

Artificial Intelligence

Lecture 06 – Trees & Ensembles; mini Kaggle Context



Contenido

1. Introducción
2. Decision Trees
3. Bootstrap aggregating
4. Boosting
5. Ejemplos de aplicación
6. Kaggle

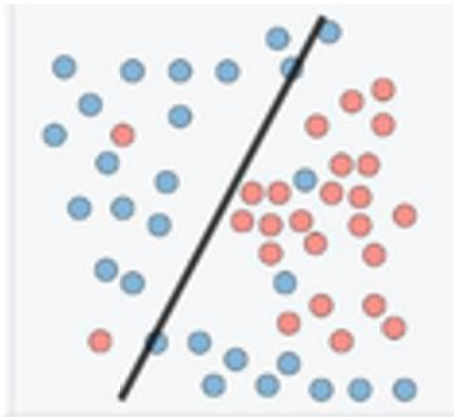


Introducción

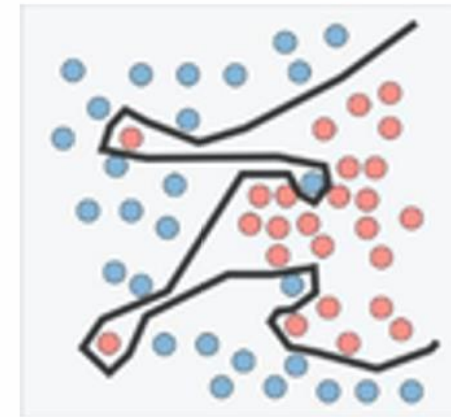


Introducción

Los modelos simples son eficientes y de rápido entrenamiento, pero pueden llegar a tener poco poder explicativo sobre datos observados (subajuste)



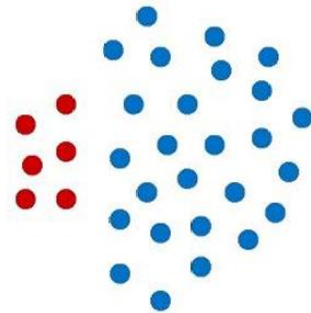
Por otro lado, los modelos muy complejos tienen mucho poder explicativo sobre un conjunto de entrenamiento pero poca habilidad predictiva para datos no observados (sobreajuste).



Introducción

Además, los datos en ocasiones tienen una estructura compleja y, específicamente en problemas de clasificación, pueden presentarse clases no balanceadas.

Imbalanced Class Distribution



Ejemplos:

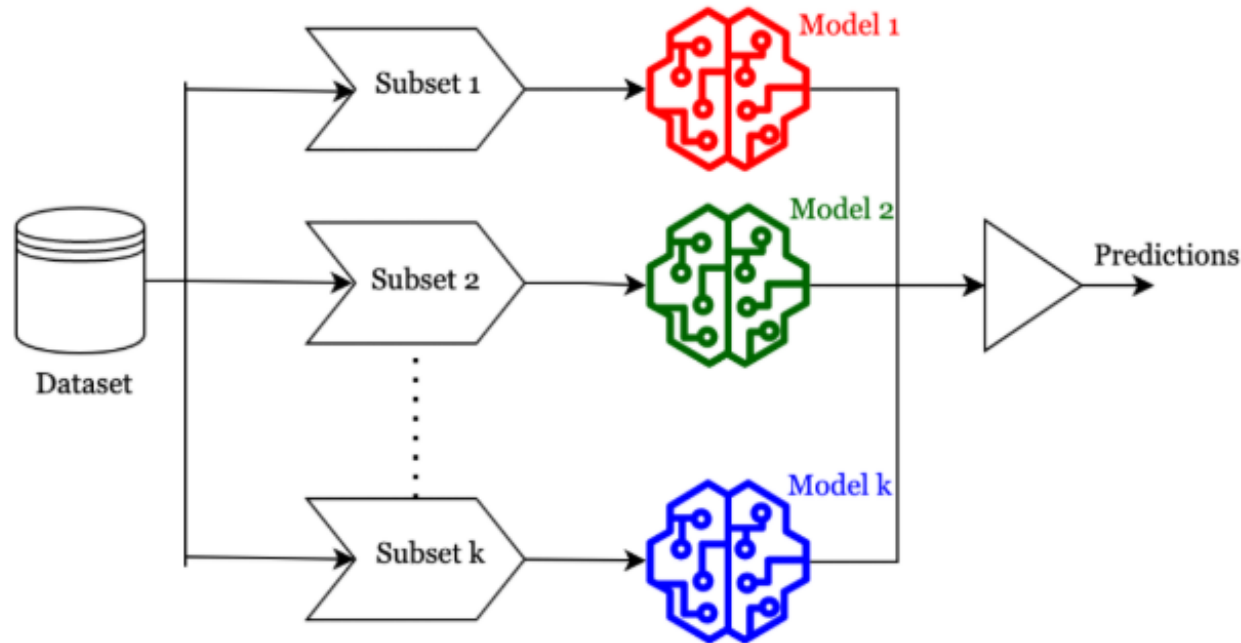
- Diagnósticos médicos
- Detección de fraude
- Mantenimiento predictivo
- Seguridad informática
- Detección de Spam

¿Cual otra se te ocurre?



