, -2.34]) => (av\_training\_set = PC) (10/10, 0.79%)  
Rule 54. (tce\_minmes in [-2.28, -2.28]) => (av\_training\_set = PC) (2/2, 0.16%)  
Rule 55. (tce\_nkoi >= 0.5)&(tce\_chisq1 < 131.45)&(tce\_model\_snr >= 9.05)&(tce\_dor < 143.86) => (av\_training\_set = PC) (59/59, 4.65%)

Rule 56. (tce\_rminmes < 0.14)&(tce\_rsnrmes >= 1.44) => (av\_training\_set = PC) (2/2, 0.16%)  
 Rule 57. (tce\_rminmes in [0.21, 0.21]) => (av\_training\_set = PC) (1/1, 0.08%)  
 Rule 58. (tce\_rminmes in [0.14, 0.14]) => (av\_training\_set = PC) (7/7, 0.55%)  
 Rule 59. (tce\_rminmes in [0.14, 0.14]) => (av\_training\_set = PC) (12/12, 0.94%)  
 Rule 60. (tce\_rminmes in [0.13, 0.13]) => (av\_training\_set = PC) (11/11, 0.87%)  
 Rule 61. (tce\_slogg >= 4.67)&(tce\_incl >= 90) => (av\_training\_set = PC) (4/4, 0.31%)  
 Rule 62. (tce\_sradius < 0.6)&(tce\_ldm\_coeff2 < -0.37) => (av\_training\_set = PC) (4/4, 0.31%)  
 Rule 63. (tce\_minmes in [-2.17, -2.14]) => (av\_training\_set = PC) (14/14, 1.1%)  
 Rule 64. (tce\_dicco\_mdec >= 9.16)&(tce\_ror >= 0.02) => (av\_training\_set = PC) (2/2, 0.16%)  
 Rule 65. (tce\_nkoi >= 0.5)&(tce\_incl < 20.96)&(tce\_dor >= 1.04) => (av\_training\_set = PC) (1/1, 0.08%)  
 Rule 66. (tce\_minmes in [-3.5, -3.49]) => (av\_training\_set = PC) (3/3, 0.24%)  
 Rule 67. (tce\_minmes in [-3.38, -3.38]) => (av\_training\_set = PC) (3/3, 0.24%)  
 Rule 68. (tce\_minmes in [-3.15, -3.15]) => (av\_training\_set = PC) (5/5, 0.39%)  
 Rule 69. (tce\_minmes in [-3.14, -3.13]) => (av\_training\_set = PC) (6/6, 0.47%)  
 Rule 70. (tce\_minmes in [-3.1, -3.1]) => (av\_training\_set = PC) (6/6, 0.47%)  
 Rule 71. (tce\_rminmes in [0.22, 0.22]) => (av\_training\_set = PC) (3/3, 0.24%)  
 Rule 72. (tce\_rminmes in [0.21, 0.21]) => (av\_training\_set = PC) (6/6, 0.47%)  
 Rule 73. (tce\_rminmes in [0.13, 0.13]) => (av\_training\_set = PC) (2/2, 0.16%)  
 Rule 74. (tce\_minmes >= -2.1)&(tce\_model\_snr >= 23.1)&(tce\_period >= 2.38) => (av\_training\_set = PC) (13/13, 1.02%)  
 Rule 75. (tce\_mesmad in [0.61, 0.61]) => (av\_training\_set = PC) (3/3, 0.24%)  
 Rule 76. (tce\_duration in [1.22, 1.25])&(tce\_time0bk >= 131.59) => (av\_training\_set = PC) (6/6, 0.47%)  
 Rule 77. (tce\_mesmad in [0.51, 0.52])&(tce\_period >= 0.57) => (av\_training\_set = PC) (2/2, 0.16%)  
 Rule 78. (tce\_duration in [0.87, 0.88]) => (av\_training\_set = PC) (1/1, 0.08%)  
 Rule 79. (tce\_rminmes < 0.01)&(tce\_chisq2 < 119.85) => (av\_training\_set = PC) (7/7, 0.55%)  
 Rule 80. (tce\_rminmes in [0.01, 0.01]) => (av\_training\_set = PC) (2/2, 0.16%)  
 Rule 81. (tce\_rminmes in [0.01, 0.01]) => (av\_training\_set = PC) (4/4, 0.31%)  
 Rule 82. (tce\_time0bk in [131.55, 131.56]) => (av\_training\_set = PC) (1/1, 0.08%)  
 Rule 83. (tce\_ldm\_coeff4 in [-0.54, -0.54]) => (av\_training\_set = PC) (3/3, 0.24%)  
 Rule 84. (tce\_time0bk in [131.53, 131.53])&(tce\_period >= 0.54) => (av\_training\_set = PC) (1/1, 0.08%)  
 Rule 85. (tce\_period in [535.36, 536.63]) => (av\_training\_set = PC) (1/1, 0.08%)  
 Rule 86. (tce\_nkoi >= 0.5)&(tce\_time0bk >= 319.58)&(tce\_period < 293.57) => (av\_training\_set = PC) (5/5, 0.39%)  
 Rule 87. (tce\_minmes in [-3.1, -3.09]) => (av\_training\_set = PC) (4/4, 0.31%)  
 Rule 88. (tce\_minmes in [-3.07, -3.03]) => (av\_training\_set = PC) (29/29, 2.28%)  
 Rule 89. (tce\_ldm\_coeff4 in [-0.55, -0.55])&(tce\_period < 21.04) => (av\_training\_set = PC) (1/1, 0.08%)  
 Rule 90. (tce\_minmes in [-3.01, -3]) => (av\_training\_set = PC) (6/6, 0.47%)  
 Rule 91. (tce\_nkoi >= 0.5)&(tce\_fwm\_sdec in [40.9, 41.09])&(tce\_period >= 1.34) => (av\_training\_set = PC) (30/30, 2.36%)  
 Rule 92. (tce\_ldm\_coeff4 in [-0.55, -0.55]) => (av\_training\_set = PC) (2/2, 0.16%)  
 Rule 93. (tce\_minmes in [-2.82, -2.82]) => (av\_training\_set = PC) (7/7, 0.55%)  
 Rule 94. (tce\_minmes in [-2.69, -2.69]) => (av\_training\_set = PC) (2/2, 0.16%)  
 Rule 95. (tce\_duration in [1.22, 1.22]) => (av\_training\_set = PC) (1/1, 0.08%)  
 Rule 96. (tce\_minmes in [-2.63, -2.63]) => (av\_training\_set = PC) (2/2, 0.16%)  
 Rule 97. (tce\_impact in [0.01, 0.01]) => (av\_training\_set = PC) (2/2, 0.16%)  
 Rule 98. (tce\_duration in [1.16, 1.17]) => (av\_training\_set = PC) (2/2, 0.16%)  
 Rule 99. (tce\_rminmes in [0, 0]) => (av\_training\_set = PC) (4/4, 0.31%)  
 Rule 100. (tce\_minmes in [-2.63, -2.62]) => (av\_training\_set = PC) (5/5, 0.39%)  
 Rule 101. (tce\_mesmad in [0.65, 0.65]) => (av\_training\_set = PC) (1/1, 0.08%)  
 Rule 102. (tce\_impact in [0.01, 0.01]) => (av\_training\_set = PC) (1/1, 0.08%)  
 Rule 103. (tce\_duration in [1.13, 1.13]) => (av\_training\_set = PC) (1/1, 0.08%)  
 Rule 104. (tce\_minmes in [-2.53, -2.52]) => (av\_training\_set = PC) (4/4, 0.31%)  
 Rule 105. (tce\_duration < 1.04)&(tce\_ldm\_coeff1 < 0.28) => (av\_training\_set = PC) (2/2, 0.16%)  
 Rule 106. (tce\_smet >= 0.13)&(tce\_impact in [0, 0.01]) => (av\_training\_set = PC) (3/3, 0.24%)  
 Rule 107. (tce\_mesmad in [0.64, 0.64])&(tce\_period < 6.62) => (av\_training\_set = PC) (1/1, 0.08%)  
 Rule 108. (tce\_minmes in [-2.24, -2.23]) => (av\_training\_set = PC) (1/1, 0.08%)  
 Rule 109. (tce\_smet >= 0.15)&(tce\_model\_snr >= 53.08)&(tce\_period < 1.54) => (av\_training\_set = PC) (2/2, 0.16%)  
 Rule 110. (tce\_chisq2 in [334.9, 343.4]) => (av\_training\_set = PC) (12/12, 0.94%)  
 Rule 111. (tce\_chisq2 in [271.7, 274]) => (av\_training\_set = PC) (3/3, 0.24%)  
 Rule 112. (tce\_chisq1 in [553.9, 555.2]) => (av\_training\_set = PC) (2/2, 0.16%)  
 Rule 113. (tce\_mesmad in [0.64, 0.64]) => (av\_training\_set = PC) (1/1, 0.08%)  
 Rule 114. (tce\_bin\_oedp\_stat in [8.22, 8.33]) => (av\_training\_set = PC) (2/2, 0.16%)  
 Rule 115. (tce\_mesmad in [0.63, 0.63]) => (av\_training\_set = PC) (8/8, 0.63%)  
 Rule 116. (tce\_bin\_oedp\_stat in [3.77, 3.82]) => (av\_training\_set = PC) (5/5, 0.39%)  
 Rule 117. (tce\_mesmad in [0.72, 0.72]) => (av\_training\_set = PC) (4/4, 0.31%)  
 Rule 118. (tce\_duration in [1.37, 1.37])&(tce\_period >= 0.99) => (av\_training\_set = PC) (1/1, 0.08%)  
 Rule 119. (tce\_model\_snr in [37.79, 38.01]) => (av\_training\_set = PC) (5/5, 0.39%)  
 Rule 120. (tce\_dikco\_mra in [1.4, 1.41]) => (av\_training\_set = PC) (1/1, 0.08%)  
 Rule 121. (tce\_depth >= 298650)&(tce\_dor < 859.01) => (av\_training\_set = AFP) (27/27, 2.13%)

