

# Modelos Keras y Scikit-Learn para Deep Learning

---

Juan Gamarra Moreno

# Introducción

---

- La biblioteca scikit-learn es la biblioteca más popular para el aprendizaje automático general en Python. Sin embargo se pueden usar modelos de aprendizaje profundo de Keras con la biblioteca scikit-learn en Python



# Introducción

---

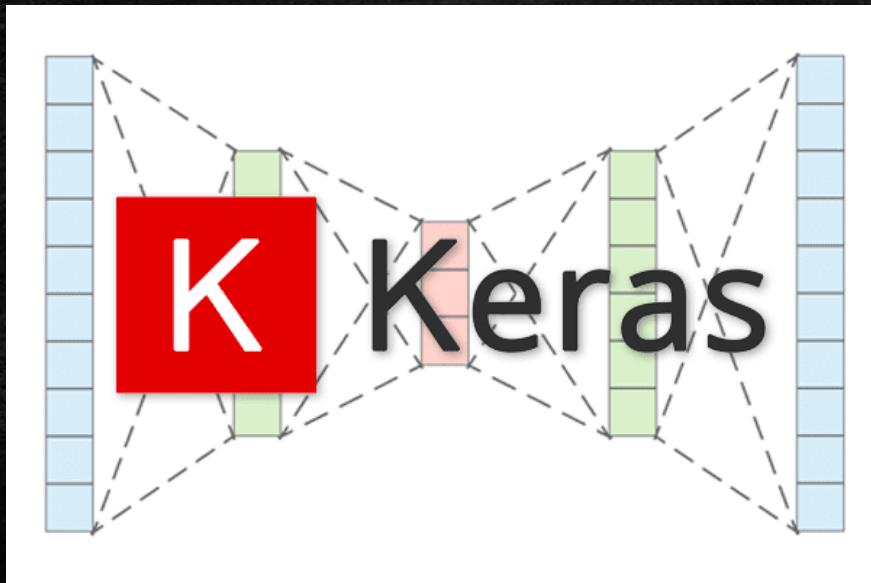
Podemos realizar lo siguiente:

- Envolver un modelo de Keras para usarlo con la biblioteca de aprendizaje automático scikit-learn.
- Evaluar fácilmente los modelos de Keras mediante la validación cruzada en scikit-learn.
- Ajustar los hiperparámetros del modelo de Keras usando la búsqueda de cuadricula (grid search) en scikit-learn.

# Keras

---

Keras es una biblioteca popular para el aprendizaje profundo en Python, no todo el aprendizaje automático, centrándose solo en lo que necesita para definir y construir modelos de aprendizaje profundo de forma rápida y sencilla.



# Scikit-learn

---

La biblioteca scikit-learn en Python se basa en SciPy para un cálculo numérico eficiente. Es una biblioteca con todas las funciones para el aprendizaje automático de uso general y proporciona muchas utilidades que son útiles en el desarrollo de modelos de aprendizaje profundo. Por ejemplo:

- Evaluación de modelos utilizando métodos de remuestreo como la validación cruzada de k-fold.
- Búsqueda y evaluación eficiente de hiperparámetros del modelo.



# Evaluar modelos con validación cruzada

---

- Las clases KerasClassifier y KerasRegressor en Keras toman un argumento build\_fn que es el nombre de la función a llamar para crear su modelo. Debe definir una función llamada lo que quiera que defina su modelo, lo compile y lo devuelva.
- Cuando se ajusta el modelo de Keras, la estimación de la precisión del modelo se puede simplificar en gran medida, en comparación con la enumeración manual de folds de validación cruzada.

# Ajuste de parámetros del modelo de aprendizaje profundo con búsqueda de cuadrícula

---

- Podemos proporcionar argumentos a la función `fit()`. La función que especificamos para el argumento `build_fn` al crear la envoltura `KerasClassifier` también puede tomar argumentos. Podemos usar estos argumentos para personalizar aún más la construcción del modelo.
- Podemos usar una búsqueda de cuadrícula para evaluar diferentes configuraciones para nuestro modelo de red neuronal y reportar qué combinación proporciona el mejor desempeño