Introducción

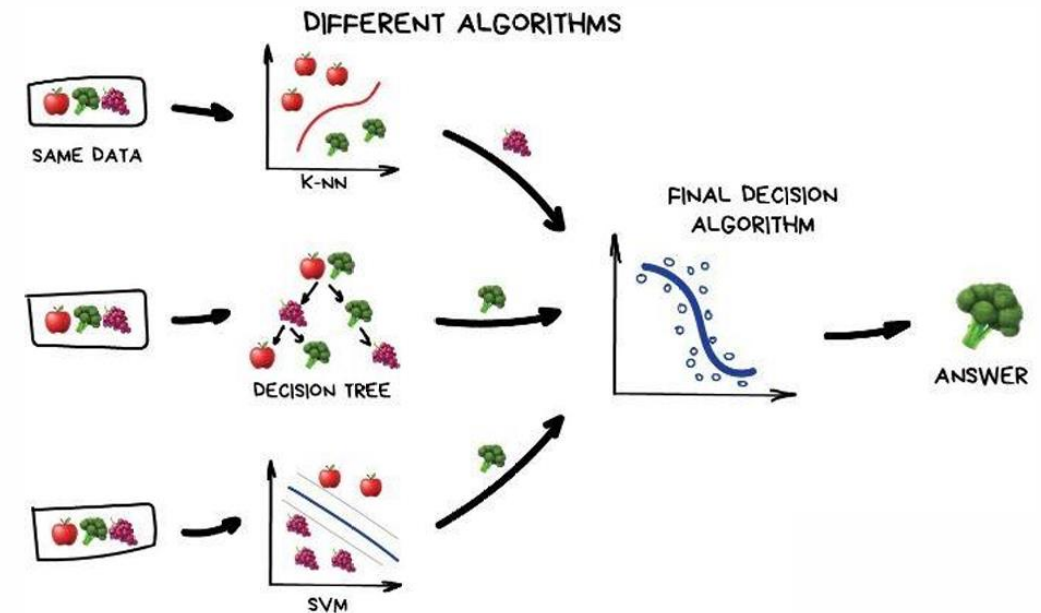
Los ensambles combinan varios modelos de Machine Learning para producir una única predicción y pueden presentar mejor desempeño que los modelos individual



Introducción

La clasificación de los modelos de ensamble se puede abordar desde tres perspectivas:

- Por el tipo de problema que procuran resolver: **subajuste o sobreajuste**.
- Por la manera en la que se entrena el ensamble: **paralelos o secuenciales**.
- Por la variedad de modelos en el ensamble: **homogéneos o heterogéneos**.



“Different ensemble strategies vary on how exactly the base classifiers are trained and how the combination of predictions is achieved.”



Introducción

Empezar con **Decision Trees (Árboles de Decisión)** antes de aprender Ensembles (como Random Forest, Gradient Boosting o XGBoost) es una excelente idea por varias razones fundamentales:

- Los métodos de ensamble más populares (**Bagging** y **Boosting**) se construyen combinando múltiples árboles de decisión.
- Entender cómo funciona un árbol individual te permite: Comprender cómo los ensembles **reducen el overfitting**
- Entender conceptos clave como impureza (Gini/Entropía) que también aplican en ensembles.
- Permiten ver las reglas de decisión en forma de if-else.
- Ayudan a entender cómo se dividen los datos en nodos (útil para debuggear ensembles).