Rule 122. (tce\_dikco\_mdec < -5.82)&(tce\_fwm\_stat >= 26.73)&(tce\_incl < 89.87) => (av\_training\_set = AFP) (44/44, 3.46%)  
 Rule 123. (tce\_ror >= 0.3)&(tce\_dor < 72.73) => (av\_training\_set = AFP) (71/71, 5.59%)  
 Rule 124. (tce\_mesmad >= 5.32)&(tce\_fwm\_sdec < 40.7)&(tce\_minmes >= -9.47) => (av\_training\_set = AFP) (33/33, 2.6%)  
 Rule 125. (tce\_ldm\_coeff1 in [-0.12, -0.02]) => (av\_training\_set = AFP) (11/11, 0.87%)  
 Rule 126. (tce\_steff >= 11021)&(tce\_fwm\_prao < -0) => (av\_training\_set = AFP) (8/8, 0.63%)  
 Rule 127. (tce\_eqt >= 4685)&(tce\_robstat >= 13.68)&(tce\_max\_mult\_ev < 14.32) => (av\_training\_set = AFP) (19/19, 1.5%)  
 Rule 128. (tce\_depth in [1173500, 2263500]) => (av\_training\_set = AFP) (5/5, 0.39%)  
 Rule 129. (tce\_dikco\_mdec >= 10.99)&(tce\_period < 1.6) => (av\_training\_set = AFP) (5/5, 0.39%)  
 Rule 130. (tce\_ingress >= 5.4)&(tce\_fwm\_stat >= 35.06)&(tce\_impact >= 0.86) => (av\_training\_set = AFP) (38/38, 2.99%)  
 Rule 131. (tce\_dof1 >= 32510)&(tce\_ldm\_coeff1 < 0.42)&(tce\_max\_mult\_ev >= 7.89) => (av\_training\_set = AFP) (15/15, 1.18%)  
 Rule 132. (tce\_mesmad >= 6.36)&(tce\_impact >= 0.97) => (av\_training\_set = AFP) (12/12, 0.94%)  
 Rule 133. (tce\_incl < 13.38)&(tce\_ingress >= 0.57) => (av\_training\_set = AFP) (8/8, 0.63%)  
 Rule 134. (tce\_mesmad >= 5.23)&(tce\_fwm\_sdec < 40.7)&(tce\_fwm\_pdeco >= 0)&(tce\_robstat < 36) => (av\_training\_set = AFP) (55/55, 4.33%)  
 Rule 135. (tce\_eqt >= 4735)&(tce\_smet < -0.08)&(tce\_mesmad >= 2.42) => (av\_training\_set = AFP) (24/24, 1.89%)  
 Rule 136. (tce\_fwm\_sdec < 39.78)&(tce\_fwm\_sra >= 19.76)&(tce\_max\_mult\_ev >= 7.54) => (av\_training\_set = AFP) (37/37, 2.91%)  
 Rule 137. (tce\_fwm\_sdec < 39.71)&(tce\_dikco\_mra >= 3.08)&(tce\_max\_sngle\_ev < 6.09) => (av\_training\_set = AFP) (31/31, 2.44%)  
 Rule 138. (tce\_dikco\_mdec >= 5.74)&(tce\_fwm\_sdeco >= 5.62) => (av\_training\_set = AFP) (33/33, 2.6%)  
 Rule 139. (tce\_time0bk < 131.51) => (av\_training\_set = AFP) (1/1, 0.08%)  
 Rule 140. (tce\_fwm\_sdec in [39.68, 39.71])&(tce\_chisq2 >= 1620) => (av\_training\_set = AFP) (13/13, 1.02%)  
 Rule 141. (tce\_fwm\_sdec < 39.63)&(tce\_max\_sngle\_ev < 4.25)&(tce\_dof1 >= 20390) => (av\_training\_set = AFP) (29/29, 2.28%)  
 Rule 142. (tce\_fwm\_sdec < 39.63)&(tce\_model\_snr >= 300.5)&(tce\_dicco\_mdec < 0.07) => (av\_training\_set = AFP) (56/56, 4.41%)  
 Rule 143. (tce\_fwm\_sdec < 39.63)&(tce\_incl < 20.31)&(tce\_rsnrmes < 1.03) => (av\_training\_set = AFP) (13/13, 1.02%)  
 Rule 144. (tce\_fwm\_sdec < 39.63)&(tce\_fwm\_stat >= 43.22)&(tce\_time0bk < 131.63) => (av\_training\_set = AFP) (26/26, 2.05%)  
 Rule 145. (tce\_fwm\_sdec < 39.63)&(tce\_dikco\_mra < -3.21)&(tce\_model\_snr >= 9.35) => (av\_training\_set = AFP) (48/48, 3.78%)  
 Rule 146. (tce\_fwm\_sdec in [40.89, 40.9]) => (av\_training\_set = AFP) (11/11, 0.87%)  
 Rule 147. (tce\_fwm\_sdec in [40.87, 40.89])&(tce\_period < 6.56) => (av\_training\_set = AFP) (11/11, 0.87%)  
 Rule 148. (tce\_rsnrmes < -0)&(tce\_robstat >= 240) => (av\_training\_set = AFP) (16/16, 1.26%)  
 Rule 149. (tce\_fwm\_sdec < 40.87)&(tce\_fwm\_stat >= 173.08)&(tce\_chisq1 < 10915)&(tce\_nkoi < 2.5) => (av\_training\_set = AFP) (141/141, 11.1%)  
 Rule 150. (tce\_fwm\_sdec in [40.82, 40.87])&(tce\_minmes < -8.35) => (av\_training\_set = AFP) (17/17, 1.34%)  
 Rule 151. (tce\_fwm\_sdec in [39.6, 39.63])&(tce\_eqt >= 1405) => (av\_training\_set = AFP) (8/8, 0.63%)  
 Rule 152. (tce\_sma < 0.01)&(tce\_period < 0.53) => (av\_training\_set = AFP) (3/3, 0.24%)  
 Rule 153. (tce\_fwm\_sdec in [39.58, 39.59])&(tce\_period >= 0.68) => (av\_training\_set = AFP) (11/11, 0.87%)  
 Rule 154. (tce\_fwm\_sdec in [39.57, 39.57]) => (av\_training\_set = AFP) (4/4, 0.31%)  
 Rule 155. (tce\_fwm\_sdec in [39.4, 39.5])&(tce\_eqt >= 2335) => (av\_training\_set = AFP) (30/30, 2.36%)  
 Rule 156. (tce\_fwm\_sdec < 39.05)&(tce\_fwm\_sra >= 19.73) => (av\_training\_set = AFP) (6/6, 0.47%)  
 Rule 157. (tce\_fwm\_sdec in [38.27, 39.05])&(tce\_ldm\_coeff2 >= 1.02)&(tce\_ldm\_coeff1 >= 0.21) => (av\_training\_set = AFP) (41/41, 3.23%)  
 Rule 158. (tce\_rsnrmes < -0)&(tce\_robstat >= 29.33)&(tce\_time0bk >= 131.62) => (av\_training\_set = AFP) (9/9, 0.71%)  
 Rule 159. (tce\_fwm\_stat >= 13344)&(tce\_fwm\_sra < 19.65) => (av\_training\_set = AFP) (53/53, 4.17%)  
 Rule 160. (tce\_duration >= 16.75)&(tce\_fwm\_sra >= 19.82)&(tce\_max\_mult\_ev >= 10) => (av\_training\_set = AFP) (38/38, 2.99%)  
 Rule 161. (tce\_max\_sngle\_ev < 3.39)&(tce\_robstat >= 17.44)&(tce\_ror >= 0) => (av\_training\_set = AFP) (11/11, 0.87%)  
 Rule 162. (tce\_dikco\_mdec < -5.82)&(tce\_depth < 48.35) => (av\_training\_set = AFP) (14/14, 1.1%)  
 Rule 163. (tce\_fwm\_sdec in [40.83, 40.85])&(tce\_ldm\_coeff1 < 0.44) => (av\_training\_set = AFP) (20/20, 1.57%)  
 Rule 164. (tce\_rsnrmes < -0.02)&(tce\_period < 0.53) => (av\_training\_set = AFP) (6/6, 0.47%)  
 Rule 165. (tce\_fwm\_sdec < 40.82)&(tce\_fwm\_sra >= 19.98)&(tce\_slogg >= 3.66) => (av\_training\_set = AFP) (13/13, 1.02%)  
 Rule 166. (tce\_fwm\_sdec in [40.15, 40.15]) => (av\_training\_set = AFP) (5/5, 0.39%)  
 Rule 167. (tce\_fwm\_sdec in [39.63, 40.13])&(tce\_fwm\_stat >= 52.19)&(tce\_period >= 0.56) => (av\_training\_set = AFP) (59/59, 4.65%)  
 Rule 168. (tce\_fwm\_sdec in [38.96, 39.05])&(tce\_duration >= 6.44) => (av\_training\_set = AFP) (19/19, 1.5%)  
 Rule 169. (tce\_impact < 0)&(tce\_fwm\_prao >= -0) => (av\_training\_set = AFP) (4/4, 0.31%)  
 Rule 170. (tce\_fwm\_sdec in [38.96, 39])&(tce\_duration < 5.66) => (av\_training\_set = AFP) (27/27, 2.13%)  
 Rule 171. (tce\_fwm\_sdec < 38.94)&(tce\_fwm\_sra >= 19.7)&(tce\_dor < 5.72) => (av\_training\_set = AFP) (16/16, 1.26%)  
 Rule 172. (tce\_fwm\_sdec in [38.92, 38.94])&(tce\_period < 2.81) => (av\_training\_set = AFP) (12/12, 0.94%)  
 Rule 173. (tce\_fwm\_sdec < 38.62)&(tce\_fwm\_srao >= 3.14)&(tce\_ror >= 0) => (av\_training\_set = AFP) (42/42, 3.31%)  
 Rule 174. (tce\_rsnrmes < -0.02)&(tce\_chisq2 >= 3829.5)&(tce\_eqt >= 2415) => (av\_training\_set = AFP) (12/12, 0.94%)  
 Rule 175. (tce\_prad < 0)&(tce\_period >= 0.57) => (av\_training\_set = AFP) (1/1, 0.08%)  
 Rule 176. (tce\_max\_sngle\_ev < 2.76)&(tce\_period < 4.68) => (av\_training\_set = AFP) (2/2, 0.16%)

