


Inteligencia Artificial con Python y scikit-learn

[Install](#) [User Guide](#) [API](#) [Examples](#) [Community](#) [More](#)

scikit-learn

Machine Learning in Python

[Getting Started](#) [Release Highlights for 1.6](#)

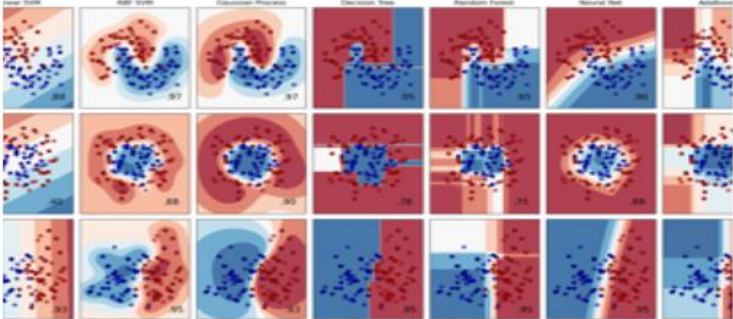
- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: [Gradient boosting](#), [nearest neighbors](#), [random forest](#), [logistic regression](#), and [more...](#)

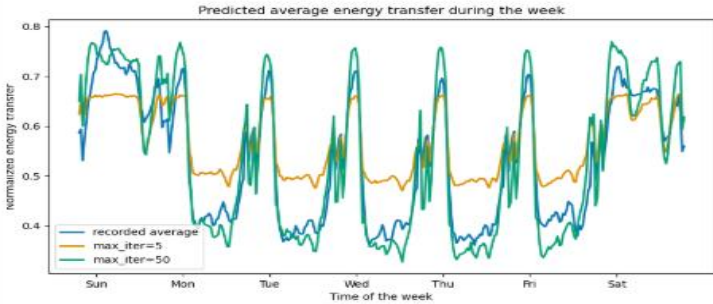


Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, stock prices.

Algorithms: [Gradient boosting](#), [nearest neighbors](#), [random forest](#), [ridge](#), and [more...](#)




Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, grouping experiment outcomes.

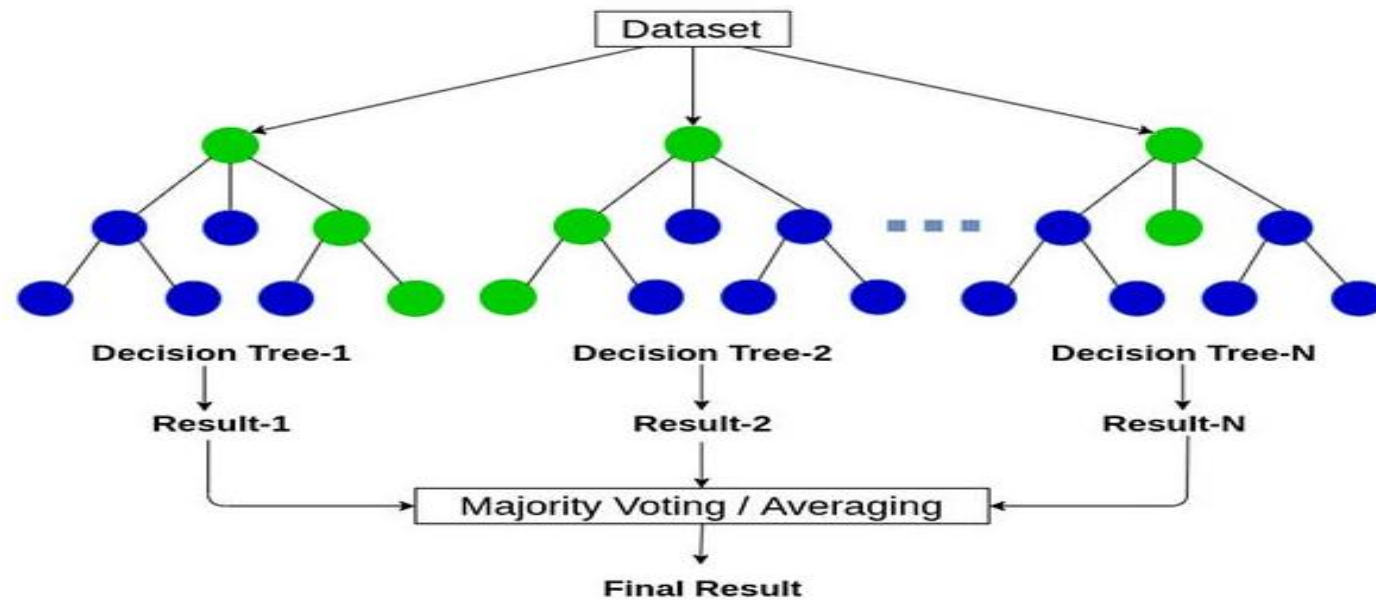
Algorithms: [k-Means](#), [HDBSCAN](#), [hierarchical clustering](#), and [more...](#)



Aprendizaje automático aplicado

Random Forest

Bosques Aleatorios



Random Forest

- El **Random Forest** es un algoritmo de aprendizaje automático basado en un conjunto de árboles de decisión.
- Combina múltiples árboles para mejorar la precisión, reducir el sobreajuste y manejar tanto problemas de clasificación como de regresión.
- Los algoritmos de random forest **tienen tres hiperparámetros principales**, que deben configurarse antes del entrenamiento:
 1. Tamaño del nodo
 2. Cantidad de árboles
 3. Cantidad de características muestreadas
- A partir de ahí, el clasificador de random forest se puede utilizar para solucionar problemas de regresión o clasificación.

