Introducción a la Ciencia de Datos

Maestría en Ciencias de la Computación

Dr. Irvin Hussein López Nava

Lectura 4: Ensambles

Irvin Hussein Lopez Nava • 16 oct (Última modificación: Ayer)

100 puntos **Fecha de entrega: Mañana, 12:00**

Preparar un reporte de máximo dos cuartillas con un análisis del artículo adjunto. El reporte debe finalizar con una crítica al artículo, el cual será abordado en las siguientes sesiones.

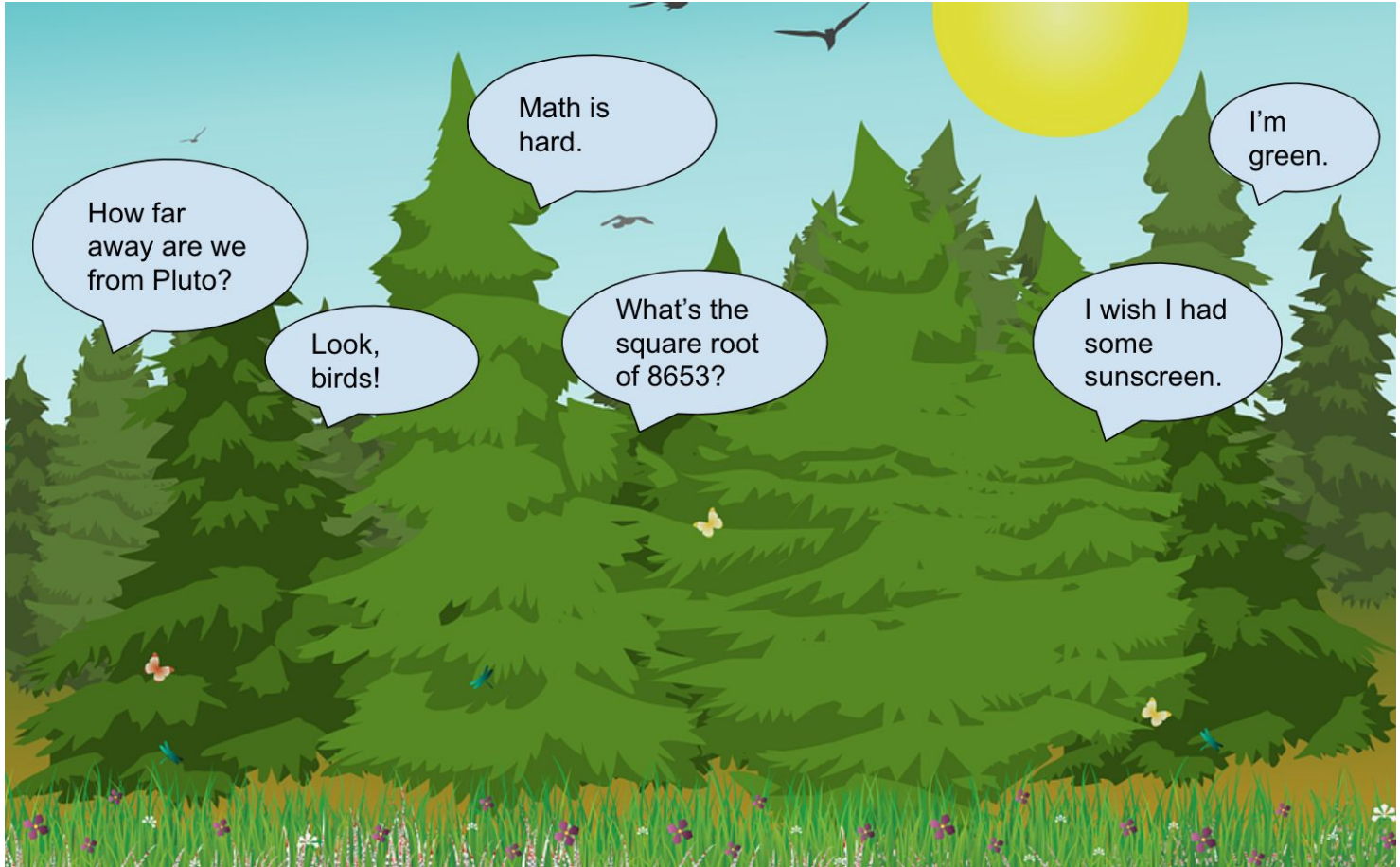
Ma, Y., Zhao, S., Wang, W., Li, Y., & King, I. (2022). Multimodality in meta-learning: A comprehensive survey. Knowledge-Based Systems, 250, 108976.

