

**UNIVERSIDAD
PANAMERICANA**

APRENDIZAJE DE MÁQUINA

COM194

Enrique González N.

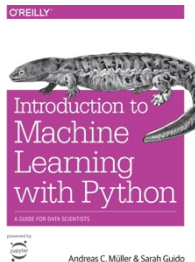
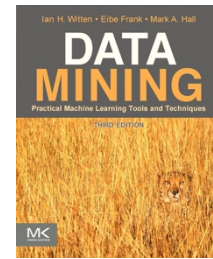
Profesor

COM194-AM-SM1-24



MATERIAL DE CONSULTA

Data Mining
Ian H. Witten, Eibe Frank, Mark A. Hall



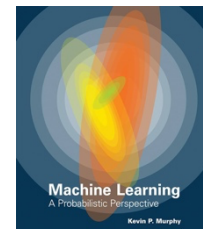
Introduction to Machine Learning with Python
Andreas C. Müller & Sarah Guido

Machine Learning
Tom Mitchell



The Elements of Statistical Learning: Data Mining, Inference, and Prediction
Trevor Hastie, Robert Tibshirani, Jerome Friedman
Springer

Machine Learning: A Probabilistic Perspective
Kevin P. Murphy
The MIT Press



¿Qué es un árbol de decisión?

<https://www.ibm.com/es-es/topics/decision-trees>

¿Qué es el random forest?

<https://www.ibm.com/mx-es/topics/random-forest>





Support Vector Machine

<https://www.ibm.com/docs/es/spss-modeler/saas?topic=mining-oracle-support-vector-machine-svm>

VARIABLE Y FUNCIÓN OBJETIVOS

En machine learning:

- **Variable objetivo:** es la variable que estamos **tratando** de **predecir** o **modelar** con nuestro modelo de ML.
- **Función objetivo (target function):** es la **función** que estamos **tratando de optimizar** durante el entrenamiento del modelo:
 - **Permite modelar** la variable objetivo.

GRAPH

- A graph is a **mathematical structure** used to **model pairwise relations between objects**. It consists of a **collection** of:
 - **Vertices (or Nodes):** Fundamental units representing **entities**.
 - **Edges (or Links):** Connections between vertices, representing a wide variety of structures and relationships between the entities.
- Nodes can be **interconnected in various ways**, including **cycles** and **mutiple paths between nodes**. There is **no strict hierarchy**.
- Graphs can be:
 - **Directed or Undirected:** In **directed grahps**, edges have a direction, indicatig a **one-way relationship**. In **undirected graphs**, edges have no direction, indicating a **two-way relationship**.
 - **Weighted or Unweighted:** In **weighthed graphs**, edges have **weights** representing the **cost of distance between vertices**. In **unweighted graphs**, all edges are considered equal.

ÁRBOLES DE DECISIÓN (DECISION TREES – DT)

- A decision tree (DT) is a **strict hierarchical data structure** that consists of **nodes** **connected by edges**.



- Is a **graphical representation** used for **decision-making** , **predictions** and **data analysis**, by **traversing** from the **root** to a **leaf**.
- Son un **método** de modelado **predictivo** y **visualización** de datos en ML y AI:
 - Se **utilizan** para **clasificación** y **regresión**.
- La idea es crear un **modelo que predice** el valor de una **variable objetivo** mediante la **toma de decisiones simples** basadas en las **características de entrada**.
- Dado un **conjunto de datos** se fabrican **diagramas de construcciones lógicas** que **representan** y **categorizan** una serie de **condiciones** que ocurren de forma **sucesiva**.
- Tienen capacidad de manejar **datos** de **diferentes tipos**: **numéricos** y **categoricos**.
- Por su forma visual son **fáciles** de **entender** e **interpretar**.
- Sin embargo, son **propensos** a **sesgos** y **sobreajuste (overfitting)** cuando el modelo se **adapta demasiado** a los **datos de entrenamiento**.