Rule 177. (tce\_mesmad < 0.42)&(tce\_period >= 0.57) => (av\_training\_set = AFP) (13/13, 1.02%)  
 Rule 178. (tce\_fwm\_sdec in [40.82, 40.82]) => (av\_training\_set = AFP) (7/7, 0.55%)  
 Rule 179. (tce\_fwm\_sdec in [40.81, 40.81]) => (av\_training\_set = AFP) (2/2, 0.16%)  
 Rule 180. (tce\_fwm\_sdec in [40.8, 40.81]) => (av\_training\_set = AFP) (6/6, 0.47%)  
 Rule 181. (tce\_fwm\_sdec in [40.69, 40.78])&(tce\_incl < 77.94) => (av\_training\_set = AFP) (24/24, 1.89%)  
 Rule 182. (tce\_fwm\_sdec in [40.08, 40.1]) => (av\_training\_set = AFP) (7/7, 0.55%)  
 Rule 183. (tce\_fwm\_sdec in [39.97, 39.98])&(tce\_period < 1.38) => (av\_training\_set = AFP) (6/6, 0.47%)  
 Rule 184. (tce\_fwm\_sdec < 38.91)&(tce\_duration < 1.31)&(tce\_dicco\_msky < 0.28) => (av\_training\_set = AFP) (16/16, 1.26%)  
 Rule 185. (tce\_ingress >= 7.81)&(tce\_rminmes < 0.23) => (av\_training\_set = AFP) (25/25, 1.97%)  
 Rule 186. (tce\_fwm\_sdec in [38.71, 38.71]) => (av\_training\_set = AFP) (3/3, 0.24%)  
 Rule 187. (tce\_fwm\_sdec in [38.68, 38.7])&(tce\_time0bk >= 131.8) => (av\_training\_set = AFP) (8/8, 0.63%)  
 Rule 188. (tce\_fwm\_sdec < 39.95)&(tce\_mesmad >= 4.57)&(tce\_fwm\_sdeco >= 0)&(tce\_ror >= 0.01) => (av\_training\_set = AFP) (52/52, 4.09%)  
 Rule 189. (tce\_rsnrmes < -0.02)&(tce\_ldm\_coeff4 < -0.53) => (av\_training\_set = AFP) (2/2, 0.16%)  
 Rule 190. (tce\_ingress >= 14.36)&(tce\_period < 12.07) => (av\_training\_set = AFP) (1/1, 0.08%)  
 Rule 191. (tce\_fwm\_prao >= 0.02)&(tce\_ror >= 0.24) => (av\_training\_set = AFP) (21/21, 1.65%)  
 Rule 192. (tce\_fwm\_sdec in [40.76, 40.78]) => (av\_training\_set = AFP) (7/7, 0.55%)  
 Rule 193. (tce\_fwm\_sdec in [40.76, 40.76]) => (av\_training\_set = AFP) (3/3, 0.24%)  
 Rule 194. (tce\_fwm\_sdec in [40.61, 40.75])&(tce\_fwm\_srao < -1.45) => (av\_training\_set = AFP) (22/22, 1.73%)  
 Rule 195. (tce\_fwm\_sdec in [39.92, 39.95])&(tce\_fwm\_srao >= 0.01) => (av\_training\_set = AFP) (11/11, 0.87%)  
 Rule 196. (tce\_fwm\_sdec in [39.85, 39.91])&(tce\_smet < -0.37) => (av\_training\_set = AFP) (7/7, 0.55%)  
 Rule 197. (tce\_fwm\_sdec in [38.59, 38.62])&(tce\_ldm\_coeff1 >= 0.24) => (av\_training\_set = AFP) (10/10, 0.79%)  
 Rule 198. (tce\_period in [0.51, 0.51]) => (av\_training\_set = AFP) (8/8, 0.63%)  
 Rule 199. (tce\_fwm\_sdec < 38.55)&(tce\_fwm\_sdeco >= 0.99)&(tce\_fwm\_sra < 19.07) => (av\_training\_set = AFP) (14/14, 1.1%)  
 Rule 200. (tce\_rsnrmes < -0.02)&(tce\_robstat >= 27.93) => (av\_training\_set = AFP) (8/8, 0.63%)  
 Rule 201. (tce\_fwm\_sdec in [39.56, 39.56])&(tce\_ror >= 0) => (av\_training\_set = AFP) (3/3, 0.24%)  
 Rule 202. (tce\_fwm\_sdec in [39.53, 39.55])&(tce\_period < 1.49) => (av\_training\_set = AFP) (10/10, 0.79%)  
 Rule 203. (tce\_fwm\_sdec in [39.46, 39.47]) => (av\_training\_set = AFP) (6/6, 0.47%)  
 Rule 204. (tce\_rsnrmes < -0.06)&(tce\_max\_mult\_ev >= 14) => (av\_training\_set = AFP) (5/5, 0.39%)  
 Rule 205. (tce\_fwm\_sdec in [39.43, 39.46])&(tce\_time0bk < 135.03) => (av\_training\_set = AFP) (13/13, 1.02%)  
 Rule 206. (tce\_fwm\_sdec in [39.39, 39.39]) => (av\_training\_set = AFP) (3/3, 0.24%)  
 Rule 207. (tce\_fwm\_sdec in [38.5, 38.55])&(tce\_ldm\_coeff3 < -0.72) => (av\_training\_set = AFP) (11/11, 0.87%)  
 Rule 208. (tce\_fwm\_sdec in [38.44, 38.47])&(tce\_incl >= 82.81) => (av\_training\_set = AFP) (6/6, 0.47%)  
 Rule 209. (tce\_fwm\_sdec in [38.44, 38.45]) => (av\_training\_set = AFP) (8/8, 0.63%)  
 Rule 210. (tce\_fwm\_sdec in [38.41, 38.43])&(tce\_ldm\_coeff4 >= -0.15) => (av\_training\_set = AFP) (10/10, 0.79%)  
 Rule 211. (tce\_fwm\_sdec in [37.97, 37.99]) => (av\_training\_set = AFP) (12/12, 0.94%)  
 Rule 212. (tce\_fwm\_sdec < 40.75)&(tce\_rsnrmes >= 1.75)&(tce\_period >= 0.51) => (av\_training\_set = AFP) (5/5, 0.39%)  
 Rule 213. (tce\_rsnrmes < -0.07)&(tce\_max\_mult\_ev >= 12.36) => (av\_training\_set = AFP) (4/4, 0.31%)  
 Rule 214. (tce\_rsnrmes < -0.09)&(tce\_max\_mult\_ev >= 10.34) => (av\_training\_set = AFP) (7/7, 0.55%)  
 Rule 215. (tce\_fwm\_sdec in [40.75, 40.75]) => (av\_training\_set = AFP) (2/2, 0.16%)  
 Rule 216. (tce\_fwm\_sdec in [40.74, 40.75]) => (av\_training\_set = AFP) (4/4, 0.31%)  
 Rule 217. (tce\_fwm\_sdec in [40.73, 40.74])&(tce\_period < 1.03) => (av\_training\_set = AFP) (5/5, 0.39%)  
 Rule 218. (tce\_fwm\_sdec in [40.69, 40.7])&(tce\_time0bk < 133.52) => (av\_training\_set = AFP) (14/14, 1.1%)  
 Rule 219. (tce\_fwm\_sdec in [40.65, 40.66])&(tce\_period < 4.34) => (av\_training\_set = AFP) (7/7, 0.55%)  
 Rule 220. (tce\_fwm\_sdec in [39.28, 39.38])&(tce\_robstat < 18.74)&(tce\_ror >= 0) => (av\_training\_set = AFP) (20/20, 1.57%)  
 Rule 221. (tce\_fwm\_sdec < 38.39)&(tce\_fwm\_srao < -1.95)&(tce\_ldm\_coeff1 >= 0.44) => (av\_training\_set = AFP) (17/17, 1.34%)  
 Rule 222. (tce\_steff >= 9306)&(tce\_ldm\_coeff2 >= 0.06) => (av\_training\_set = AFP) (5/5, 0.39%)  
 Rule 223. (tce\_fwm\_sdec in [40.65, 40.65]) => (av\_training\_set = AFP) (6/6, 0.47%)  
 Rule 224. (tce\_rsnrmes < -0.1)&(tce\_max\_mult\_ev >= 9.64) => (av\_training\_set = AFP) (4/4, 0.31%)  
 Rule 225. (tce\_fwm\_sdec in [40.64, 40.65]) => (av\_training\_set = AFP) (8/8, 0.63%)  
 Rule 226. (tce\_fwm\_sdec in [40.07, 40.07])&(tce\_period >= 1.75) => (av\_training\_set = AFP) (5/5, 0.39%)  
 Rule 227. (tce\_fwm\_sdec in [39.24, 39.24]) => (av\_training\_set = AFP) (3/3, 0.24%)  
 Rule 228. (tce\_fwm\_sdec in [39.19, 39.2]) => (av\_training\_set = AFP) (1/1, 0.08%)  
 Rule 229. (tce\_fwm\_sdec in [39.19, 39.19])&(tce\_period >= 0.65) => (av\_training\_set = AFP) (2/2, 0.16%)  
 Rule 230. (tce\_fwm\_pdeco < -0)&(tce\_impact >= 0.81) => (av\_training\_set = AFP) (38/38, 2.99%)  
 Rule 231. (tce\_fwm\_sdec in [38.39, 38.39]) => (av\_training\_set = AFP) (3/3, 0.24%)  
 Rule 232. (tce\_fwm\_sdec in [38.31, 38.34])&(tce\_period < 3.31) => (av\_training\_set = AFP) (14/14, 1.1%)  
 Rule 233. (tce\_rsnrmes < -0.11)&(tce\_max\_mult\_ev >= 9) => (av\_training\_set = AFP) (2/2, 0.16%)  
 Rule 234. (tce\_fwm\_sdec in [40.62, 40.64])&(tce\_model\_chisq < 30065) => (av\_training\_set = AFP) (15/15, 1.18%)  
 Rule 235. (tce\_fwm\_sdec in [40.06, 40.06]) => (av\_training\_set = AFP) (5/5, 0.39%)  
 Rule 236. (tce\_fwm\_sdec in [40.04, 40.05]) => (av\_training\_set = AFP) (1/1, 0.08%)  
 Rule 237. (tce\_fwm\_sdec in [39.13, 39.16])&(tce\_period < 3.22) => (av\_training\_set = AFP) (17/17, 1.34%)  
 Rule 238. (tce\_fwm\_sdec < 37.97)&(tce\_fwm\_sra >= 19.53)&(tce\_eqt >= 1575) => (av\_training\_set = AFP) (8/8, 0.63%)  
 Rule 239. (tce\_fwm\_sdec in [38.87, 38.91])&(tce\_dor < 1.56) => (av\_training\_set = AFP) (4/4, 0.31%)