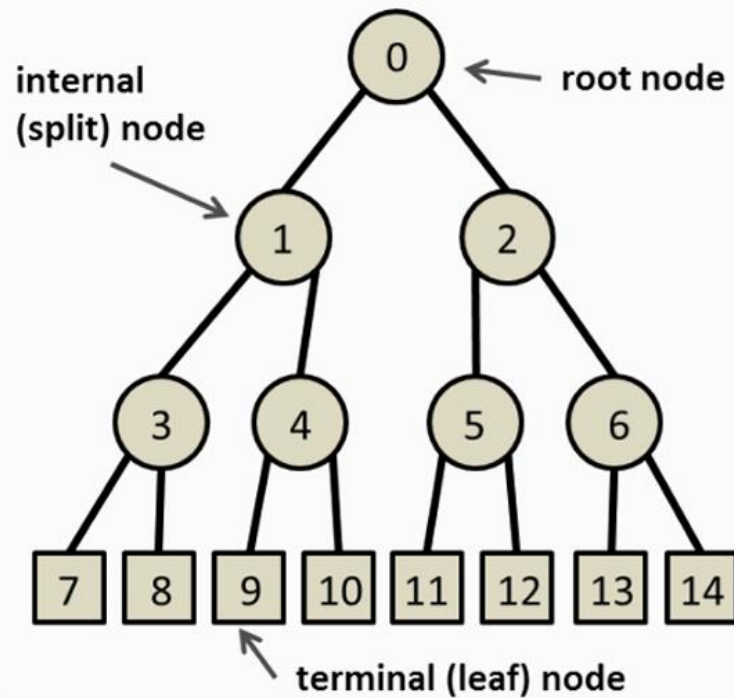


DECISION TREES-ÁRBOLES DE DECISION



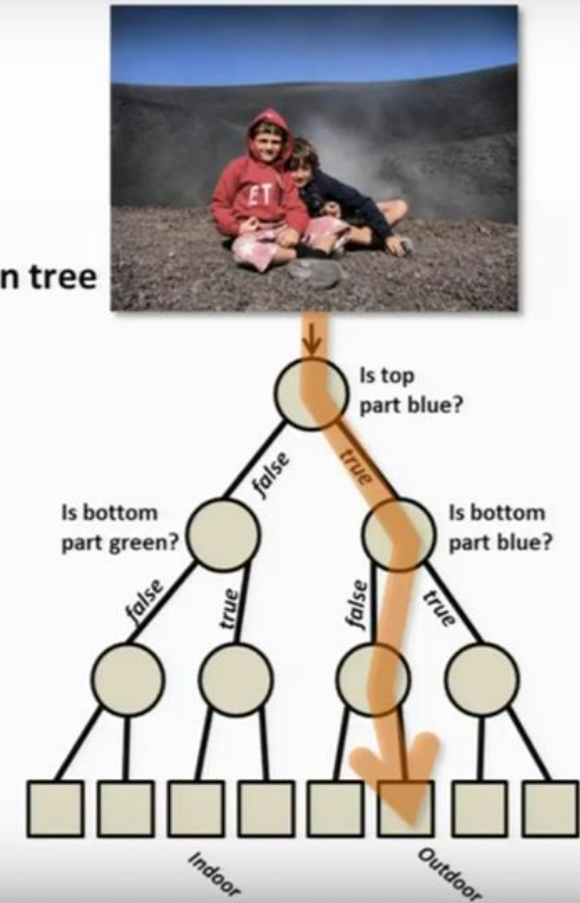
DECISION TREES

A general tree structure



a

A decision tree



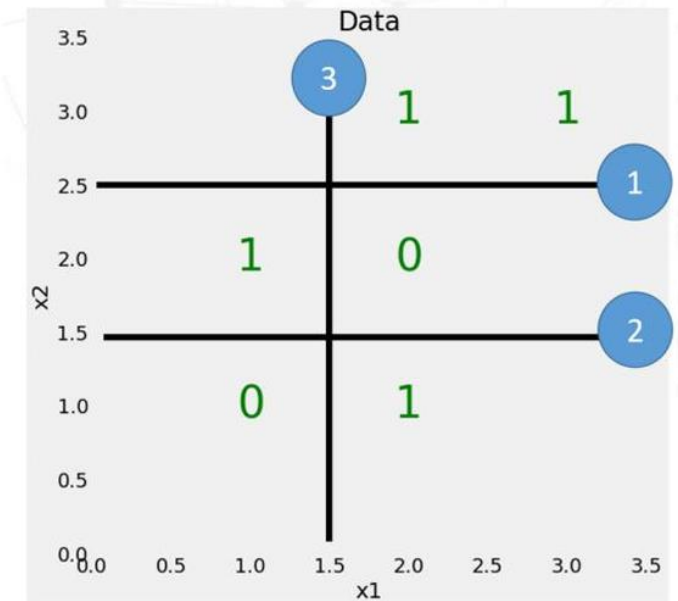
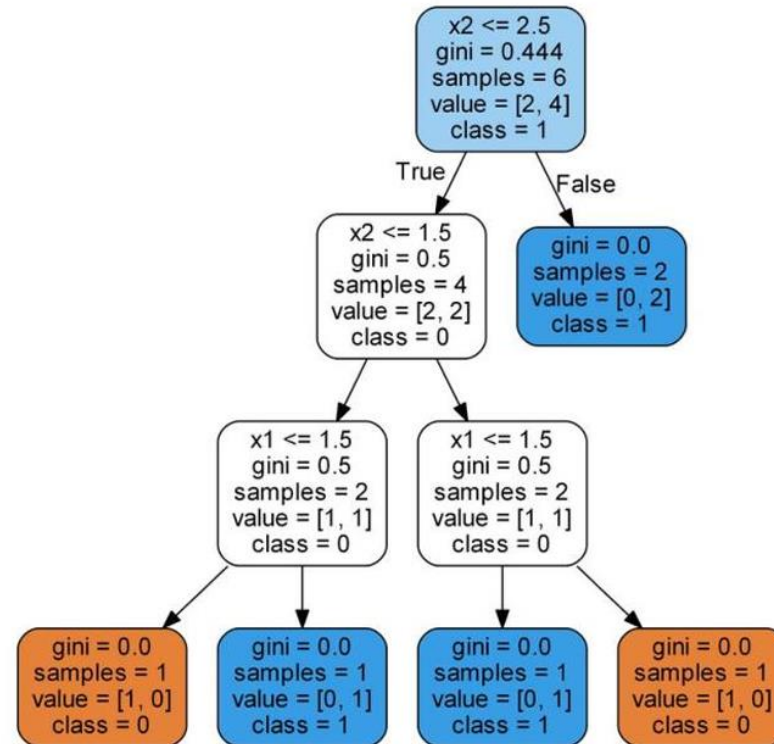
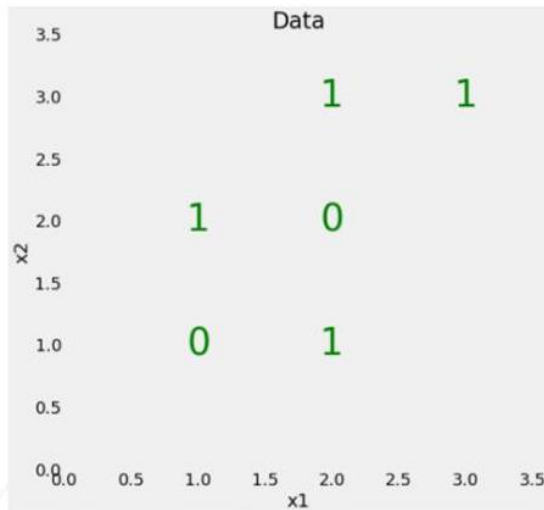
b



DECISION TREES

Como funciona?