1-s2.0-S0950705122004737...
PDF

05

Meta-aprendizaje

Just a random forest...



5.3 Bagging

Centrarse en los bootstraps

- En los **métodos paralelos**, los diferentes modelos débiles se ajustan de forma independiente entre sí y es posible entrenarlos simultáneamente.
- El enfoque más famoso es el *bagging (bootstrap aggregating)*, o también conocido como "agregación de arranque".
- El *bagging* produce un modelo ensamble que es más robusto que los modelos individuales que lo componen.

Bootstrapping

- Esta técnica estadística consiste en generar muestras de tamaño *B* (denominadas muestras *bootstrap*) a partir de un conjunto de datos inicial de tamaño *N* extrayendo aleatoriamente B observaciones con reemplazo.

Bootstrapping

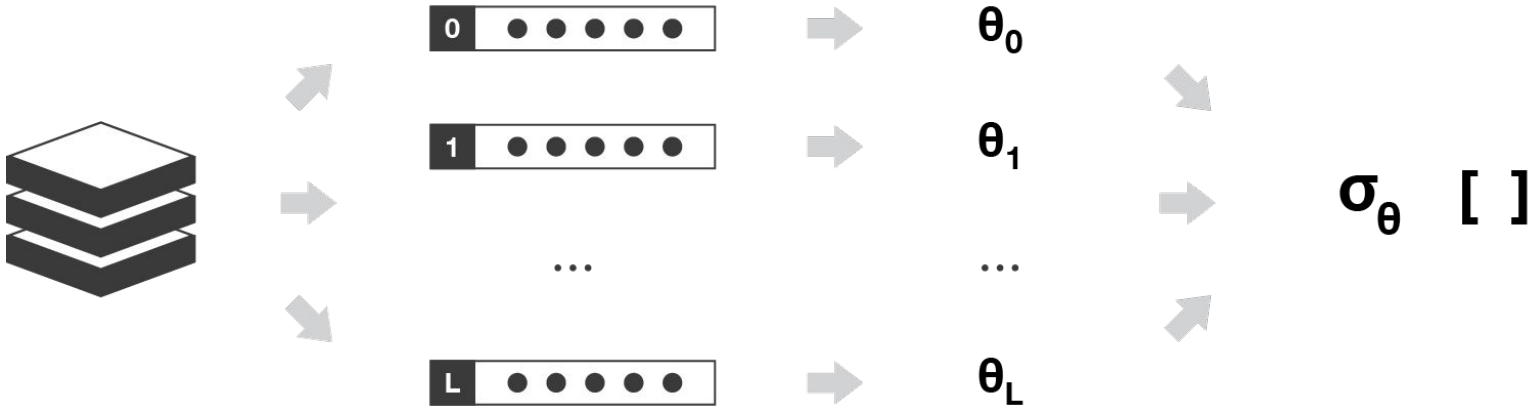
- Bajo ciertas suposiciones, estas **muestras** tienen buenas propiedades estadísticas: pueden considerarse muestras representativas e independientes de la distribución real de los datos.
- Las hipótesis a verificar para que esta aproximación sea válida son dos
  - **Representatividad**: *N* debe ser lo suficientemente grande como para capturar la mayor parte de la complejidad de la distribución subyacente.
  - **Independencia:** *N* debe ser lo suficientemente grande en comparación con *B* para que las muestras no estén demasiado correlacionadas.