Rule 240. (tce\_fwm\_sdec in [38.28, 38.31])&(tce\_dor >= 2.1) => (av\_training\_set = AFP) (6/6, 0.47%)  
Rule 241. (tce\_fwm\_sdeco >= 3.82)&(tce\_dor < 1.09) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 242. (tce\_dikco\_mdec >= 2.71)&(tce\_duration < 1.29)&(tce\_period >= 0.51) => (av\_training\_set = AFP) (12/12, 0.94%)  
Rule 243. (tce\_time0bk < 131.53)&(tce\_sma < 0.02)&(tce\_rmesmad >= 1.73) => (av\_training\_set = AFP) (9/9, 0.71%)  
Rule 244. (tce\_fwm\_sdec < 37.97)&(tce\_fwm\_pdeco < -0)&(tce\_period < 8.39) => (av\_training\_set = AFP) (14/14, 1.1%)  
Rule 245. (tce\_fwm\_sdec in [40.01, 40.04])&(tce\_period < 8.67) => (av\_training\_set = AFP) (9/9, 0.71%)  
Rule 246. (tce\_impact in [0, 0]) => (av\_training\_set = AFP) (1/1, 0.08%)  
Rule 247. (tce\_fwm\_sdec in [39.91, 39.91]) => (av\_training\_set = AFP) (1/1, 0.08%)  
Rule 248. (tce\_fwm\_sdec in [37.96, 37.97]) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 249. (tce\_fwm\_sdec in [37.91, 37.93])&(tce\_period < 1.94) => (av\_training\_set = AFP) (11/11, 0.87%)  
Rule 250. (tce\_ldm\_coeff1 < 0.17)&(tce\_max\_sngle\_ev < 3.72)&(tce\_period >= 1.75) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 251. (tce\_fwm\_sdec in [37.89, 37.9]) => (av\_training\_set = AFP) (7/7, 0.55%)  
Rule 252. (tce\_fwm\_sdec in [39.91, 39.91]) => (av\_training\_set = AFP) (1/1, 0.08%)  
Rule 253. (tce\_rsnrmes < -0.12)&(tce\_ingress < 0.03)&(tce\_period >= 0.57) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 254. (tce\_depth < 0.01)&(tce\_ror >= 0) => (av\_training\_set = AFP) (1/1, 0.08%)  
Rule 255. (tce\_fwm\_sdec in [39.89, 39.9]) => (av\_training\_set = AFP) (6/6, 0.47%)  
Rule 256. (tce\_fwm\_sdec in [39.85, 39.86]) => (av\_training\_set = AFP) (1/1, 0.08%)  
Rule 257. (tce\_fwm\_sdec in [39.84, 39.85])&(tce\_period >= 0.74) => (av\_training\_set = AFP) (4/4, 0.31%)  
Rule 258. (tce\_steff >= 9985.5)&(tce\_ldm\_coeff1 < 0.6) => (av\_training\_set = AFP) (4/4, 0.31%)  
Rule 259. (tce\_fwm\_sdec in [39.83, 39.83]) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 260. (tce\_fwm\_sdec in [39.82, 39.82]) => (av\_training\_set = AFP) (8/8, 0.63%)  
Rule 261. (tce\_fwm\_sdec in [38.9, 38.9]) => (av\_training\_set = AFP) (5/5, 0.39%)  
Rule 262. (tce\_ldm\_coeff4 < -0.57)&(tce\_duration >= 10.43) => (av\_training\_set = AFP) (1/1, 0.08%)  
Rule 263. (tce\_num\_transits < 1)&(tce\_duration < 2) => (av\_training\_set = AFP) (7/7, 0.55%)  
Rule 264. (tce\_fwm\_sdec in [38.85, 38.86]) => (av\_training\_set = AFP) (4/4, 0.31%)  
Rule 265. (tce\_max\_mult\_ev in [7.14, 7.14])&(tce\_period < 0.67) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 266. (tce\_fwm\_sdec in [38.78, 38.79]) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 267. (tce\_period in [0.52, 0.52]) => (av\_training\_set = AFP) (3/3, 0.24%)  
Rule 268. (tce\_chisq2 >= 174050)&(tce\_time0bk < 131.71) => (av\_training\_set = AFP) (7/7, 0.55%)  
Rule 269. (tce\_fwm\_sdec in [38.74, 38.75]) => (av\_training\_set = AFP) (6/6, 0.47%)  
Rule 270. (tce\_fwm\_sdec in [37.85, 37.89])&(tce\_smet < -0.2) => (av\_training\_set = AFP) (6/6, 0.47%)  
Rule 271. (tce\_dicco\_mdec < -4.62)&(tce\_minmes >= -2.31) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 272. (tce\_fwm\_sdec in [37.64, 37.68]) => (av\_training\_set = AFP) (11/11, 0.87%)  
Rule 273. (tce\_fwm\_sdec in [40.61, 40.62]) => (av\_training\_set = AFP) (4/4, 0.31%)  
Rule 274. (tce\_fwm\_sdec in [40.61, 40.61]) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 275. (tce\_fwm\_sdec in [40.59, 40.6]) => (av\_training\_set = AFP) (7/7, 0.55%)  
Rule 276. (tce\_fwm\_sdec in [39.8, 39.81]) => (av\_training\_set = AFP) (4/4, 0.31%)  
Rule 277. (tce\_ingress in [5.4, 6.3]) => (av\_training\_set = AFP) (17/17, 1.34%)  
Rule 278. (tce\_ldm\_coeff4 >= 0.45)&(tce\_ldm\_coeff3 >= -1.3) => (av\_training\_set = AFP) (5/5, 0.39%)  
Rule 279. (tce\_fwm\_sdec in [37.42, 37.52])&(tce\_period < 2.57) => (av\_training\_set = AFP) (11/11, 0.87%)  
Rule 280. (tce\_steff >= 8382)&(tce\_ldm\_coeff4 < -0.08)&(tce\_period < 0.68) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 281. (tce\_fwm\_sdec in [39.78, 39.78]) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 282. (tce\_rsnrmes >= 1.63)&(tce\_smet >= 0.19) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 283. (tce\_fwm\_sdeco >= 7.6)&(tce\_duration >= 14.96) => (av\_training\_set = AFP) (1/1, 0.08%)  
Rule 284. (tce\_incl in [19.55, 20.31]) => (av\_training\_set = AFP) (3/3, 0.24%)  
Rule 285. (tce\_bin\_oedp\_stat >= 21.42)&(tce\_minmes >= -3.52) => (av\_training\_set = AFP) (13/13, 1.02%)  
Rule 286. (tce\_depth in [3.61, 3.95])&(tce\_period >= 0.54) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 287. (tce\_rsnrmes < -0.11)&(tce\_max\_mult\_ev >= 8.96)&(tce\_period >= 0.57) => (av\_training\_set = AFP) (1/1, 0.08%)  
Rule 288. (tce\_fwm\_sdec in [40.57, 40.59])&(tce\_incl >= 77.44) => (av\_training\_set = AFP) (8/8, 0.63%)  
Rule 289. (tce\_fwm\_sdec in [39.77, 39.77]) => (av\_training\_set = AFP) (1/1, 0.08%)  
Rule 290. (tce\_fwm\_sdec in [39.76, 39.76]) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 291. (tce\_fwm\_sdec in [39.72, 39.74]) => (av\_training\_set = AFP) (3/3, 0.24%)  
Rule 292. (tce\_fwm\_sdec in [38.26, 38.27]) => (av\_training\_set = AFP) (4/4, 0.31%)  
Rule 293. (tce\_depth < 2.52)&(tce\_ror >= 0) => (av\_training\_set = AFP) (1/1, 0.08%)  
Rule 294. (tce\_ldm\_coeff1 in [0.16, 0.16]) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 295. (tce\_eqt in [4275, 4285]) => (av\_training\_set = AFP) (3/3, 0.24%)  
Rule 296. (tce\_fwm\_sdec in [40.56, 40.56])&(tce\_period < 4.12) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 297. (tce\_fwm\_sdec in [39.36, 39.38])&(tce\_impact >= 0.29) => (av\_training\_set = AFP) (8/8, 0.63%)  
Rule 298. (tce\_time0bk < 131.55)&(tce\_dor < 1.01) => (av\_training\_set = AFP) (1/1, 0.08%)  
Rule 299. (tce\_chisq2 in [19025, 19280]) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 300. (tce\_fwm\_sdec in [38.71, 38.73]) => (av\_training\_set = AFP) (6/6, 0.47%)  
Rule 301. (tce\_fwm\_sdec in [38.2, 38.21]) => (av\_training\_set = AFP) (6/6, 0.47%)  
Rule 302. (tce\_eqt >= 4360)&(tce\_fwm\_stat >= 107.69) => (av\_training\_set = AFP) (6/6, 0.47%)  
Rule 303. (tce\_eqt in [4595, 4605]) => (av\_training\_set = AFP) (3/3, 0.24%)  
Rule 304. (tce\_fwm\_sdeco in [5.48, 5.6]) => (av\_training\_set = AFP) (2/2, 0.16%)  
Rule 305. (tce\_fwm\_sdec in [39.67, 39.68]) => (av\_training\_set = AFP) (2/2, 0.16%)

Rule 306. (tce\_fwm\_sdec in [39.6, 39.6]) => (av\_training\_set = AFP) (1/1, 0.08%)  
 Rule 307. (tce\_fwm\_sdec in [39.29, 39.31]) => (av\_training\_set = AFP) (5/5, 0.39%)  
 Rule 308. (tce\_fwm\_sdec in [37.77, 37.78]) => (av\_training\_set = AFP) (6/6, 0.47%)  
 Rule 309. (tce\_fwm\_pdeco in [-0, -0]) => (av\_training\_set = AFP) (1/1, 0.08%)  
 Rule 310. (tce\_period in [0.55, 0.55]) => (av\_training\_set = AFP) (4/4, 0.31%)  
 Rule 311. (tce\_fwm\_sdec in [37.62, 37.63]) => (av\_training\_set = AFP) (2/2, 0.16%)  
 Rule 312. (tce\_fwm\_sdec < 37.32)&(tce\_fwm\_sra >= 19.45)&(tce\_period < 3.99) => (av\_training\_set = AFP) (14/14, 1.1%)  
 Rule 313. (tce\_mesmad >= 4.35)&(tce\_minmes >= -6.18) => (av\_training\_set = AFP) (5/5, 0.39%)  
 Rule 314. (tce\_mesmad >= 5.05)&(tce\_duration >= 15) => (av\_training\_set = AFP) (3/3, 0.24%)  
 Rule 315. (tce\_ldm\_coeff1 in [0.2, 0.2]) => (av\_training\_set = AFP) (5/5, 0.39%)  
 Rule 316. (tce\_mesmad in [5.32, 5.41]) => (av\_training\_set = AFP) (7/7, 0.55%)  
 Rule 317. (tce\_bin\_oedp\_stat in [7.25, 7.31]) => (av\_training\_set = AFP) (1/1, 0.08%)  
 Rule 318. (tce\_fwm\_srao in [-3.79, -3.74]) => (av\_training\_set = AFP) (2/2, 0.16%)  
 Rule 319. (tce\_minmes in [-16.74, -16.13]) => (av\_training\_set = AFP) (8/8, 0.63%)  
 Rule 320. (tce\_max\_sngle\_ev in [3.74, 3.74]) => (av\_training\_set = AFP) (2/2, 0.16%)  
 Rule 321. (tce\_fwm\_pdeco in [-0, -0]) => (av\_training\_set = AFP) (3/3, 0.24%)  
 Rule 322. (tce\_robstat in [8.35, 8.36]) => (av\_training\_set = AFP) (1/1, 0.08%)  
 Rule 323. (tce\_incl in [30.52, 30.6]) => (av\_training\_set = AFP) (1/1, 0.08%)  
 Rule 324. (tce\_max\_sngle\_ev in [3.67, 3.68])&(tce\_period < 1.98) => (av\_training\_set = AFP) (1/1, 0.08%)  
 Rule 325. (tce\_ingress in [1.7, 1.72]) => (av\_training\_set = AFP) (5/5, 0.39%)  
 Rule 326. (tce\_duration in [14.13, 14.14]) => (av\_training\_set = AFP) (1/1, 0.08%)  
 Rule 327. (tce\_chisq2 in [5817.5, 5875.5]) => (av\_training\_set = AFP) (3/3, 0.24%)  
 Rule 328. (tce\_duration in [14.63, 14.64])&(tce\_ror >= 0.01) => (av\_training\_set = AFP) (1/1, 0.08%)  
 Rule 329. (tce\_fwm\_sdec < 37.21)&(tce\_model\_snr >= 24.65) => (av\_training\_set = AFP) (8/8, 0.63%)  
 Rule 330. (tce\_max\_sngle\_ev in [20.52, 20.59]) => (av\_training\_set = AFP) (2/2, 0.16%)  
 Rule 331. (tce\_time0bk in [131.83, 131.83]) => (av\_training\_set = AFP) (4/4, 0.31%)  
 Rule 332. (tce\_impact in [0.02, 0.03]) => (av\_training\_set = AFP) (2/2, 0.16%)  
 Rule 333. (tce\_max\_sngle\_ev in [22.6, 22.7]) => (av\_training\_set = AFP) (2/2, 0.16%)  
 Rule 334. (tce\_max\_sngle\_ev in [23.81, 23.93]) => (av\_training\_set = AFP) (2/2, 0.16%)  
 Rule 335. (tce\_duration in [1.55, 1.56]) => (av\_training\_set = AFP) (6/6, 0.47%)  
 Rule 336. (tce\_max\_sngle\_ev in [34.01, 34.17]) => (av\_training\_set = AFP) (2/2, 0.16%)  
 Rule 337. (tce\_eqt < 163.5) => (av\_training\_set = NTP) (23/23, 1.81%)  
 Rule 338. (tce\_max\_mult\_ev < 7.13) => (av\_training\_set = NTP) (17/17, 1.34%)  
 Rule 339. (tce\_ingress < 0.01)&(tce\_model\_chisq >= 27390) => (av\_training\_set = NTP) (39/39, 3.07%)  
 Rule 340. (tce\_chisq1 < 24.03)&(tce\_time0bk < 373.84) => (av\_training\_set = NTP) (35/35, 2.76%)  
 Rule 341. (tce\_sma >= 1.03)&(tce\_nkoi < 0.5)&(tce\_depth < 1173500) => (av\_training\_set = NTP) (220/220, 17.32%)  
 Rule 342. (tce\_max\_mult\_ev < 7.3)&(tce\_fwm\_sdec >= 40.78)&(tce\_nkoi < 1.5) => (av\_training\_set = NTP) (64/64, 5.04%)  
 Rule 343. (tce\_model\_dof >= 45595)&(tce\_fwm\_sdec >= 39.67) => (av\_training\_set = NTP) (51/51, 4.02%)  
 Rule 344. (tce\_ldm\_coeff1 >= 0.88) => (av\_training\_set = NTP) (16/16, 1.26%)  
 Rule 345. (tce\_depth < 3.61)&(tce\_prad >= 0.29) => (av\_training\_set = NTP) (21/21, 1.65%)  
 Rule 346. (tce\_robstat < 7.88)&(tce\_ldm\_coeff2 < 0.58) => (av\_training\_set = NTP) (28/28, 2.2%)  
 Rule 347. (tce\_max\_mult\_ev < 7.44)&(tce\_ror < 0) => (av\_training\_set = NTP) (46/46, 3.62%)  
 Rule 348. (tce\_sma in [0.01, 0.01])&(tce\_duration >= 1.5) => (av\_training\_set = NTP) (53/53, 4.17%)  
 Rule 349. (tce\_ldm\_coeff1 < -0.12) => (av\_training\_set = NTP) (3/3, 0.24%)  
 Rule 350. (tce\_dof1 < 45.5)&(tce\_max\_mult\_ev < 10.18)&(tce\_chisq1 >= 28.63) => (av\_training\_set = NTP) (152/152, 11.97%)  
 Rule 351. (tce\_ingress < 0) => (av\_training\_set = NTP) (12/12, 0.94%)  
 Rule 352. (tce\_time0bk < 131.52)&(tce\_ldm\_coeff1 >= 0.3) => (av\_training\_set = NTP) (8/8, 0.63%)  
 Rule 353. (tce\_max\_mult\_ev < 7.59)&(tce\_fwm\_stat < 0.47) => (av\_training\_set = NTP) (34/34, 2.68%)  
 Rule 354. (tce\_max\_mult\_ev < 8.27)&(tce\_chisq2 >= 4753)&(tce\_depth < 54.18) => (av\_training\_set = NTP) (42/42, 3.31%)  
 Rule 355. (tce\_time0bk in [430.95, 450.32]) => (av\_training\_set = NTP) (18/18, 1.42%)  
 Rule 356. (tce\_incl < 7.68) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 357. (tce\_impact < 0) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 358. (tce\_steff >= 13591.5) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 359. (tce\_fwm\_prao < -1.79) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 360. (tce\_dicco\_mra < -10.82) => (av\_training\_set = NTP) (3/3, 0.24%)  
 Rule 361. (tce\_max\_mult\_ev < 8.27)&(tce\_fwm\_sdec >= 45.85) => (av\_training\_set = NTP) (89/89, 7.01%)  
 Rule 362. (tce\_max\_mult\_ev in [7.37, 7.44])&(tce\_dicco\_mra < -0.01) => (av\_training\_set = NTP) (23/23, 1.81%)  
 Rule 363. (tce\_ror < 0)&(tce\_ingress >= 0.02) => (av\_training\_set = NTP) (19/19, 1.5%)  
 Rule 364. (tce\_ingress in [0, 0.01]) => (av\_training\_set = NTP) (33/33, 2.6%)  
 Rule 365. (tce\_period < 0.51)&(tce\_dor >= 2.34) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 366. (tce\_dor in [1, 1]) => (av\_training\_set = NTP) (7/7, 0.55%)  
 Rule 367. (tce\_fwm\_sdec >= 46.32)&(tce\_minmes < -5.45) => (av\_training\_set = NTP) (160/160, 12.6%)  
 Rule 368. (tce\_rsnrmes < -0.11)&(tce\_num\_transits < 860.5)&(tce\_ror >= 0.01) => (av\_training\_set = NTP) (12/12, 0.94%)  
 Rule 369. (tce\_sma < 0.01)&(tce\_fwm\_sdec >= 41.29)&(tce\_robstat < 17.2) => (av\_training\_set = NTP) (83/83, 6.54%)  
 Rule 370. (tce\_ingress < 0)&(tce\_period < 0.57) => (av\_training\_set = NTP) (10/10, 0.79%)  
 Rule 371. (tce\_fwm\_sdeco < -161.8)&(tce\_period >= 2.85) => (av\_training\_set = NTP) (1/1, 0.08%)