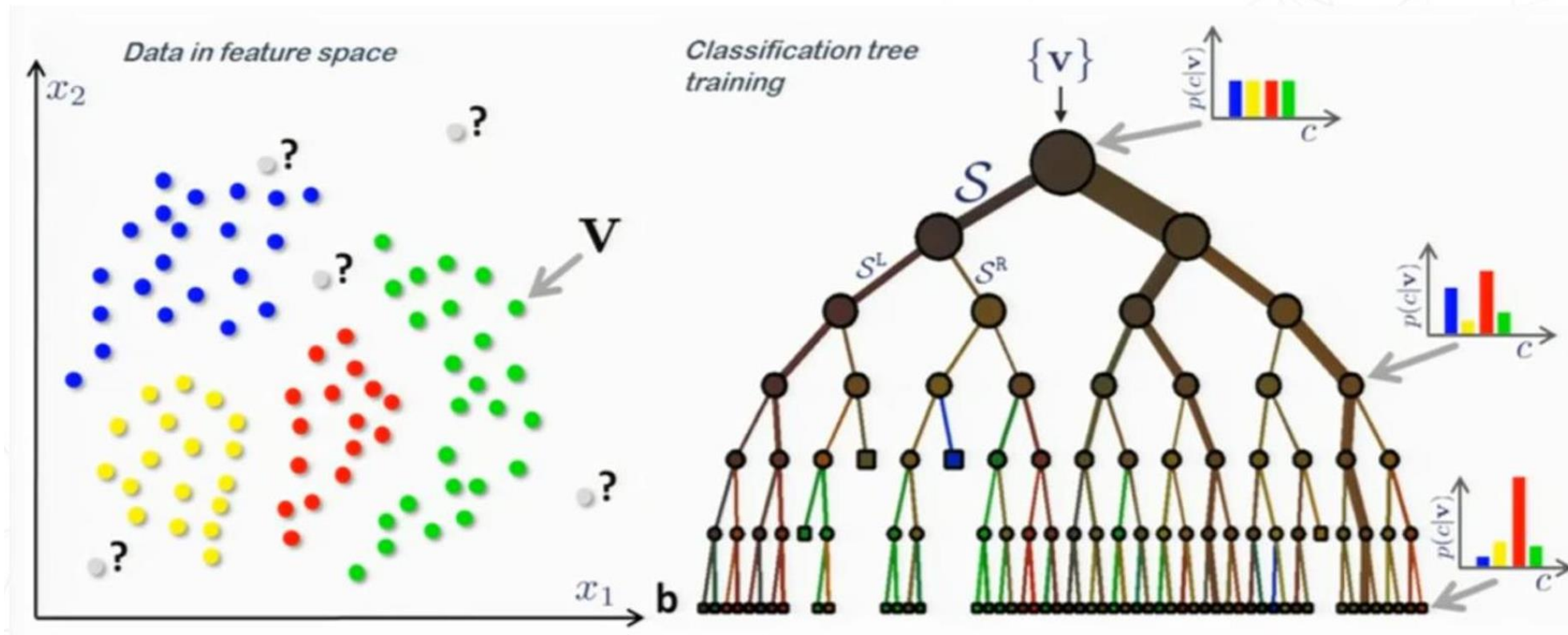
Ejemplo en un problema biclase



DECISION TREES

Como funciona?

Ejemplo en un problema multiclase

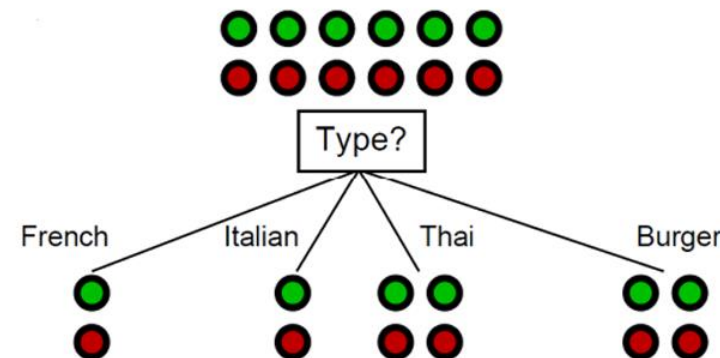
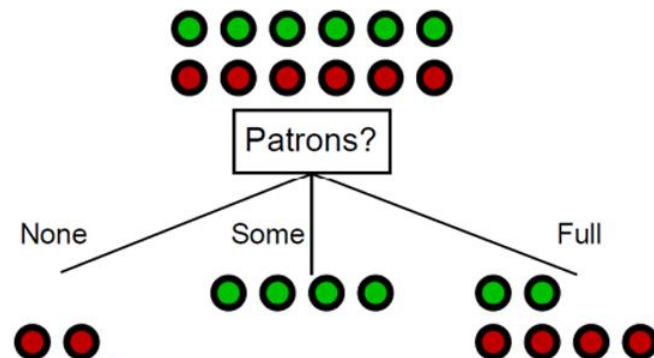


DECISION TREES

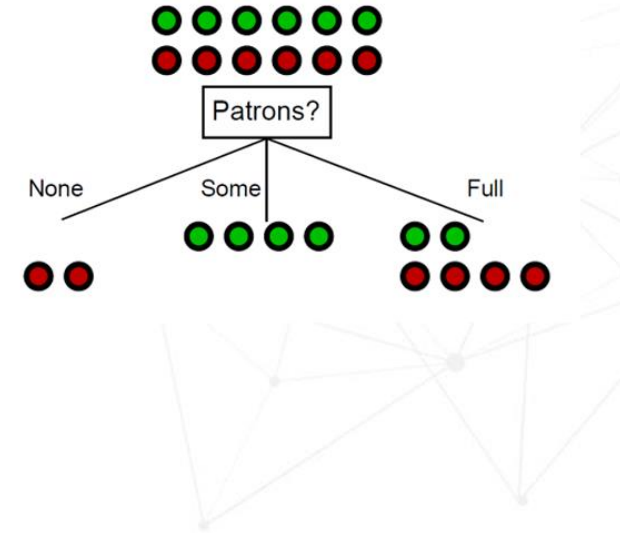
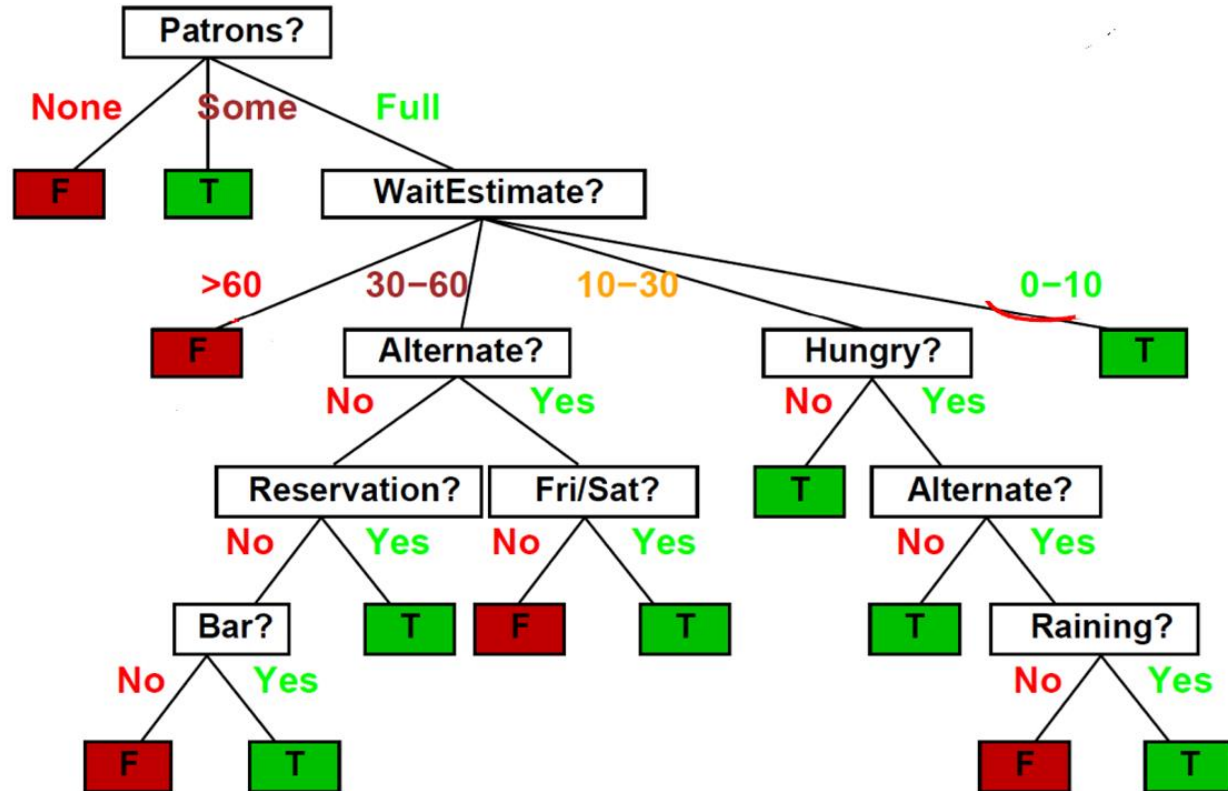
Como tomar Las mejores preguntas para generar la división de datos?

