

A couple people have been trying this. Check out

[Differentiable Spearman in PyTorch \(Optimize for CORR directly\)](#) [

Data Science

](/c/data-science/5)

[@mdo](#) previously showed how to use a [custom loss function](#) which involved taking the gradient of the sharpe ratio of the Pearson correlations over different eras. Although Pearson and Spearman might return similar values, it could be rewarding to optimize for Spearman directly (or Sharpe of Spearman). Since the ranked Spearman correlation needs a sort operation (which is not differentiable), it has not been possible to compute the gradient with respect to predictions, which eliminated the possibil...

and

[Objective Function](#) [

Data Science

](/c/data-science/5)

Just writing this to share which target functions you use the most when training your models. I was thinking of customizing an Objective Function for boosted models in order to beat the common methods already developed. I know Spearman's correlation is non-differentiable due to sort and rank steps, but I found some references to try to deal with these problems:

[SoDeep fast-soft-sort](#)

I've tried to use SoDeep loss functions when training my MLPs and it was a complete disaster. So it would be ni...

I can't find where I read it, but I believe the fast-soft-sort package's Spearman function is only differentiable once, which would cause your None

issue. You could try [@mdo's solution, which is to just use a matrix of ones as the Hessian](#)