|  |
| --- |
| Close-up image showing the leaf-sides of two oversized books side-by-side on a bookshelf, with additional books in soft focus background |
| [Document title]  [Document subtitle] |
| |  |  |  | | --- | --- | --- | | Windows User | [Date] | [Course title] | |

Contenido

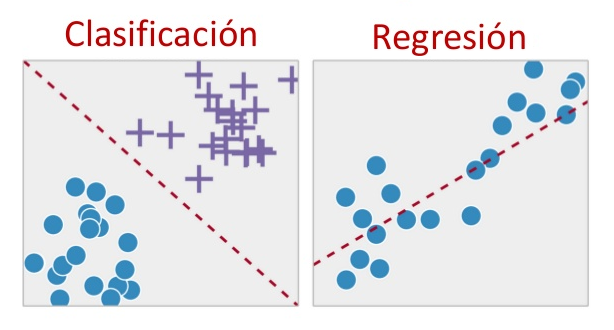
[Introducción 2](#_Toc514358492)

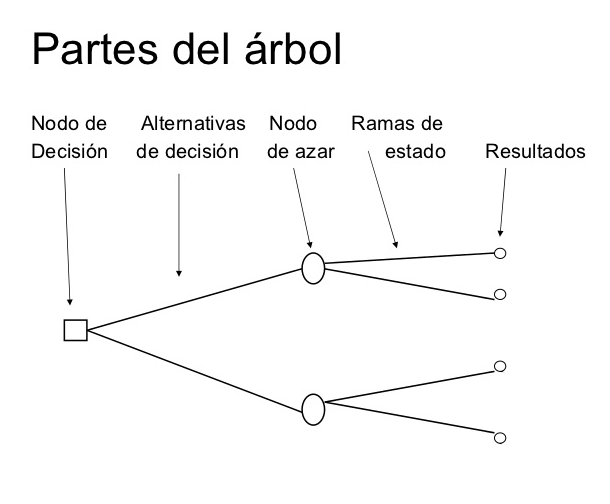
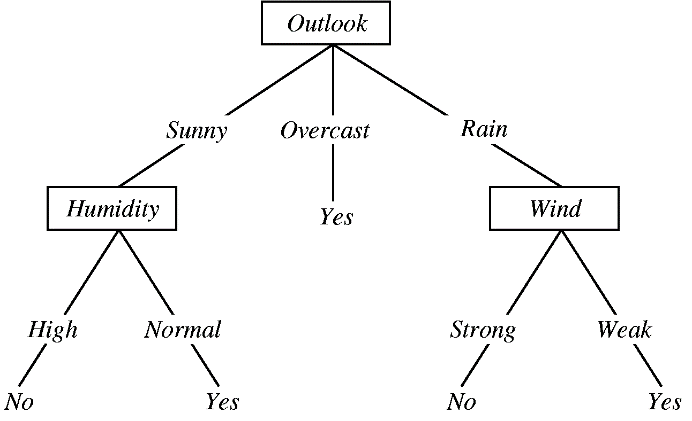
[Objetivos 3](#_Toc514358493)

[Dataset 4](#_Toc514358494)

# Introducción

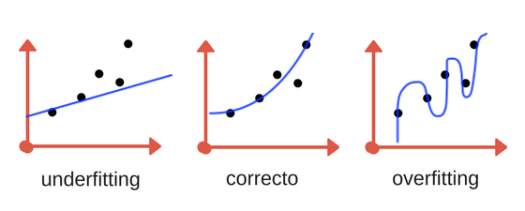
El Machine Learning o Aprendizaje Automático es una rama de la inteligencia artificial que nace durante la segunda mitad del siglo XX. Su principal objetivo es desarrollar métodos capaces de generalizar comportamientos y reconocer patrones a partir de la información de entrada, permitiendo que computadoras encuentren información y se comporten de cierta forma sin que hayan sido explícitamente programadas para ello.  
  
Existen principalmente 3 categorias de algoritmos de aprendizaje automático o machine learning: aprendizaje supervisado (supervised learning), aprendizaje no supervisado (unsupervised learning) y aprendizaje por refuerzo (reinforcement learning).  
Los algoritmos de aprendizaje supervisado tienen como objetivo identificar un modelo a partir de datos de entrenamiento en los cuales se conoce el valor verdadero de salida, y usar este valor para predecir datos futuros o desconocidos.  
El aprendizaje supervisado tiene 2 tipos de actividad como resultado de la aplicación de un algoritmo de machine learning. Las 2 actividades pueden ser de clasificación o de regresión.  
La clasificación tiene como objetivo predecir etiquetas o clases. Estas etiquetas son valores discretos no ordenados que pueden ser entendidos como categorías. Un ejemplo son los filtros de spam, donde a partir de un set de correos electrónicos marcados correctamente como SPAM / NO-SPAM, se entrena un modelo para determinar qué etiqueta se debe asignar a un próximo correo revisando el valor/contenido de sus atributos (contenido, asunto, remitente, etc.).  
Por otro lado, la regresión, se utilizan para la predicción de valores continuos. En el análisis de regresiones, se entrega un número o números (predictor) que explican el valor de salida, es decir, los valores de entrada guardan relación con la variable a explicar. Dentro de las regresiones, la más común es la regresión lineal, que mediante una ecuación de primer orden busca explicar un valor de salida. Un ejemplo de regresión podría ser predecir el volumen de ventas de un próximo periodo para un sitio de e-commerce.