Bootstrap samples

- Las muestras *bootstrap* se utilizan a menudo para evaluar la varianza o los intervalos de confianza de los estimadores estadísticos.
- Por definición, un **estimador estadístico** es una función de algunas observaciones, i.e., una variable aleatoria con varianza procedente de estas observaciones.
- Para estimar la varianza de un estimador de este tipo, se evalúan varias muestras independientes extraídas de la distribución de interés.

Bootstrap samples

- En la mayoría de los casos, considerar **muestras verdaderamente independientes** requeriría demasiados datos en comparación con la cantidad realmente disponible.
- El *bootstrapping* se utiliza para generar varias muestras *bootstrap* que pueden considerarse "casi-representativas" y "casi-independientes".
- Estas muestras *bootstrap* **aproximan** la varianza del estimador, evaluando su valor para cada una de ellas.



initial dataset

L bootstrap samples

estimator of interest evaluated for each bootstrap sample

variance and confidence intervals computed based on the L realisations of the estimator

Bagging

- Cuando se entrena un modelo, ya sea para problemas de clasificación o de regresión, una función (modelo) devuelve un resultado que se define con respecto al conjunto de datos de entrenamiento.
  - Debido a la **varianza teórica** del conjunto de datos de entrenamiento, el modelo ajustado también está sujeto a la **variabilidad**: si se observa otro conjunto de datos, se construye un modelo diferente.
- La idea del *bagging* es sencilla: ajustar varios modelos independientes y "promediar" sus predicciones para obtener un modelo con una varianza menor.
  - Sin embargo, en la práctica, ajustar modelos totalmente independientes requiere demasiados datos.

Bagging

- Por tanto, debemos confiar en las buenas "propiedades aproximadas" de las muestras *bootstrap* (representatividad e independencia) para ajustar modelos que sean casi independientes.
    - Crear múltiples muestras *bootstrap* para que cada una de ellas actúe como otro conjunto de datos (casi) independiente extraído de la distribución real.
- Ajustar un aprendiz débil para cada una de estas muestras y, finalmente, agregarlas para obtener un modelo de ensamble con menos varianza que sus componentes.
    - El promediado de los resultados de los aprendices débiles no cambia la respuesta esperada, pero reduce su varianza (las variables aleatorias conservan el valor esperado, pero reducen la varianza).

Bagging

- Sean *L* muestras bootstrap (aproximaciones de *L* conjuntos de datos independientes) de tamaño B denotadas por

$$\{z_1^1, z_2^1, ..., z_B^1\}, \{z_1^2, z_2^2, ..., z_B^2\}, ..., \{z_1^L, z_2^L, ..., z_B^L\}$$

$$z_b^l \equiv b\text{-th observation of the } l\text{-th bootstrap sample}$$

ajustar L aprendices débiles casi independientes (uno en cada conjunto de datos)