Example	Attributes										Target
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X ₁	T	F	F	T	Some	\$\$\$	F	T	French	0-10	T
X ₂	T	F	F	T	Full	\$	F	F	Thai	30-60	F
X ₃	F	T	F	F	Some	\$	F	F	Burger	0-10	T
X ₄	T	F	T	T	Full	\$	F	F	Thai	10-30	T
X ₅	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X ₆	F	T	F	T	Some	\$\$	T	T	Italian	0-10	T
X ₇	F	T	F	F	None	\$	T	F	Burger	0-10	F
X ₈	F	F	F	T	Some	\$\$	T	T	Thai	0-10	T
X ₉	F	T	T	F	Full	\$	T	F	Burger	>60	F
X ₁₀	T	T	T	T	Full	\$\$\$	F	T	Italian	10-30	F
X ₁₁	F	F	F	F	None	\$	F	F	Thai	0-10	F
X ₁₂	T	T	T	T	Full	\$	F	F	Burger	30-60	T

¿Para ti, cual pregunta es mejor?



DECISION TREES



DECISION TREES

Como tomar las mejores preguntas para generar la división de datos?

ENTROPÍA

$$H(\pi) = - \sum_{\forall \pi} \pi \log_2(\pi)$$

Para un conjunto de entrenamiento con p ejemplos positivos y n ejemplos negativos, tenemos:

$$H(\pi_p, \pi_n) = -\pi_p \log_2 \pi_p - \pi_n \log_2 \pi_n$$

$$\pi_p = \frac{p}{p+n}$$

$$\pi_n = \frac{n}{p+n}$$



DECISION TREES

Como tomar las mejores preguntas para generar la división de datos?

Una variable A, con K valores diferentes, divide el conjunto de entrenamiento E en subconjuntos E1, ..., Ek La Expected Entropy (EH) del atributo A (con ramas i=1,2,...,K) es

$$EH(A) = \sum_{i=1}^K \frac{p_i + n_i}{p + n} H(\pi_{p_i}, \pi_{n_i})$$

Podemos seleccionar una pregunta que minimice EH(A)

La Information Gain del atributo A es:

$$I(A) = H(\pi_p, \pi_n) - EH(A)$$

Ó podemos seleccionar una pregunta que maximice I(A)