Random Forest - Principios básicos

1. Ensemble Learning (aprendizaje en conjunto):

- Random Forest es un método de aprendizaje en conjunto que combina los resultados de múltiples árboles de decisión independientes para generar una predicción robusta.
- Utiliza el principio de agregación: el resultado final es el promedio (para regresión) o el voto mayoritario (para clasificación) de las predicciones de todos los árboles.

2. Diversidad de los árboles:

- Cada árbol se entrena con un subconjunto diferente de los datos y de las características. Esto asegura que los árboles sean diversos y evita la correlación entre ellos.

Random Forest - Principios básicos

3. Bagging (Bootstrap Aggregating):

- Se generan múltiples subconjuntos de datos de entrenamiento mediante muestreo con reemplazo (bootstrap). Cada árbol se entrena con uno de estos subconjuntos, lo que mejora la robustez del modelo.

4. Selección aleatoria de características:

- Para cada división en un árbol, solo se considera un subconjunto aleatorio de las características disponibles. Esto introduce diversidad adicional y reduce la posibilidad de sobreajuste.

Random Forest

Ventajas

1. Robustez al sobreajuste:

- Gracias al bagging y la aleatorización, Random Forest es menos propenso a sobreajustar los datos en comparación con un solo árbol de decisión.

2. Versatilidad:

- Puede manejar datos categóricos y numéricos, además de trabajar con problemas de clasificación y regresión.

3. Manejo de datos faltantes:

- Puede imputar valores faltantes mediante promedios ponderados de los valores predichos por otros árboles.

Random Forest

Desventajas

1. Complejidad computacional:

- Entrenar múltiples árboles y realizar predicciones puede ser computacionalmente costoso, especialmente con conjuntos de datos grandes.

2. Falta de interpretabilidad:

- Aunque proporciona importancia de características, el modelo en sí es un "caja negra" en comparación con los árboles de decisión individuales.

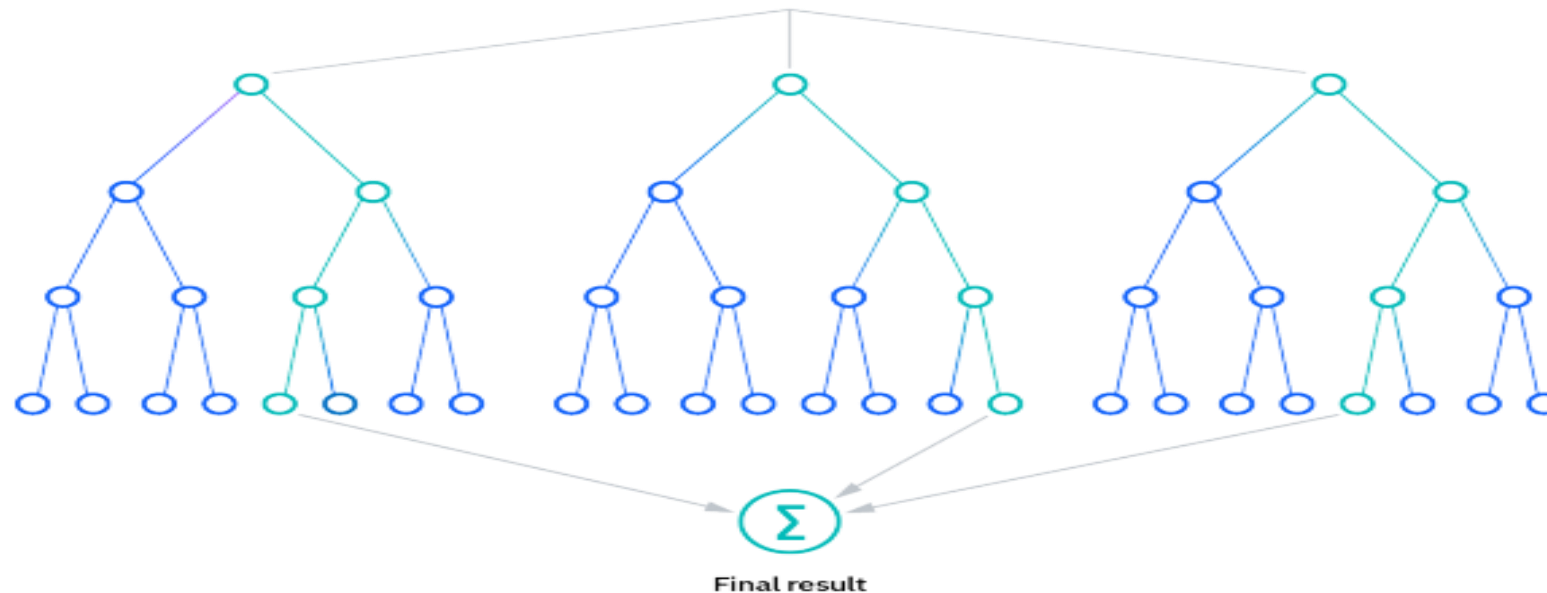
3. Datos desbalanceados:

- Puede tener dificultades con clases desbalanceadas si no se ajustan adecuadamente los parámetros o no se utilizan técnicas específicas.

Random Forest

Dependiendo del tipo de problema, la determinación de la predicción variará.

- Para una tarea de **regresión**, se promediarán los árboles de decisión individuales, y
- Para una tarea de **clasificación**, **un voto mayoritario**, es decir, la variable categórica más frecuente, arrojará la clase predicha.



Referencias

Random Forest

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

Pandas_Cheat_Sheet.

https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf

NearestNeighborsClassification

https://scikit-learn.org/stable/auto_examples/neighbors/plot_classification.html

Confusionmatrix

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html