Rule 372. (tce\_max\_mult\_ev < 8.03)&(tce\_prad >= 3.79)&(tce\_ror < 0.02) => (av\_training\_set = NTP) (17/17, 1.34%)  
 Rule 373. (tce\_dor < 1.01)&(tce\_sradius < 1.46) => (av\_training\_set = NTP) (8/8, 0.63%)  
 Rule 374. (tce\_fwm\_sra >= 20.06)&(tce\_time0bk < 132.14) => (av\_training\_set = NTP) (11/11, 0.87%)  
 Rule 375. (tce\_max\_mult\_ev < 8.27)&(tce\_ldm\_coeff1 < 0.19)&(tce\_time0bk >= 131.76) => (av\_training\_set = NTP) (15/15, 1.18%)  
 Rule 376. (tce\_max\_mult\_ev < 7.16)&(tce\_period >= 0.88) => (av\_training\_set = NTP) (23/23, 1.81%)  
 Rule 377. (tce\_mesmad >= 81.13)&(tce\_period >= 2.88) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 378. (tce\_rsnrmes < -0.08)&(tce\_max\_sngle\_ev >= 6.72) => (av\_training\_set = NTP) (13/13, 1.02%)  
 Rule 379. (tce\_model\_dof >= 31980)&(tce\_fwm\_sdec >= 41.7)&(tce\_ror < 0.01) => (av\_training\_set = NTP) (281/281, 22.13%)  
 Rule 380. (tce\_max\_mult\_ev in [7.98, 8.03])&(tce\_period < 1.56) => (av\_training\_set = NTP) (8/8, 0.63%)  
 Rule 381. (tce\_max\_mult\_ev in [7.88, 7.92])&(tce\_period < 1.35) => (av\_training\_set = NTP) (7/7, 0.55%)  
 Rule 382. (tce\_max\_mult\_ev < 7.82)&(tce\_model\_snr < 4.43)&(tce\_incl >= 38.22) => (av\_training\_set = NTP) (42/42, 3.31%)  
 Rule 383. (tce\_rmesmad < 6.14)&(tce\_fwm\_sdec >= 41.7) => (av\_training\_set = NTP) (407/407, 32.05%)  
 Rule 384. (tce\_max\_mult\_ev in [7.37, 7.4]) => (av\_training\_set = NTP) (22/22, 1.73%)  
 Rule 385. (tce\_model\_dof in [42070, 42225]) => (av\_training\_set = NTP) (7/7, 0.55%)  
 Rule 386. (tce\_max\_mult\_ev < 7.31)&(tce\_duration >= 6.35) => (av\_training\_set = NTP) (23/23, 1.81%)  
 Rule 387. (tce\_time0bk >= 375.17)&(tce\_period < 358.02) => (av\_training\_set = NTP) (10/10, 0.79%)  
 Rule 388. (tce\_chisq1 < 31.05)&(tce\_period < 44.15) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 389. (tce\_model\_dof in [43785, 43995])&(tce\_time0bk >= 131.53) => (av\_training\_set = NTP) (10/10, 0.79%)  
 Rule 390. (tce\_rsnrmes < -0.1)&(tce\_dof1 < 4725.5)&(tce\_period < 1.13) => (av\_training\_set = NTP) (9/9, 0.71%)  
 Rule 391. (tce\_steff >= 10830.5)&(tce\_duration >= 5.4) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 392. (tce\_dof1 >= 33635)&(tce\_period >= 0.51) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 393. (tce\_eqt >= 6690)&(tce\_dor < 1.06) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 394. (tce\_max\_mult\_ev < 8.5)&(tce\_robstat >= 25.8)&(tce\_ldm\_coeff1 < 0.37) => (av\_training\_set = NTP) (30/30, 2.36%)  
 Rule 395. (tce\_ror < 0.01)&(tce\_robstat >= 32.77)&(tce\_time0bk >= 132.31) => (av\_training\_set = NTP) (10/10, 0.79%)  
 Rule 396. (tce\_max\_mult\_ev < 8.34)&(tce\_smet >= 0.22)&(tce\_period >= 1.18) => (av\_training\_set = NTP) (4/4, 0.31%)  
 Rule 397. (tce\_rsnrmes >= 1.57)&(tce\_impact < 0.18) => (av\_training\_set = NTP) (4/4, 0.31%)  
 Rule 398. (tce\_duration >= 23.36)&(tce\_dor < 1.35)&(tce\_period >= 4.34) => (av\_training\_set = NTP) (3/3, 0.24%)  
 Rule 399. (tce\_dor >= 766.44)&(tce\_time0bk < 165.38) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 400. (tce\_fwm\_prao >= 0.11)&(tce\_period < 0.89) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 401. (tce\_max\_mult\_ev in [7.72, 7.73]) => (av\_training\_set = NTP) (4/4, 0.31%)  
 Rule 402. (tce\_nkoi < 0.5)&(tce\_fwm\_sdec >= 40.9) => (av\_training\_set = NTP) (761/761, 59.92%)  
 Rule 403. (tce\_rsnrmes < -0)&(tce\_dof1 >= 28670) => (av\_training\_set = NTP) (14/14, 1.1%)  
 Rule 404. (tce\_fwm\_sdeco >= 30.69)&(tce\_fwm\_sdec >= 40.89) => (av\_training\_set = NTP) (4/4, 0.31%)  
 Rule 405. (tce\_ldm\_coeff4 >= 0.42)&(tce\_ldm\_coeff1 >= 0.23) => (av\_training\_set = NTP) (4/4, 0.31%)  
 Rule 406. (tce\_model\_dof >= 44275)&(tce\_duration < 2.41)&(tce\_period >= 0.51) => (av\_training\_set = NTP) (35/35, 2.76%)  
 Rule 407. (tce\_max\_mult\_ev in [8.33, 8.34]) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 408. (tce\_max\_mult\_ev < 7.65)&(tce\_dikco\_msky < 0.09) => (av\_training\_set = NTP) (38/38, 2.99%)  
 Rule 409. (tce\_steff >= 8933.5)&(tce\_ldm\_coeff1 < 0.5) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 410. (tce\_max\_mult\_ev < 7.27)&(tce\_duration >= 4.36) => (av\_training\_set = NTP) (33/33, 2.6%)  
 Rule 411. (tce\_slogg < 2.71)&(tce\_sradius < 8.96) => (av\_training\_set = NTP) (3/3, 0.24%)  
 Rule 412. (tce\_dof1 >= 32975)&(tce\_period >= 0.52) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 413. (tce\_rsnrmes < -0.1)&(tce\_ror < 0) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 414. (tce\_steff >= 9650)&(tce\_max\_mult\_ev >= 15.58) => (av\_training\_set = NTP) (3/3, 0.24%)  
 Rule 415. (tce\_dor < 1.01)&(tce\_ror >= 0.02) => (av\_training\_set = NTP) (6/6, 0.47%)  
 Rule 416. (tce\_model\_chisq in [72975, 87810]) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 417. (tce\_max\_mult\_ev in [8.42, 8.45])&(tce\_ldm\_coeff1 >= 0.27) => (av\_training\_set = NTP) (9/9, 0.71%)  
 Rule 418. (tce\_max\_mult\_ev in [8.29, 8.3])&(tce\_ror < 0) => (av\_training\_set = NTP) (3/3, 0.24%)  
 Rule 419. (tce\_model\_dof in [38735, 38760]) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 420. (tce\_fwm\_sdeco < -32.85)&(tce\_impact < 0.04) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 421. (tce\_max\_mult\_ev in [8.24, 8.24]) => (av\_training\_set = NTP) (5/5, 0.39%)  
 Rule 422. (tce\_rminmes >= 1.29)&(tce\_minmes >= -11.33)&(tce\_ror < 0.01) => (av\_training\_set = NTP) (7/7, 0.55%)  
 Rule 423. (tce\_rminmes in [0.74, 0.74]) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 424. (tce\_rminmes in [0.75, 0.75]) => (av\_training\_set = NTP) (3/3, 0.24%)  
 Rule 425. (tce\_rminmes in [0.75, 0.75]) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 426. (tce\_ror < 0)&(tce\_model\_snr >= 10.68)&(tce\_period >= 1.22) => (av\_training\_set = NTP) (7/7, 0.55%)  
 Rule 427. (tce\_fwm\_srao >= 44.75)&(tce\_time0bk < 131.7) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 428. (tce\_rminmes >= 0.75)&(tce\_impact >= 0.99)&(tce\_time0bk < 133) => (av\_training\_set = NTP) (3/3, 0.24%)  
 Rule 429. (tce\_max\_mult\_ev in [8.23, 8.23]) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 430. (tce\_mesmad >= 8.13)&(tce\_max\_mult\_ev < 12.56)&(tce\_model\_chisq < 18905) => (av\_training\_set = NTP) (3/3, 0.24%)  
 Rule 431. (tce\_max\_mult\_ev in [7.64, 7.65]) => (av\_training\_set = NTP) (4/4, 0.31%)  
 Rule 432. (tce\_rsnrmes < -0.05)&(tce\_time0bk >= 132.94)&(tce\_num\_transits < 28.5) => (av\_training\_set = NTP) (5/5, 0.39%)  
 Rule 433. (tce\_duration in [14.17, 14.31])&(tce\_period < 24.57) => (av\_training\_set = NTP) (9/9, 0.71%)