DECISION TREES

Class Example

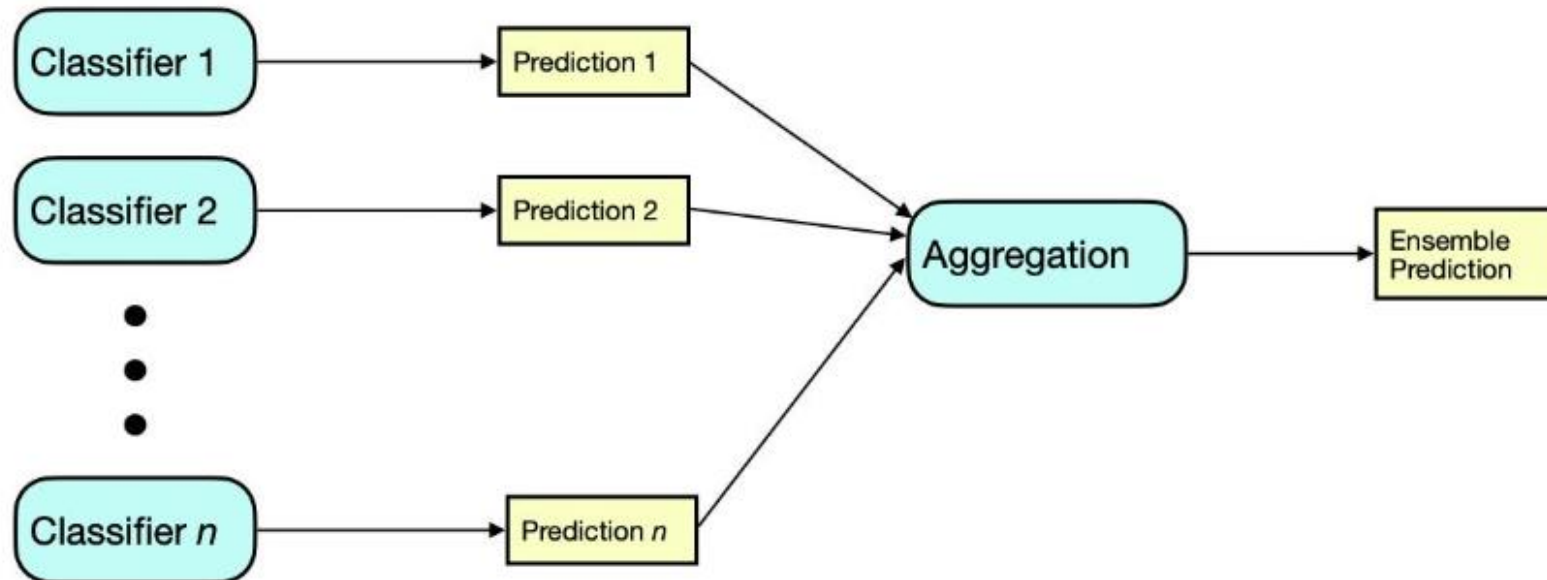


ENSAMBLES: Bootstrap Aggregating



Bootstrap Aggregating

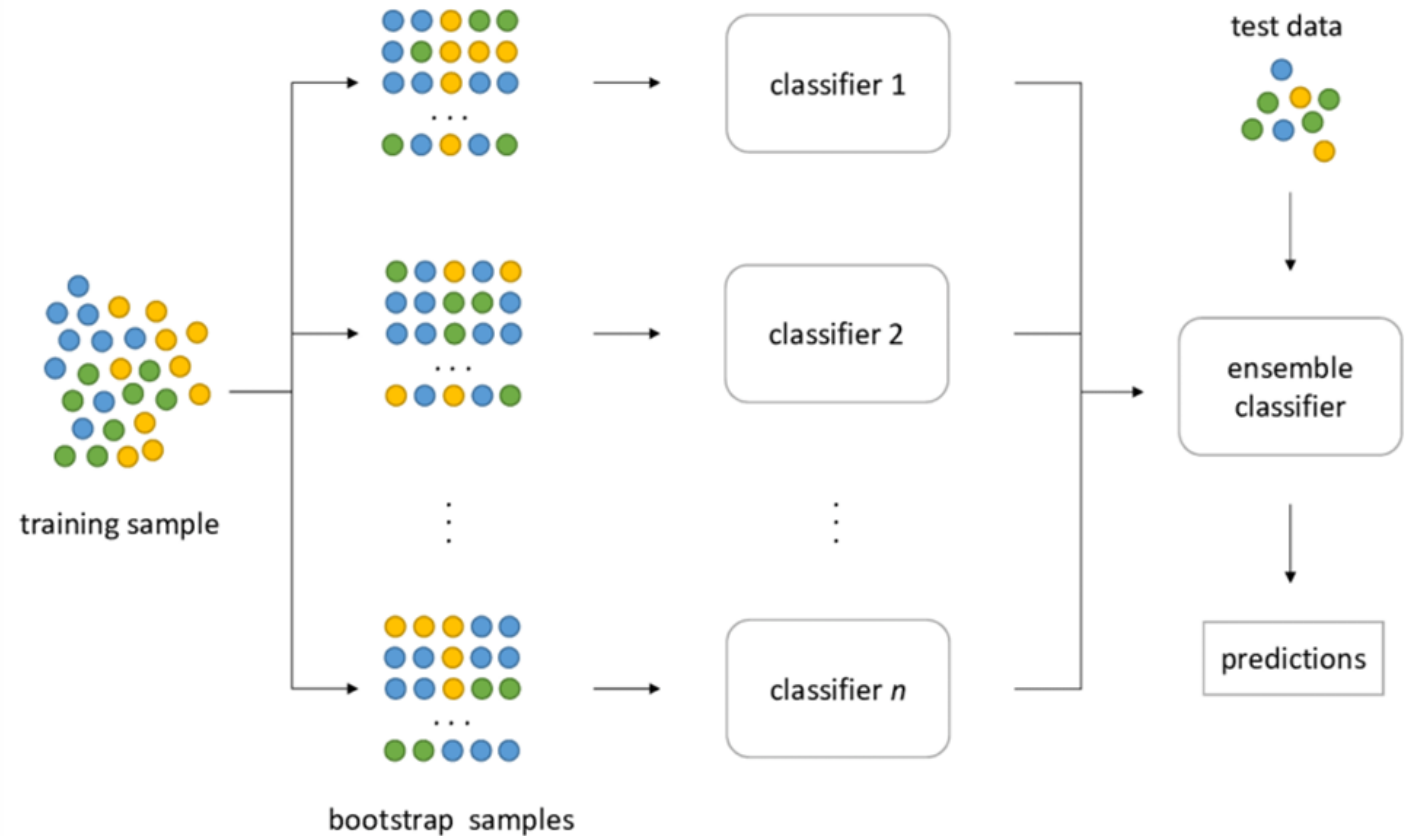
Bagging (bootstrap aggregating) es uno de los métodos más simples de ensamble. La idea es tomar varios clasificadores simples y entrenar cada uno con un subconjunto de los datos.



Bootstrap Aggregating

La **predicción** será:

- El promedio de las predicciones de todos los clasificadores simples en el caso de un problema de **regresión**.
- La clase con el mayor número de votos entre todos los clasificadores en el caso de una **clasificación**



Bootstrap Aggregating

Un ensamble basado en bagging crea los subconjuntos de datos para cada clasificador usando un método conocido como bootstrapping. De manera que el algoritmo se puede resumir en los siguientes pasos:

1. Para cada uno de los modelos simples:

- a. Cree un subconjunto de entrenamiento usando una muestra del conjunto de entrenamiento (tomada aleatoriamente con reemplazo). Puede ser un porcentaje definido.
- b. Entrene el modelo con el subconjunto de datos muestreado.