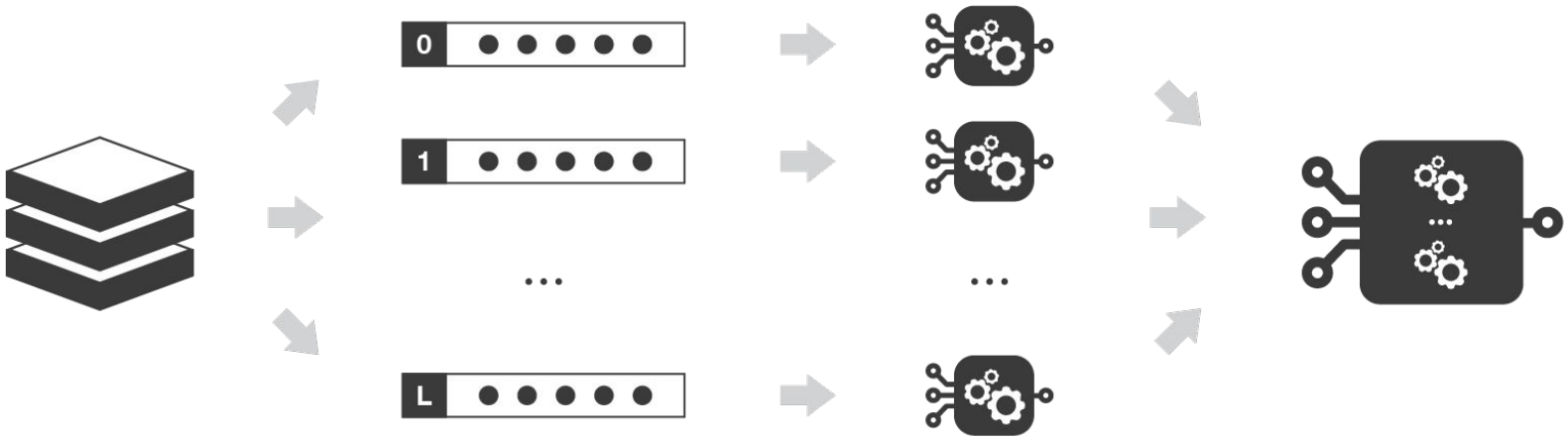
$$w_1(x), w_2(x), \cdots, w_L(x)$$

y agregarlos en algún tipo de proceso de promediado para obtener un modelo de conjunto con una varianza menor.

$$s_L = arg\ \max_k [f(l|w_l(x) = k)]$$

Bagging

- Hay varias formas de agregar múltiples modelos ajustados en paralelo.
  - Para **regresión**, los resultados de los modelos individuales pueden promediarse, literalmente, para obtener el resultado del modelo de conjunto.
  - Para **clasificación**, la salida de cada modelo es un voto y la clase que recibe la mayoría de los votos es devuelta por el modelo de conjunto: *hard-voting*.
  - Si se tienen en cuenta las probabilidades de cada clase obtenidas por todos los modelos, se calcula la media de estas probabilidades y se mantiene la clase con la probabilidad media más alta: *soft-voting*.



initial dataset

L bootstrap samples

weak learners fitted on each bootstrap sample

ensemble model (kind of average of the weak learners)

Random Forest

- Los árboles de decisión son modelos base muy populares para los métodos de ensamble.
- Los ‘aprendices fuertes’ compuestos por varios árboles se denominan bosques (***forests***).
- Los árboles que componen un bosque pueden ser superficiales o *shallow* (poca profundidad), o profundos (mucha profundidad).
    - Los árboles poco profundos tienen menos varianza pero mayor sesgo, por lo que son la mejor opción para los **métodos secuenciales**.
    - Los árboles profundos tienen un sesgo bajo, pero una varianza alta, por lo que son una opción adecuada para *bagging*.

Random Forest

- El enfoque de ‘bosques aleatorios’ es un método basado en *bagging* en el que se combinan árboles profundos, ajustados en muestras *bootstrap*, para producir un resultado con menor varianza.
- *Random forest* también utiliza otro truco para hacer que los múltiples árboles ajustados estén un poco menos correlacionados entre sí:
    - al hacer crecer cada árbol, en lugar de muestrear sólo sobre las observaciones para generar un *bootstrap*, **muestrea sobre las características** y utiliza sólo un subconjunto aleatorio de ellas para construir el árbol.

Random Forest

- El **muestreo sobre características** hace que todos los árboles no tengan exactamente la misma información y reduce la correlación entre los distintos resultados devueltos.
- Otra ventaja es que hace que el proceso de toma de decisiones sea más robusto frente a los datos que faltan:
    - Las observaciones (del conjunto de datos de entrenamiento o no) en las que faltan datos pueden clasificarse basándose en los árboles que utilizan características en las que no faltan datos.
- *Random forest* combina los conceptos de *bagging* y selección aleatoria de subespacios de características para crear modelos más robustos.

initial dataset

bootstrap samples + selected features

deep trees fitted on each bootstrap sample and considering only selected features

random forest (kind of average of the trees)