DT AS A TYPE OF GRAPH

- A decision tree is a specialized type of graph called tree:
 - It has specific properties and constraints.
 - It shares the fundamental characteristics of graphs (nodes and edges to represent relationships), but their structure and purpose are tailored to decision-making and classification tasks.
- Directed Acyclic Graph (DAG): A tree is a directed connected, acyclic graph.
 - It is a collection of nodes connected by edges; it has directed edges (each branch points in a single direction from parent to child).
 - There is exactly one path between any two nodes, implying no cycles; it has no cycles (no path returns to a previously visited node).
- Key differences:
 - Cycles
 - Connectivity
 - Hierarchy
 - Applications



Feature	Graph	Tree
Definition	A collection of nodes (vertices) and edges, where edges connect nodes.	A hierarchical data structure consisting of nodes connected by edges with a single root node.
Structure	Can have cycles (loops) and disconnected components.	No cycles; connected structure with exactly one path between any two nodes.
Root Node	No root node; nodes may have multiple parents or no parents at all.	Has a designated root node that has no parent.
Node Relationship	Relationships between nodes are arbitrary.	Parent-child relationship; each node (except the root) has exactly one parent.
Edges	Each node can have any number of edges.	If there is n nodes then there would be n-1 number of edges
Traversal Complexity	Traversal can be complex due to cycles and disconnected components.	Traversal is straightforward and can be done in linear time.
Application	Used in various scenarios like social networks, maps, network optimization, etc.	Commonly used for hierarchical data representation like file systems, organization charts, HTML DOM, XML documents, etc.
Examples	Social networks, road networks, computer networks.	File systems, family trees, HTML DOM structure.

DIVIDE AND CONQUER

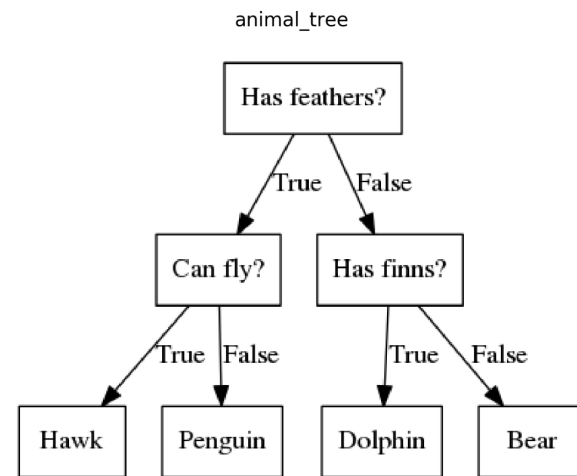
- Operan **dividiendo repetidamente** el conjunto de datos en subconjuntos más pequeños **basados en características específicas**.
- El **proceso** de **división** se **realiza** de manera recursiva mediante la **selección** de la **característica** que mejor separa los datos de acuerdo con **cierto criterio**:
 - La ganancia de información.
 - La reducción de la “impureza” en el conjunto de datos.

REPRESENTACIÓN

- Raíz:
 - A decision tree has **one unique starting point**, called the **root node**.
- Nodos:
 - Describen las **características** o **atributos** del conjunto de datos.



- Representan una pregunta, definen el momento en el que se ha de tomar una decisión de entre varias posibles.
- Each node can have multiple child nodes, but only one parent node.
- A medida que aumenta el número de nodos aumenta el número de posibles finales a los que se puede llegar.
- Ramas:
 - Son las uniones (flechas) entre un nodo y otro.
 - Representan las reglas de decisión basadas en las características.
 - Cada rama representa un posible valor de una característica en un nodo.
 - Representan una acción distinta.
- Hojas:
 - Son los nodos terminales de un árbol.
 - Representan las predicciones de la variable objetivo.



- El proceso de tomar una decisión en un árbol de decisión implica seguir las reglas definidas por las características en cada nodo.
- Al llegar a una hoja, se obtiene una predicción o clasificación para la instancia de entrada basada en las características que se han utilizado para llegar a esa hoja.