2. Para realizar inferencia:

Promedie el resultado de todos los modelos si es **regression**.

Haga votación de todos los modelos si es **clasificación** (escoja la moda)



Bootstrap Aggregating

Características:

1. Trata de resolver problemas de sobreajuste.
2. En un ensamble paralelo, es decir que cada modelo es entrenado independiente del otro.
3. Suele ser homogéneo, es decir que se entrena el mismo tipo de clasificadores simples, aunque no hay una razón estricta para no entrenar diferentes.
4. Un ensamble de Árboles de Decisión usualmente se llama **Random Forest**.
5. En bagging también se puede hacer submuestreo de las variables de entrada; así lo hace Random Forest.
6. No funcionan bien con modelos lineales

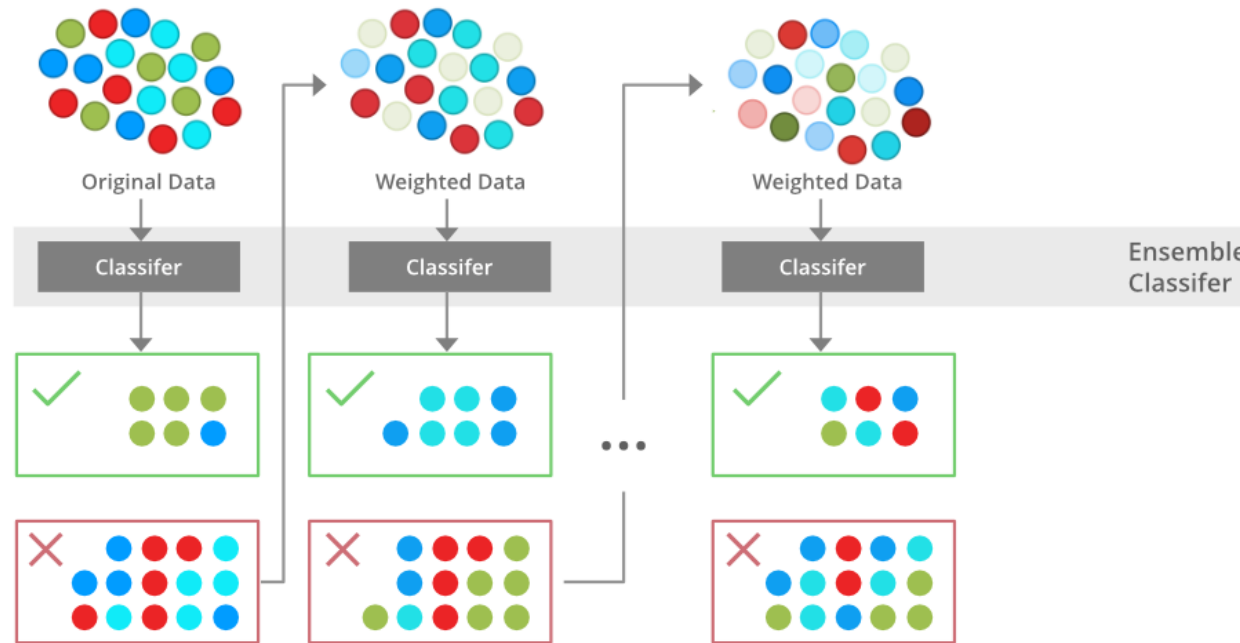


ENSAMBLES: Boosting



Boosting

Boosting engloba a una familia de algoritmos cuya idea general es tomar modelos sencillos (por lo general árboles de decisión) y mejorar sus predicciones de manera secuencial.