Rule 434. (tce\_duration >= 14.83)&(tce\_period < 2.86) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 435. (tce\_rmesmad >= 7119.5)&(tce\_robstat < 2975.5) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 436. (tce\_dikco\_mdec < -8.18)&(tce\_max\_sngle\_ev < 4.4) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 437. (tce\_duration >= 14.83)&(tce\_smet >= 0.33) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 438. (tce\_max\_mult\_ev in [8.17, 8.19]) => (av\_training\_set = NTP) (8/8, 0.63%)  
 Rule 439. (tce\_model\_dof in [38805, 38930])&(tce\_period < 3.88) => (av\_training\_set = NTP) (4/4, 0.31%)  
 Rule 440. (tce\_impact in [0, 0]) => (av\_training\_set = NTP) (4/4, 0.31%)  
 Rule 441. (tce\_duration in [14.89, 14.94]) => (av\_training\_set = NTP) (4/4, 0.31%)  
 Rule 442. (tce\_max\_mult\_ev in [7.62, 7.63])&(tce\_period >= 0.93) => (av\_training\_set = NTP) (4/4, 0.31%)  
 Rule 443. (tce\_ingress in [0.01, 0.01]) => (av\_training\_set = NTP) (6/6, 0.47%)  
 Rule 444. (tce\_duration >= 14.96)&(tce\_ldm\_coeff4 >= 0.37) => (av\_training\_set = NTP) (9/9, 0.71%)  
 Rule 445. (tce\_ingress in [0.01, 0.01]) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 446. (tce\_model\_dof in [42305, 42395]) => (av\_training\_set = NTP) (10/10, 0.79%)  
 Rule 447. (tce\_slogg in [2.84, 2.92]) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 448. (tce\_smet < -1.07)&(tce\_model\_dof >= 32225) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 449. (tce\_duration >= 14.96)&(tce\_depth < 22.05)&(tce\_period >= 5.87) => (av\_training\_set = NTP) (4/4, 0.31%)  
 Rule 450. (tce\_rsnrmes < -0.08)&(tce\_prad >= 5.16) => (av\_training\_set = NTP) (5/5, 0.39%)  
 Rule 451. (tce\_ingress in [0.01, 0.01]) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 452. (tce\_dicco\_mra < -6.71)&(tce\_duration < 0.96) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 453. (tce\_max\_mult\_ev < 7.52)&(tce\_max\_sngle\_ev >= 10.42) => (av\_training\_set = NTP) (35/35, 2.76%)  
 Rule 454. (tce\_dikco\_mdec < -7.24)&(tce\_model\_dof >= 37225) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 455. (tce\_model\_dof in [43275, 43395]) => (av\_training\_set = NTP) (6/6, 0.47%)  
 Rule 456. (tce\_rminmes in [0.48, 0.48]) => (av\_training\_set = NTP) (6/6, 0.47%)  
 Rule 457. (tce\_rminmes in [0.49, 0.49]) => (av\_training\_set = NTP) (4/4, 0.31%)  
 Rule 458. (tce\_rminmes in [0.51, 0.51]) => (av\_training\_set = NTP) (11/11, 0.87%)  
 Rule 459. (tce\_rminmes in [0.75, 0.76]) => (av\_training\_set = NTP) (5/5, 0.39%)  
 Rule 460. (tce\_rminmes in [0.78, 0.78]) => (av\_training\_set = NTP) (3/3, 0.24%)  
 Rule 461. (tce\_rminmes in [0.8, 0.8]) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 462. (tce\_rminmes in [0.8, 0.81]) => (av\_training\_set = NTP) (4/4, 0.31%)  
 Rule 463. (tce\_duration in [15.37, 15.52]) => (av\_training\_set = NTP) (8/8, 0.63%)  
 Rule 464. (tce\_mesmad in [8.6, 8.68]) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 465. (tce\_impact in [0.01, 0.01]) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 466. (tce\_max\_sngle\_ev < 3.29)&(tce\_time0bk >= 142.23)&(tce\_period < 14.11) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 467. (tce\_steff >= 8114.5)&(tce\_ldm\_coeff1 < 0.45) => (av\_training\_set = NTP) (5/5, 0.39%)  
 Rule 468. (tce\_impact in [0.01, 0.01]) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 469. (tce\_steff in [8388.5, 8428]) => (av\_training\_set = NTP) (8/8, 0.63%)  
 Rule 470. (tce\_ldm\_coeff4 in [0.37, 0.38])&(tce\_period < 1.29) => (av\_training\_set = NTP) (6/6, 0.47%)  
 Rule 471. (tce\_minmes in [-2.03, -2.02])&(tce\_period < 1.73) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 472. (tce\_dof1 < 84.5)&(tce\_rsnrmes < -0) => (av\_training\_set = NTP) (10/10, 0.79%)  
 Rule 473. (tce\_slogg in [3.47, 3.48])&(tce\_period >= 0.58) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 474. (tce\_model\_dof >= 28405)&(tce\_model\_chisq < 19040)&(tce\_period >= 0.62) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 475. (tce\_ingress in [0.01, 0.01]) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 476. (tce\_model\_dof in [28625, 28650])&(tce\_period < 2.97) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 477. (tce\_rminmes in [0, 0]) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 478. (tce\_model\_dof in [29035, 29050])&(tce\_period >= 0.65) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 479. (tce\_rminmes in [1.18, 1.18])&(tce\_period >= 0.58) => (av\_training\_set = NTP) (3/3, 0.24%)  
 Rule 480. (tce\_chisq2 >= 114650)&(tce\_ingress < 0.09) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 481. (tce\_model\_dof in [31460, 31485]) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 482. (tce\_model\_dof in [40030, 40280]) => (av\_training\_set = NTP) (12/12, 0.94%)  
 Rule 483. (tce\_fwm\_sdeco in [-3.93, -3.86]) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 484. (tce\_fwm\_srao in [-4.53, -4.27]) => (av\_training\_set = NTP) (6/6, 0.47%)  
 Rule 485. (tce\_impact in [0.01, 0.01]) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 486. (tce\_ror in [0, 0]) => (av\_training\_set = NTP) (2/2, 0.16%)  
 Rule 487. (tce\_rsnrmes in [1.23, 1.23]) => (av\_training\_set = NTP) (4/4, 0.31%)  
 Rule 488. (tce\_time0bk in [131.64, 131.64])&(tce\_period < 0.91) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 489. (tce\_fwm\_sdeco in [-2.75, -2.74]) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 490. (tce\_rsnrmes in [1.25, 1.25])&(tce\_period < 1.84) => (av\_training\_set = NTP) (1/1, 0.08%)  
 Rule 491. (tce\_fwm\_stat in [0.39, 0.4]) => (av\_training\_set = NTP) (4/4, 0.31%)  
 Rule 492. (tce\_ldm\_coeff2 in [1.02, 1.02]) => (av\_training\_set = NTP) (5/5, 0.39%)  
 Rule 493. (tce\_prad in [0.92, 0.92]) => (av\_training\_set = NTP) (1/1, 0.08%)

Number of Rules: 493

---

Artículo AAI-17

---

# Effectiveness of Rule Induction Algorithms for Classification of Kepler Threshold Crossing Events

**Sergio Morales and Francisco Torres-Rojas**

Instituto Tecnológico de Costa Rica  
San José, Costa Rica  
{smorales, torres}@itcr.ac.cr

## Abstract

The Kepler Objects of Interest (KOI) catalogs are composed of thousands of interesting, exoplanet transit-like signals produced by the Kepler Science Operations Center. The number is significant enough to make it unfeasible and impractical to inspect them by human eyes, making machine learning an important step in the exoplanet detection data pipeline. In this paper, we propose a machine learning approach centering around the use of the rule induction algorithm MODLEM in order to classify these signals between 3 classes: non-transiting phenomena, astrophysical false positives, and planet candidates, as well as extract features important to classification. A comparison of the classification accuracy between various ensemble approaches using MODLEM as a base classifier is provided, showing that a boosting approach is more suited to this scenario.

## Introduction

The Kepler telescope is a single-purpose spacecraft with the purpose of collecting photometric series of a single area of space during long periods of time. These photometric series are subject to the Kepler science processing pipeline (Jenkins et al. 2010), which performs a series of transformation upon the collected data and obtains a series of Threshold Crossing Events, or TCEs for short, describing sets of features in the light-curve of stars observed in the area the spacecraft is designated to observe that may indicate the existence of a transiting planet around said star.

TCEs are then further processed and vetted by the Kepler TCE Review Team in order to further refine the candidate selection for potential extrasolar planet findings (Batalha et al. 2013). Those that are determined to most likely resemble the signature of a transiting planet are designated as Kepler Objects of Interest, or KOIs. Even after said classification process, it's possible that phenomena such as eclipsing binaries may produce features similar to those observed in transiting planets, resulting in false positives in the KOI designation process, requiring further validation from further data from Kepler or other sources.

Due to the sheer amount of observations collected by the spacecraft, efforts to automate this vetting process have been performed, resulting in those such as the Kepler Project's Autovetter (Joseph H. Catanzarite 2015), which classifies TCEs in 3 different classes: Non-transiting Phenomena, which corresponds to those TCEs which fail the TCE Review Team's triage process, planet candidates which correspond to curves consistent with transiting planets and no evidence to the contrary, and astrophysical false-positives, such as the aforementioned eclipsing binaries. The autovetter employs Random Forest as its classification algorithm.

In this paper, we propose to explore the effectiveness of rule-induction algorithms in conjunction with common ensemble classification approaches in order to classify Kepler Threshold Crossing Events, using the same classes as the aforementioned Autovetter described above, as well as its training data set, the Q1-Q17 DR24 table (Joseph H. Catanzarite 2015).