Random Forest algorithm

Algorithm 1: Pseudo code for the random forest algorithm

To generate c classifiers:
for i = 1 to c **do**
  Randomly sample the training data D with replacement to produce D_i
  Create a root node, N_i containing D_i
  Call BuildTree(N_i)
end for

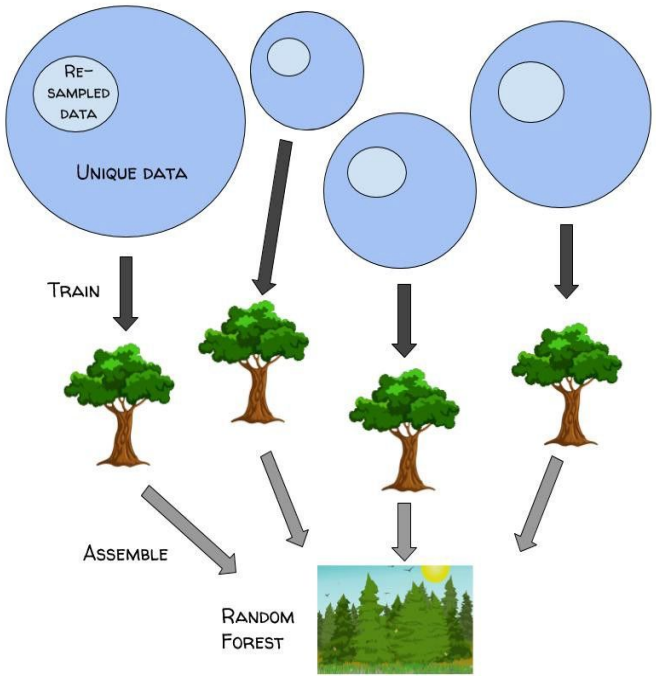
BuildTree(N):
if N contains instances of only one class **then**
  return
else
  Randomly select x% of the possible splitting features in N
  Select the feature F with the highest information gain to split on
  Create f child nodes of N, N_1,..., N_f, where F has f possible values (F_1, … , F_f)
  for i = 1 to f **do**
    Set the contents of N_i to D_i, where D_i is all instances in N that match
    F_i
    Call BuildTree(N_i)
  end for
end if