CARACTERÍSTICAS

- Profundidad del árbol:
 - Es la longitud máxima desde la raíz hasta la hoja más lejana en el árbol.
 - Una profundidad mayor indica un árbol más complejo y detallado, que puede capturar relaciones más intrincadas en los datos de entrenamiento.
 - Una profundidad excesiva puede conducir a un sobreajuste.



- Número de nodos:
 - Es el total de nodos (internos y hojas) en el árbol.
 - Un mayor número de nodos también sugiere un árbol más complejo y detallado.
 - Un gran número de nodos también puede llevar a un sobreajuste si el árbol no se poda adecuadamente o si se permite crecer demasiado.

Ejemplo

<https://scikit-learn.org/stable/modules/tree.html#regression>

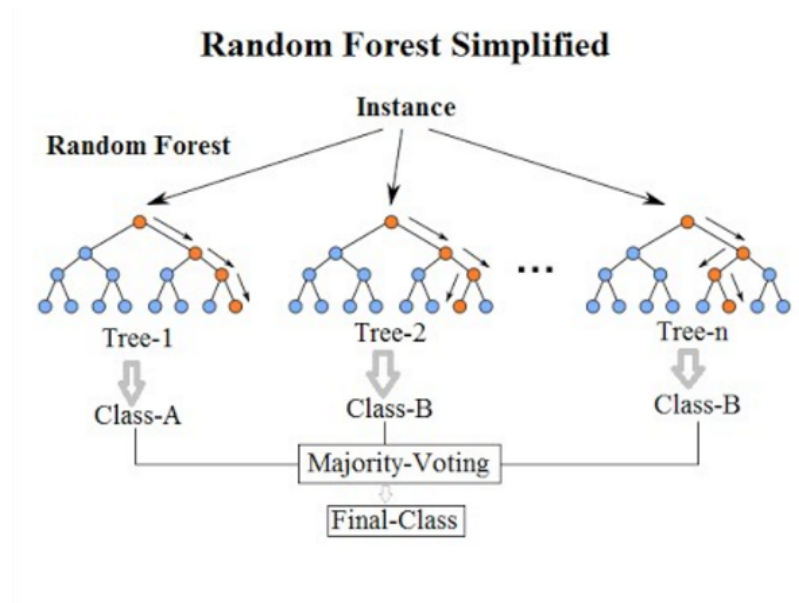
BOSQUE ALEATORIO (RANDOM FOREST - RF)

- El modelo de random forest **se compone** de **varios árboles de decisión**.
- Combina la salida de múltiples árboles de decisión para alcanzar un solo resultado:
 - Cuando varios árboles de decisión forman un conjunto en el algoritmo de random forest, predicen resultados más precisos.
 - Cada árbol se compone (encarga) de una submuestra de los datos extraídos del conjunto de entrenamiento.
 - El resultado final se obtiene:
 - Por voto mayoritario (para clasificación).
 - Por promedio de las predicciones (para regresión).
- Si bien DT es **propenso** a **sesgos** y **sobreajuste**, cuando varios DT forman un conjunto en el algoritmo de RF predicen resultados más precisos.
- Al igual que DT, son un **método** de modelado **predictivo** y **visualización** de datos en ML y AI:
 - Se **utilizan** para **clasificación** y **regresión**.

REPRESENTACIÓN Y CARACTERÍSTICAS

Hiperparámetros principales:

- Profundidad del árbol.
- Número de nodos.
- Cantidad de árboles.