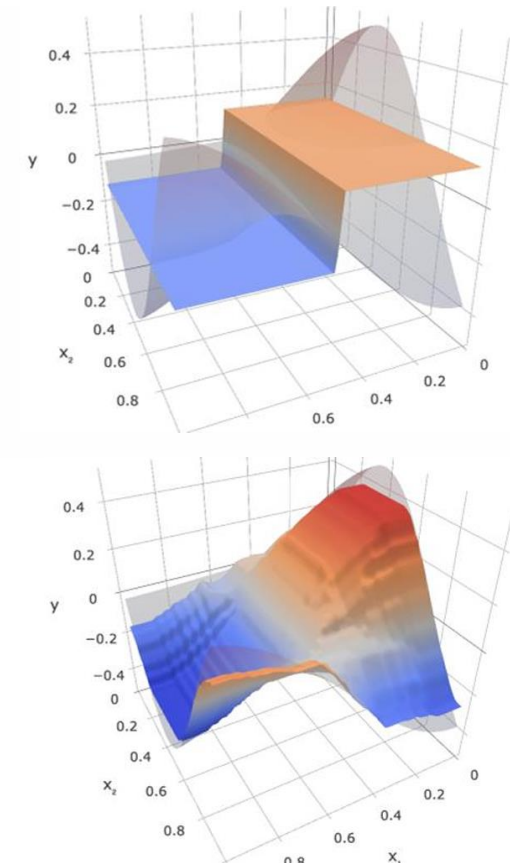
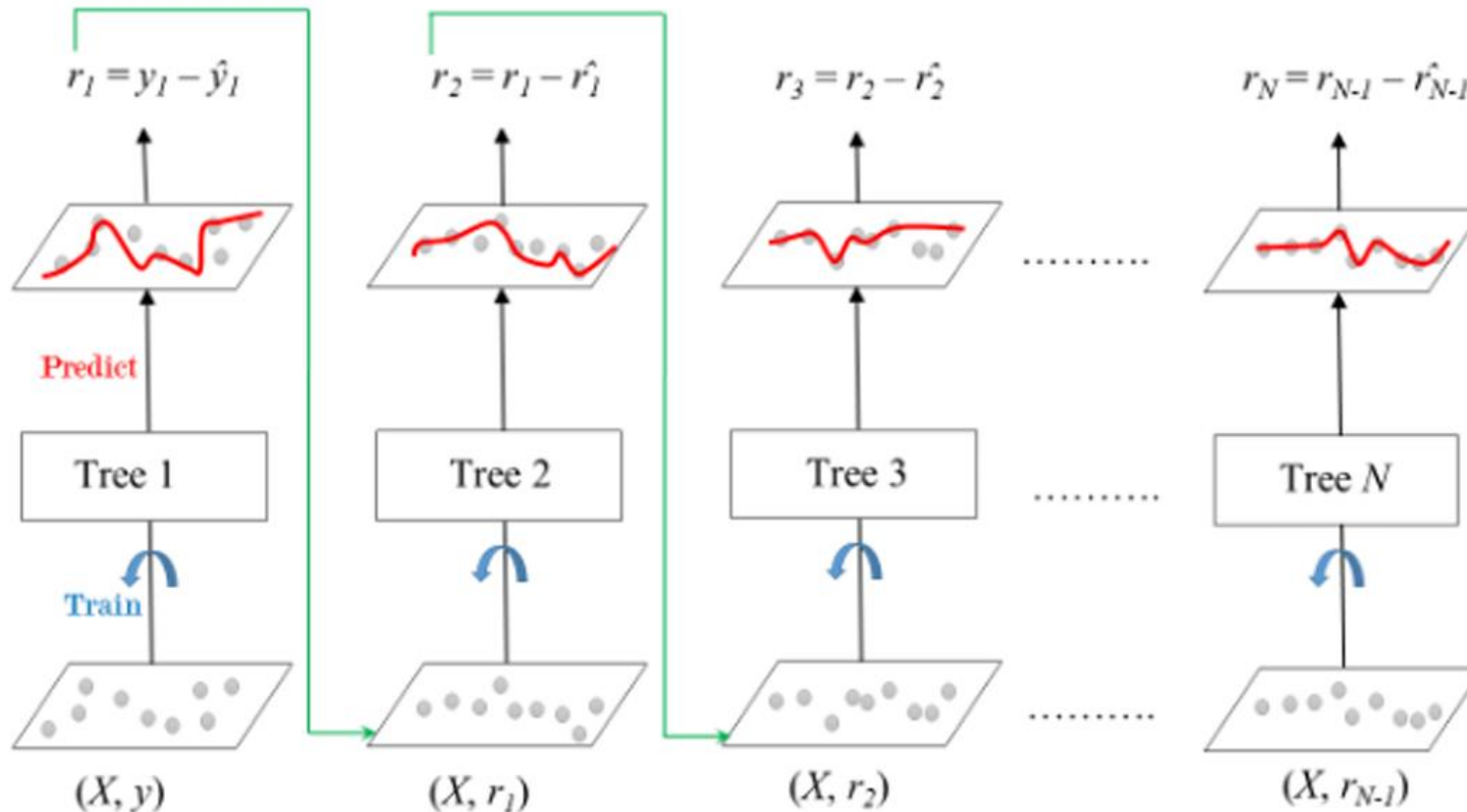


Para mejorar esas predicciones el algoritmo entrena cada modelo secuencialmente con todos los datos y, para cada nuevo modelo, se le da más peso a los datos que no fueron bien clasificados o cuyo error en regresión sea más alto.



Boosting

En lugar de darle más peso directamente a los datos con el mayor error, el gradient boosting entrena el siguiente modelo para que minimice el residual (la diferencia entre las etiquetas y las predicciones del ensamble actual) o para que se ajuste al gradiente de la pérdida



ENSAMBLES

Class Example and Exercise



KAGGLE: Mini-context



KAGGLE: Mini-context

Qué es Kaggle?

Kaggle es una plataforma en línea especializada en ciencia de datos y aprendizaje automático (machine learning), donde usuarios de todos los niveles pueden colaborar, competir y aprender.

1. Acceder a datasets públicos
2. Participar en competencias de ML
3. Explorar notebooks (kernels) de otros usuarios
4. Aprender con cursos y tutoriales
5. Desarrollar y compartir proyectos propios
6. Experimentar con herramientas en la nube (sin instalar nada)
7. Obtener feedback de la comunidad

kaggle

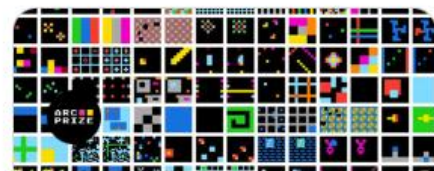


KAGGLE: Mini-context

kaggle

☆ Featured Competitions

Hotness ▾



ARC Prize 2025

Create an AI capable of novel reasoning

Featured · Code Competition

851 Teams

\$1,000,000 ⓘ

3 months to go



Red-Teaming Challenge - OpenAI gpt-oss-20b

Find any flaws and vulnerabilities in gpt-...

Featured · Hackathon

5011 Entrants

\$500,000

9 days to go



Make Data Count - Finding Data References

Identify scientific data use in papers an...

Research · Code Competition

1067 Teams

\$100,000

23 days to go



Jigsaw - Agile Community Rules Classification

Using AI models to help moderators uph...

Featured · Code Competition

900 Teams

\$100,000

2 months to go



MITSUI&CO. Commodity Prediction Challenge

Develop a robust model for accurate an...

Featured · Code Competition

592 Teams

\$100,000

2 months to go



NeurIPS 2025 - Google Code Golf Championship

Implement a variety of programs using t...

Research

412 Teams

\$100,000

2 months to go



BigQuery AI - Building the Future of Data

Build AI solutions with BigQuery

Featured · Hackathon

2526 Entrants

\$100,000

A month to go



MAP - Charting Student Math Misunderstandings

Predict the affinity between misconcepti...

Featured · Code Competition

915 Teams

\$55,000

2 months to go



Referencias

<https://www.geeksforgeeks.org/machine-learning/boosting-in-machine-learning-boosting-and-adaboost/>

<https://www.kaggle.com/>

<https://towardsdatascience.com/ensembles-in-machine-learning-9128215629d1/>

Russell, S., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson.