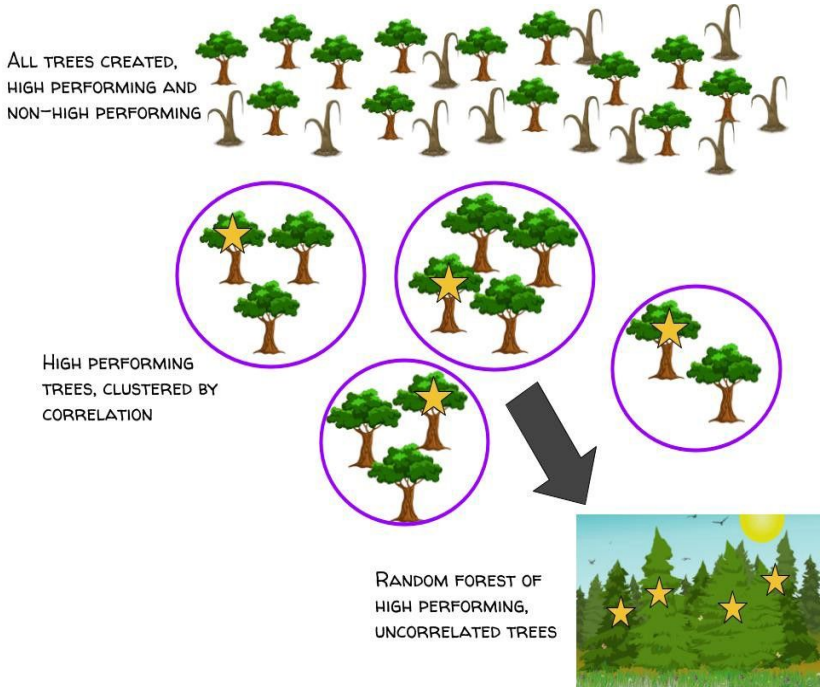
Variantes de Random Forest

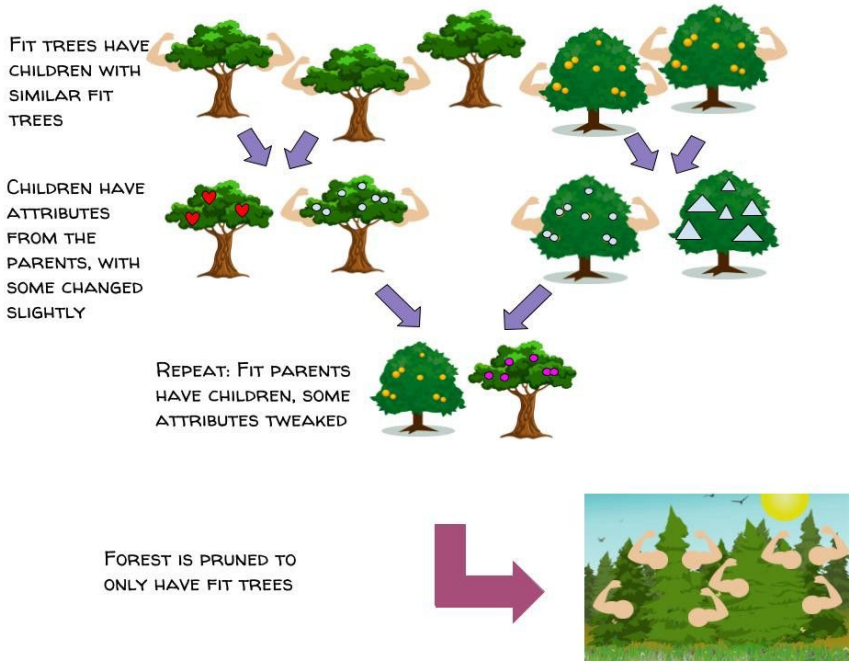
- *Kernel Random Forest* establece la conexión entre los bosques aleatorios y los métodos kernel:
  - El **bosque uniforme** selecciona uniformemente una característica entre todas las características y realiza divisiones en un punto trazado uniformemente en el lado de la celda, a lo largo de la característica preseleccionada.
  - El KeRF de nivel k es el mismo que para el bosque centrado o uniforme, salvo que las predicciones se realizan mediante la función kernel correspondiente.

Resumen de Bagging

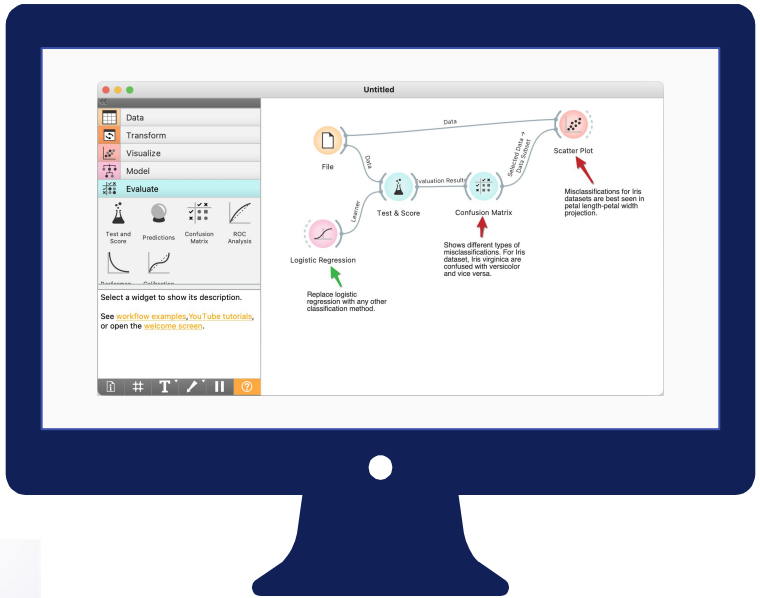
- En los métodos de *bagging*, varias instancias del mismo aprendiz débil se entrenan en paralelo en diferentes *bootstraps* y luego se agregan promediando o votando.
- Los bosques aleatorios difieren sólo en una cosa del *bagging* puro:
  - Utilizan un algoritmo de aprendizaje de árboles modificado que selecciona, en cada *bootstrap*, un subconjunto aleatorio de las características.
  - Este proceso se denomina '*feature bagging*'.
- Los modelos base con un sesgo bajo pero una varianza alta se adaptan bien al *bagging*.