VENTAJAS Y DESVENTAJAS

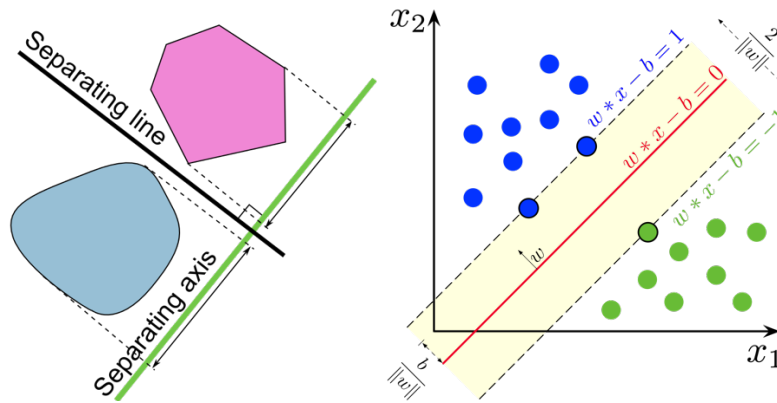
VENTAJAS	DESVENTAJAS
Riesgo reducido de overfitting	Su proceso requiere mucho tiempo
Aporta flexibilidad	Requiere más recursos
Importancia de la característica facil de determinar	Más complejo

SUPPORT VECTOR MACHINES (SVM)

- AKA: Support Vector Networks.
- As a type of supervised learning method, is a powerful tool for both linear and nonlinear classification tasks, as well as for regression tasks.
- Are supervised max-margin (Maximum-margin hyperplane or hard margin) models with associated learning algorithms:
 - In geometry and **optimization** theory, the Hyperplane Separation Theorem is a fundamental concept about disjoint convex sets in n-dimensional Euclidean space.
 - It states that given two non-empty disjoint sets in a vector space, it is always possible to separate them by a hyperplane.
 - This hyperplane acts as a boundary that divides the vector space into two regions, with each set lying entirely on one side of the hyperplane.
 - In the classification task the objective is to maximize the **margin** (the **distance** between the **hyperplane** and the **nearest data points of each class**) while minimizing the errors:



- SVM selects the hyperplane that maximizes the margin.
- Find the hyperplane that creates the **biggest margin** between the training points for the classes (this is an **optimization problem**).



“If both these sets are closed and at least one of them is compact, then there is a hyperplane in between them and even two parallel hyperplanes in between them separated by a gap.”

- In regression tasks, the objective is to find a hyperplane that best fits the data points, for a certain margin of error.
- In classification the margin represents the separation between classes, in regression the margin defines the acceptable deviation from the regression line.
- SVM finds the optimal hyperplane by maximizing the margin while considering misclassifications and using the kernel trick for nonlinear data.

CLASSIFICATION

- SVM aims to separate the data into different classes using a hyperplane.
- Suppose we have N pairs: $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$, with $x_i \in \mathbb{R}$ and $y_i \in \{-1, 1\}$ (two classes like: “red” and “blue”).
- The equation of a hyperplane in N-dimensional space is given by:

$$\omega_1 x_1 + \omega_2 x_2 + \dots + \omega_N x_N + b = 0$$

Where

$\omega_1, \omega_2, \dots, \omega_N$, are the coefficients (weights).
b is the bias term.

- Mathematically, the margin M is given by:

$$M = \frac{1}{\|\omega\|}$$



Where

ω is the coefficient (weight) vector.

- To maximize the margin, we want to find w and b that satisfy

Maximize $\left(\frac{1}{\|\omega\|}\right)$ subject to $y_i(\omega x_i + b) \geq M$ for all data point x_i , $i=1, \dots, N$

ADVANTAGES

SVMs have several advantages:

- Effectiveness in high-dimensional spaces.
- Still effective in cases where the number of dimensions is greater than the number of samples.
- Robustness against overfitting.
- Ability to handle nonlinear data using the kernel trick.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

DISADVANTAGES

- They can be computationally expensive, especially with large datasets.
- They require a careful selection of parameters like the kernel and regularization parameters.
- If the number of features is much greater than the number of samples, avoiding overfitting in choosing Kernel functions and regularization terms is crucial.