The paper is organized as follows: In section 2 we provide a brief overview of some of the most relevant previous work on the subject, followed by section 3 in which we provide a brief description of the MODLEM algorithm, used as base classifier for the ensemble learning approaches evaluated, as well as the rationale for this choice. We then describe the features provided as part of the Kepler dataset used as input to these algorithms in section 4. Section 5 consists on an overview of the methodology used to classify this data is provided, along with its results. Finally, we discuss potential future work as well as conclusions provided by the performed evaluation in sections 6 and 7, respectively.

## Related Work

Much work has been performed on the Kepler data in general and also in the classification of said data, such as efforts to detect planet-like signatures in Kepler lightcurve data (Debray and Wu 2013), however our work is primarily inspired by the automatic classification efforts of McCauliff et al (McCauliff et al. 2014). They present a machine learning approach using Random Forests to classify TCEs into the same three classes (PC, NTP and AFP) as the approach explored in this paper does.

The decision to explore the effectiveness of rule induction algorithms as a base classifier is inspired in works such as Stefanowski’s (Stefanowski 2007) who first introduced the MODLEM algorithm.

## Rule Induction

Rule induction machine learning approaches center on the supervised classification of data instances based on a series of rules following an *IF Condition THEN Class* pattern derived from a training data set, where the *Condition* may be composed of conjunctive or disjunctive rules:

$$\begin{aligned} \text{IF } A = 1 \text{ AND } B = 1 \text{ THEN class} &= x \\ \text{IF } C = 1 \text{ AND } D = 1 \text{ THEN class} &= x \end{aligned}$$

Work on assessing the effectiveness of this kind of approaches has been performed, demonstrating satisfying accuracy in large and noisy data sets when employed in conjunction with ensemble machine learning approaches (Stahl and Bramer 2012).

Rules present several advantages over other approaches to decision representation, thanks to their compactness, the ease with which they can be naturally understood by humans and used as input for further operations and the way in which they can be represented in an ordered way to represent which are more important over others, clearly outlining important features for data mining purposes. These approaches iteratively generate a series of rules by selecting the best (most class-discriminating) feature in each iteration, based on the provided training data set:

```
Rule_set rules = new Rule_set()
while Stop_Criterion not satisfied:
    Rule = Learn_Rule()
    Remove instances covered by Rule
    rules.add(Rule)
```

After each rule is introduced into the ruleset, all instances which classification is covered by the selected rule are removed from the data set, and a new rule is generated using the resulting reduced data set as an input. Typically, the induced rule corresponds to one that covers the class that contains the least instances within the training set, in order to attempt to generate a minimal rule set (Stahl and Bramer 2012).

Some examples of rule induction classifiers are *PRISM*, introduced by Bramer, CN2, and MODLEM, the last of which is used as base classifier for those evaluated in this article. MODLEM is based on the concept on sequential coverage, and as previously described attempts to generate a minimal ruleset for each class by covering all positive examples in each class and avoiding negative ones, by performing an exhaustive search of all elemental features within the data set’s instances and removing matching instances for each rule, in a similar fashion as described

above (Grzymala-Busse and Stefanowski 2001).

Another important part of MODLEM’s appeal for the evaluation discussed here consists on its ability to handle numeric features, as other approaches such as PRISM are unable to handle non-nominal attributes. Numeric attributes are handled in MODLEM by being represented through equality relationships between values found in the data set’s instances’ features during the supervised learning process (Stefanowski 2007).

## KOI Catalog

The raw data retrieved from the Kepler spacecraft is transformed by a refinement pipeline performed by the Kepler Mission Science Operations Center (Jenkins et al. 2010), in order to account for known issues in the spacecraft’s photometric equipment and extract features that will further help identify potential KOIs and discard false positives. This process is roughly divided in the following stages:

1. **Pixel-level calibration:** Cleaning operations are performed on the retrieved images’ pixel data, in order to correct known disturbances caused by the photometric equipment onboard the Kepler spacecraft.
2. **Photometric Analysis:** Before photometry and astrometry can be extracted from the calibrated pixel time series, the Pipeline detects noise in the background pixel data thought to be produced by small dust particles from Kepler achieving escape velocity after micrometeorite hits and reflecting sunlight into the barrel of the telescope.
3. **Transiting Planet Search:** TPS searches for transiting planets by ”stitching” the quarterly segments of data together to remove gaps and edge effects and then applies the wavelet-based, adaptive-matched filter (Jenkins 2002b).
4. **Data Validation:** A series of tests and diagnostics designed to establish confidence levels for detected transiting planets candidates in order to remove false positives and eclipsing binaries from the KOI database is performed.

The data set used for the training and evaluation of the machine learning algorithms used in this research corresponds to Kepler’s KOI Catalog for the quarters Q1 to Q17, Data Release 24 (NASA 2015). The main motivation behind selecting this data set consists on the fact that it is the same data with which the Kepler Autovetter has been trained, and as such contains training class labels that correspond to those we’re interested in (PC, NTP and AFP) (Joseph H. Catanzarite 2015). This data contains a total of 3600 instances of the Planetary Candidate class, 9596 Astronomical False-Positives and 2541 Non-Transiting Phenomena instances.

In regards to the features included in this data set for each sample (corresponding to a Kepler Object of Interest), these can be intuitively classified in the following sub-groups,

according to the stage in the Kepler data pipeline in which they were retrieved (NASA 2015):

1. **Target Labels and Flags:** Basic identification attributes for the sample, such as the TCE id number, its assigned planet number, the set of data to which it belongs and the date in which it was retrieved.
2. **Transit Fit Parameters:** Transit features produced by the Kepler data pre-processing pipeline, adjusted by a Mandel-Agol model (Mandel and Agol 2002). Contains features such as the detected orbital period, the planet-star radius ratio and their separation.
3. **Scaled Planetary Parameters:** Features related to the target star combined with the previous features in order to establish planetary features in known physical units.
4. **Stellar Parameters:** Features belonging to the target star in which the threshold-crossing event was identified such as its temperature, gravity, metallicity, radius and mass.
5. **Light Curve-Based KOI Vetting Statistics:** Advanced features derived from the Kepler lightcurve analysis. This module applies an adaptative filter in order to produce features previously established as relevant for the classification process (Jenkins 2002a).
6. **Pixel-Based KOI Vetting Statistics:** Planetary transit false positives are commonly caused by light curve contamination from an eclipsing binary falling partially within the target aperture (i.e., the pixels used to collect and sum target flux). Two pixel analysis methods are used to identify such eclipsing binaries (Bryson et al. 2010).
7. **Autovetter Parameters:** Attributes associated to the training and classification of the data set for the Autovetter module previously mentioned. Among these the training class label is provided, as well as its classification result, accuracy and others.

## Evaluation and Results

In this section we provide a brief overview of the pre-processing procedures applied on the retrieved Kepler data set (described above), following by the methodology employed to evaluate the machine learning algorithms targeted by this research and its results.

### Data Set Pre-processing

A series of transformations were applied on the retrieved data set. The order of samples was randomized, and the proportion between classes was balanced such that all three class labels were represented by the same amount of samples, for a total of 7623 Threshold Crossing Events, with 2541 assigned to each class. The selection of samples for each class was also randomized.

Some of the attributes on the data set were removed, namely those that could be used as discriminants such as identifying information, and those related to the Autovetter classification process, except for the training label which was used as training label for the algorithms to be evaluated.

## Experimental Design

An experiment was designed in order to determine the effect on classification accuracy of samples, based on various approaches to ensemble learning using MODLEM as a base classifier, selecting various other experimental input factors in order to measure their effect in comparison with the classifiers and also establish their influence on accuracy. The selected factors are as follows:

1. Classification Method
  - (a) MODLEM
  - (b) Bagging MODLEM
  - (c) Boosting MODLEM
  - (d) Random Forest
2. Data Volume
  - (a) Full data set (100%)
  - (b) Half data set (50%)
3. Sample Features
  - (a) All features
  - (b) Pixel analysis features removed
4. Training-Test Set Ratio
  - (a) 25-75
  - (b) 50-50
  - (c) 75-25

The chosen factors other than the classification algorithms are related to the supervised learning methodology, and seek to influence accuracy by varying the amount of data available for the classification algorithms to build their classification model, which is expected to increase as the amount of samples available during the learning stage increases (Halevy, Norvig, and Pereira 2009).

In order to perform at least on run of every possible combination of the aforementioned factors, a total of 48 ( $4 \times 2 \times 2 \times 3$ ) executions were needed. In order to guarantee a reliable degree of confidence in the experiment, in addition each of these was repeated 5 times for a total of 240 runs, which were randomized in order to further avoid potential external factors related to hardware or software conditions that could influence the result of the experiment. Data sets that fit all requirements regarding data size, training and test set ratios as well as feature sets were produced, for a total of 12 training data sets and 12 test sets. The experiment was run using the Java-based Weka platform (Hall et al. 2009) on the Costa Rica National Collaboratory of Advanced Computing cluster.

## Results

The results were validated using ANOVA (Montgomery 2001), verifying independence of observations, normality and homogeneity of variances within the experiment. All observed factors were considered as significant by the ANOVA evaluation, with both algorithms and the inclusion or exclusion of the later-stage pixel-level attributes having the biggest influence on the classification accuracy.

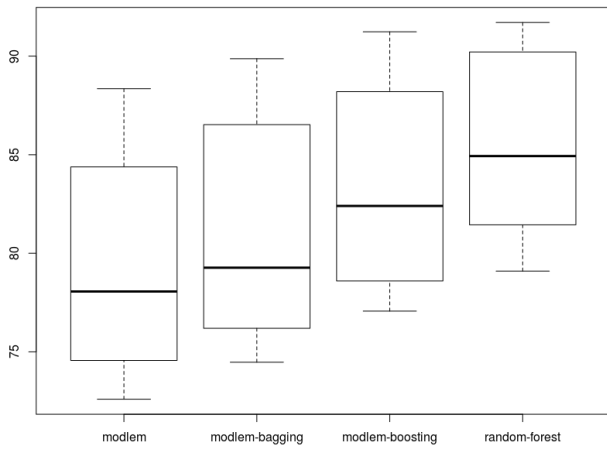


Figure 1: Effect of measured classification methods on accuracy.

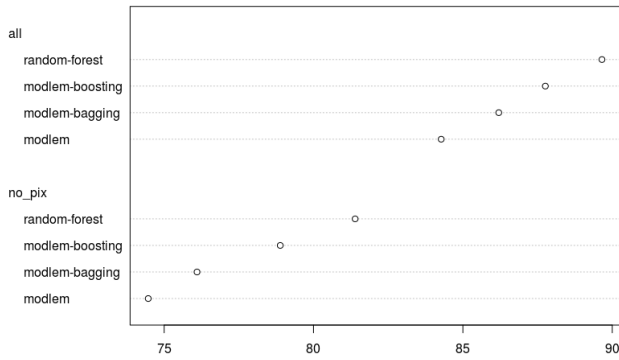


Figure 2: Effect of measured classification methods and attribute exclusion on accuracy.

Figure 1 shows the effect of the evaluated algorithms on the classification accuracy of the Kepler data. Random Forest was used as a base comparison due to its use in the NASA autovetter, and indeed it performs better than any of the other measured algorithms. In regards to using MODLEM as a base classifier, it is when it is combined with the boosting meta-classifier that it performs better, with very close results to those of Random Forest.

When two-way interactions are taken into account, no major divergence from this trend can be noticed. As expected, the more information is available for classifiers to build their model, the more the classification accuracy increases, as shown in figures 2, 3 and 4.

