---

super() is a special use of the super keyword where you call a parameterless parent constructor. In general, the super keyword can be used to call overridden methods, access hidden ...

As for chaining super::super, as I mentionned in the question, I have still to find an interesting use to that. For now, I only see it as a hack, but it was worth mentioning, if only for the differences with Java ...

Thirdly, when you call super() you do not need to specify what the super is, as that is inherent in the class definition for Child. Below is a fixed version of your code which should perform ...

In fact, multiple inheritance is the only case where super() is of any use. I would not recommend using it with classes using linear inheritance, where it''s just useless overhead.

"super" object has no attribute "__sklearn_tags__". This occurs when I invoke the fit method on the RandomizedSearchCV object. I suspect it could be related to compatibility issues ...

A diretiva super, sem parênteses, permite ainda invocar métodos da classe que foi derivada através da seguinte syntax. super.metodo(); Isto é útil nos casos em que faças override ...

The implicit __class__ used by super does not exist at this point. Thus, referencing the superclass by the hardcoded name, as one had to do prior to super in Python2 will work - and is the ...

super() lets you avoid referring to the base class explicitly, which can be nice. But the main advantage comes with multiple inheritance, where all sorts of fun stuff can happen.