Re-
sampled
data
Unique data
Train
Assemble
Random
Forest

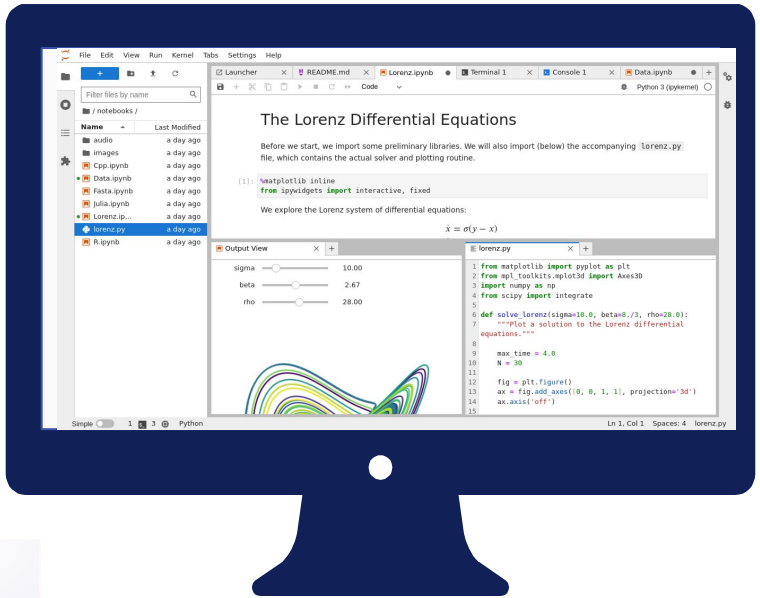

All trees created,
high performing and
non-high performing
High performing
trees, clustered by
correlation
Random forest of
high performing,
uncorrelated trees


Fit trees have children with similar fit trees
Children have attributes from the parents, with some changed slightly
Repeat: Fit parents have children, some attributes tweaked
Forest is pruned to only have fit trees

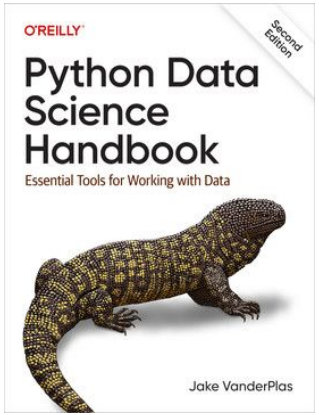
(Go to live)


Untitled
Data
Transform
Visualize
Model
Evaluate
Test and Score
Predictions
Confusion Matrix
ROC Analysis
Select a widget to show its description.
See workflow examples, YouTube tutorials, or open the welcome screen.
File
Test & Score
Confusion Matrix
Scatter Plot
Logistic Regression
Data
Learner
Evaluation Results
Selected Data
Data Subset
Misclassifications for Iris datasets are best seen in petal length-petal width projection.
Shows different types of misclassifications. For Iris dataset, Iris virginica are confused with Versicolor and vice versa.
Replace logistic regression with any other classification method.

(Go to live notebook)



Extra Libro



GitHub

- 05.08-Random-Forests.ipynb

Gracias!

¿Alguna pregunta?

hussein@cicese.mx

https://sites.google.com/view/husseinlopeznava

SCAN ME

CREDITS: This presentation was based on a template by **Slidesgo**, and includes icons by **Flaticon**.