As mentioned above, another output of interest was the classification model generated by the rule induction classifiers. The ease with which their output - ordered lists of rules - can be naturally understood by humans or used as input for further operations were thought to potentially offer insight into the characterization of transit-like signatures. In figure 5 a small sample of one of the generated classification mod-

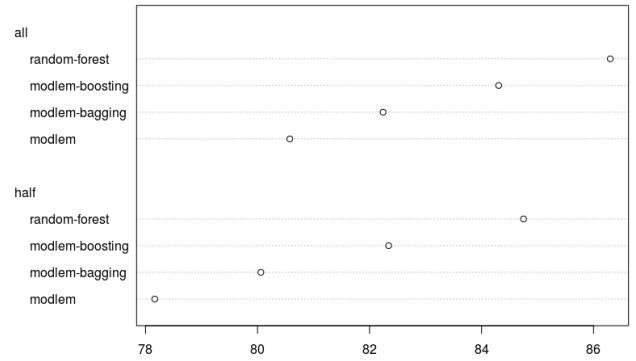


Figure 3: Effect of measured classification methods and sample size on accuracy.

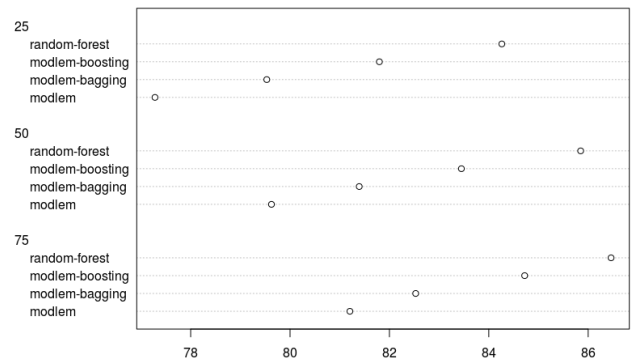


Figure 4: Effect of measured classification methods and train-test ratio on accuracy.

els is reproduced, focusing on the rules generated for one of the classification labels in order of relevance (*PC*).

```

1. (tce_nkoi >= 2.5)&(tce_smet >= -0.25)
=> (class = PC) (229/229, 18.03%)

2. (tce_nkoi>=1.5)&(tce_fwm_sdec>=40.68)
&(tce_ingress >= 0.02)
=> (class = PC) (401/401, 31.57%)

3. (tce_nkoi >= 1.5)&(tce_mesmad < 0.67)
&(tce_robstat<93.67)
=> (class = PC) (129/129, 10.16%)

4. (tce_nkoi >= 1.5)&(tce_steff <6340.5)
&(tce_dicco_msky < 0.46)
=> (class = PC) (346/346, 27.24%)

5. (tce_sradius < 0.51)&(tce_ror < 0.11)
&(tce_rminmes < 0.2)
=> (class = PC) (36/36, 2.83%)

6. (tce_rminmes<0.19)&(tce_chisq2< 1632)
&(tce_fwm_sdec >= 43)
=> (class = PC) (336/336, 26.46%)

7. (tce_minmes>=-3.55)&(tce_smet>= 0.23)
&(tce_model_snr>=17.28)
=> (class = PC) (48/48, 3.78%)

8. (tce_minmes>=-3.55)&(tce_fwm_sra>=20.03)
=> (class = PC) (13/13, 1.02%)

9. (tce_chisq2<258.55)&(tce_minmes>=-2.7)
&(tce_dicco_msky < 3.38)
=> (class = PC) (88/88, 6.93%)

```

Figure 5: Subset of a MODLEM-generated classification model.

Some of the highlighted attributes in these rules are those of target star metallicity (*tce\_smet*) and the amount of TCE observations on the target star the sample that's being observed belongs to (*tce\_nkoi*), pointing towards the importance of both planetary and stellar properties in the classification process.

### Future Work

Results shown above lead us to various new potential avenues of research that may be enquired upon to both determine optimal conditions for the classification of the available Kepler data, and improve upon the MODLEM algorithm, both by combining it with ensemble methods, and determining when it can be employed to its fullest. A potential approach of interest would be to create a custom ensemble classifier to be used with MODLEM as a base classifier, incorporating properties of methods such as Random Forest, which was shown here to be a more accurate classifier for the Kepler Data. Attempting to integrate

Bramer's Random PRISM (Stahl and Bramer 2012), which incorporates Random Forests' random subspace method for feature pruning could very well lead to improved performance in accuracy.

Results also show that the execution of MODLEM is significantly more time-consuming than the Random Forest approach we compared it to. This could be shown to be due to implementation details in the MODLEM package employed, however reducing the time complexity in the algorithm could make it more fitting for large datasets such as Kepler's. Similarly, this research didn't perform any sort of pre-processing for the dataset beyond what was described during the previous section; a more exhaustive feature pre-selection as those performed by McCauliff (McCauliff et al. 2014) would presumably improve upon the classification accuracy of all algorithms evaluated here.

### Conclusion

Machine learning approaches can not only be used to partially automate the process of exoplanet discovery with satisfying rates of accuracy - indeed considering the amount of data retrieved and curated by the Kepler project, the use of automatic classification methods becomes necessary - but as shown in this paper they can also be used to provide valuable insight into the relevancy of features within these data sets, in a way that can be easily comprehended by researchers and from which new information regarding transit-like signatures detection may be accrued.

In general, we've shown that as expected, classification methods, data volume and feature selection play a significant role in the classification accuracy of Threshold Crossing Events as provided by the NASA Exoplanet Archives, with ensemble methods based on the boosting meta-algorithm such as Random Forest and the boosting-enabled MODLEM performing better overall than other approaches in this aspect. Data volume is known to improve accuracy of machine learning as it increases due to a bigger sample selection leading to more of these providing better learning data sets (Halevy, Norvig, and Pereira 2009), however its non-proportional growth seems to point to there being diminishing returns in this aspect.

Feature selection on the other hand plays a very significant role in accuracy, with features derived in the latter part of the Kepler data pipeline being the most relevant when used in the training portion of supervised learning for any of the algorithms used.

### Acknowledgements

We would like to thank Jeudy Blanco for assistance and guidance in the planning stages of the project, Szymon Wojciechowski for use of the MODLEM implementation for Weka, and the Costa Rica National Collaboratory of Advanced Computing (CNCA) for use of its computational cluster. This research has made use of the NASA Exoplanet Archive, which is operated by the California Institute of

Technology, under contract with NASA under the Exoplanet Exploration Program.

## References

- Batalha, N. M.; Rowe, J. F.; Bryson, S. T.; Barclay, T.; Burke, C. J.; Caldwell, D. A.; Christiansen, J. L.; Mullally, F.; Thompson, S. E.; Brown, T. M.; Dupree, A. K.; Fabrycky, D. C.; Ford, E. B.; Fortney, J. J.; Gilliland, R. L.; Isaacson, H.; Latham, D. W.; Marcy, G. W.; Quinn, S. N.; Ragozzine, D.; Shporer, A.; Borucki, W. J.; Ciardi, D. R.; Thomas N. Gautier, I.; Haas, M. R.; Jenkins, J. M.; Koch, D. G.; Lissauer, J. J.; Rapin, W.; Basri, G. S.; Boss, A. P.; Buchhave, L. A.; Carter, J. A.; Charbonneau, D.; Christensen-Dalsgaard, J.; Clarke, B. D.; Cochran, W. D.; Demory, B.-O.; Desert, J.-M.; Devore, E.; Doyle, L. R.; Esquerdo, G. A.; Everett, M.; Fressin, F.; Geary, J. C.; Girouard, F. R.; Gould, A.; Hall, J. R.; Holman, M. J.; Howard, A. W.; Howell, S. B.; Ibrahim, K. A.; Kinemuchi, K.; Kjeldsen, H.; Klaus, T. C.; Li, J.; Lucas, P. W.; Meibom, S.; Morris, R. L.; Prša, A.; Quintana, E.; Sanderfer, D. T.; Sasselov, D.; Seader, S. E.; Smith, J. C.; Steffen, J. H.; Still, M.; Stumpe, M. C.; Tarter, J. C.; Tenenbaum, P.; Torres, G.; Twicken, J. D.; Uddin, K.; Cleve, J. V.; Walkowicz, L.; and Welsh, W. F. 2013. Planetary candidates observed by kepler. iii. analysis of the first 16 months of data. *The Astrophysical Journal Supplement Series* 204(2):24.
- Bryson, S. T.; Tenenbaum, P.; Jenkins, J. M.; Chandrasekaran, H.; Klaus, T.; Caldwell, D. A.; Gilliland, R. L.; Haas, M. R.; Dotson, J. L.; Koch, D. G.; and Borucki, W. J. 2010. The Kepler Pixel Response Function. 713:L97–L102.
- Debray, A., and Wu, R. 2013. Astronomical implications of machine learning.
- Grzymala-Busse, J. W., and Stefanowski, J. 2001. Three discretization methods for rule induction. *International Journal of Intelligent Systems* 16(1):29–38.
- Halevy, A.; Norvig, P.; and Pereira, F. 2009. The unreasonable effectiveness of data. *IEEE Intelligent Systems* 24(2):8–12.
- Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The weka data mining software: An update. *SIGKDD Explor. Newsl.* 11(1):10–18.
- Jenkins, J. M.; Caldwell, D. A.; Chandrasekaran, H.; Twicken, J. D.; Bryson, S. T.; Quintana, E. V.; Clarke, B. D.; Li, J.; Allen, C.; Tenenbaum, P.; Wu, H.; Klaus, T. C.; Mid-dour, C. K.; Cote, M. T.; McCauliff, S.; Girouard, F. R.; Gunter, J. P.; Wohler, B.; Sommers, J.; Hall, J. R.; Uddin, A. K.; Wu, M. S.; Bhavsar, P. A.; Cleve, J. V.; Pletcher, D.; Dotson, J. A.; Haas, M. R.; Gilliland, R. L.; Koch, D. G.; and Borucki, W. 2010. Overview of the kepler science processing pipeline. *The Astrophysical Journal Letters* 713(2):L87.
- Jenkins, J. M. 2002a. The Impact of Solar-like Variability on the Detectability of Transiting Terrestrial Planets. 575:493–505.
- Jenkins, J. M. 2002b. The impact of solar-like variability on the detectability of transiting terrestrial planets. *The Astrophysical Journal* 575(1):493.
- Joseph H. Catanzarite. 2015. Autovetter planet candidate catalog for q1-q17 data release 24.
- Mandel, K., and Agol, E. 2002. Analytic Light Curves for Planetary Transit Searches. 580:L171–L175.
- McCauliff, S.; Jenkins, J. M.; Catanzarite, J.; Burke, C. J.; Coughlin, J. L.; Twicken, J. D.; Tenenbaum, P.; Seader, S.; Li, J.; and Cote, M. 2014. Automatic classification of kepler threshold crossing events.
- Montgomery, D. 2001. *Design and Analysis of Experiments*. John Wiley & Sons.
- NASA. 2015. Q1-q17 data release 24.
- Stahl, F., and Bramer, M. 2012. Random prism: an alternative to random forests. In *Thirty-first SGA International Conference on Artificial Intelligence*.
- Stefanowski, J. 2007. *Transactions on Rough Sets VI: Commemorating the Life and Work of Zdzisław Pawlak, Part I*. Berlin, Heidelberg: Springer Berlin Heidelberg. chapter On Combined Classifiers, Rule Induction and Rough Sets, 329–350.