En el presente informe se utilizará un algoritmo perteneciente a la categoría de aprendizaje supervisado y se realizará una clasificación a través del mismo, hay versiones del algoritmo que también soporta tareas de regresión.  
El algoritmo en cuestión, se basa en la **inducción** a partir de la creación de árboles de decisión y es conocido simplemente como arboles de decisión, decision trees o IDT (induction decision trees).  
Muy resumidamente, un árbol puede ser "aprendido" mediante el fraccionamiento del conjunto inicial en subconjuntos basados en una prueba de valor de atributo. Este proceso se repite en cada subconjunto derivado de una manera recursiva llamada particionamiento recursivo. La recursividad termina cuando el subconjunto en un nodo tiene todo el mismo valor de la variable objetivo, o cuando la partición ya no agrega valor a las predicciones.  
En el presente trabajo se utilizará el algoritmo de árbol de decisión J48, es una implementación que forma parte de una herramienta de datamining open source llamada Weka.  
MAS DETALLES Y LIMITACIONES DEL ALGORTIMO   
CAMBIAR GRAFICOS  
 

# Objetivos

Todo algoritmo tiene sus fortalezas y debilidades, no hay un algoritmo perfecto que entregue excelentes modelos sin importar el problema a resolver. El presente trabajo tiene como objetivo someter al algoritmo J48 a una serie de entrenamientos cuyos resultados puedan ser evaluados y asi poder realizar un análisis sobre el comportamiento del mismo.  
Analizando los resultados de las pruebas de validación y testing, podremos sacar conclusiones sobre como actua en determinadas situaciones según el tipo de dataset que le estemos ingestando en el entrenamiento y como actua contra las diversidades.  
Existen 2 cuestiones con los algoritmos de aprendizaje supervisado: overfitting (sobreajuste) y underfitting (subajuste) de los datos. El overfitting y el underfitting son los responsables de obtener malos resultados. Cuando entrenamos un modelo intentamos “hacer encajar” (fit en inglés) los datos de entrada entre ellos y con la salida. El fallo común de nuestro modelo generado es de generalizar (“encajar”) el conocimiento que pretendemos que adquieran.  
El underfitting se podría explicar al entrenar un modelo para el reconocimiento de razas de perros, pero entrenarlo con una sola raza de perro. Cuando se intente reconocer una raza nueva, el modelo fallará en el reconocimiento por falta de suficientes muestras; no pudo generalizar el conocimiento.  
Por otra parte, el overfitting se puede ejemplificar siguiendo el ejemplo anterior, al entrenar un modelo para el reconocimiento de razas de perros, pero entrenarlo con 10 razas de perros de color marrón. Cuando se quiera reconocer una raza de perro color blanco, el modelo fallará porque no tiene estrictamente los mismos valores de las muestras de entrenamiento (entrenó a la perfección solamente razas de perros marrones).  
El algoritmo debería ser capaz de generalizar conceptos y que al intentar predecir una nueva muestra con algunos datos desconocidos, sea capaz de dar una clasificación fiable dependiendo del grado de generalización que haya tenido en su entrenamiento.



Deberemos encontrar un punto medio en el aprendizaje de nuestro modelo en el que no estemos incurriendo en underfitting y tampoco en overfitting.  
Veremos como se comporta el algoritmo al inducir ruido en alguna variable y si es capaz de manejar cuestiones como el underfitting y overfitting.

# Dataset

Para las pruebas se utilizará un dataset de la plataforma de crowdfunding (financiamiento colectivo) Kickstarter.  
Kickstarter funciona facilitando la captación de recursos monetarios del público en general, un modelo que evita muchas vías tradicionales de inversión. Los proyectos pertenecen a una categoría determinada y deben cumplir con las directrices de Kickstarter para ponerse en marcha (proyectos de caridad, de causas, de “financiación de vida” y recaudación de fondos sin límites fijos no están permitidos). Los dueños del proyecto eligen una fecha límite y un mínimo objetivo de fondos a recaudar. Si el objetivo monetario elegido no es recolectado en el plazo, el dueño del proyecto no percibe los fondos y el proyecto no se financia.

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo | Valores posibles | Descripción |
| main\_category | categórico | 15 | Categoría del proyecto |
| country | categórico | 20 | País del proyecto |
| days\_funding | numérico entero | 7 a 62 | Plazo de días para llegar al objetivo de dinero a recaudar. (desde la creación del proyecto) |
| year\_launched | categórico | 9 (2009 a 2017) | Año de lanzamiento del proyecto |
| month\_launched | categórico | 12 (January a December) | Mes de lanzamiento del proyecto |
| usd\_goal | numérico entero | 1000 a 50000 | Objetivo monetario a recaudar en dolares |
| funded | binario | yes / no | Valor binario que determina si el proyecto llegó al objetivo a financiar o no |

Preprocesamiento del csv, script.R , etc trtamiento de países, conversión de moendas.

# Procedimiento

Se realizaran pruebas desde R balbalbablalblba