Implementing custom loss function in scikit learn

Asked 2 years, 5 months ago Active 1 year, 8 months ago Viewed 10k times



I want to implement a custom loss function in scikit learn. I use the following code snippet:



```
def my_custom_loss_func(y_true,y_pred):
    diff3=max((abs(y_true-y_pred))*y_true)
    return diff3
```



score=make_scorer(my_custom_loss_func,greater_ is_better=False)
clf=RandomForestRegressor()
mnn= GridSearchCV(clf,score)
knn = mnn.fit(feam,labm)



What should be the arguments passed into <code>my_custom_loss_func</code>? My label matrix is called <code>labm</code>. I want to calculate the difference between the actual and the predicted output (by the model) multiplied by the true output. If I use <code>labm</code> in place of <code>y_true</code>, what should I use in place of <code>y_pred</code>?

python machine-learning scikit-learn data-science gridsearchev

Share Improve this question

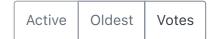
Follow



asked Jan 19 '19 at 13:47



3 Answers





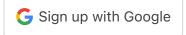
Okay, there's 3 things going on here:

Join Stack Overflow

to learn, share knowledge, and build your

career.

Sign up with email



Sign up with GitHub

Sign up with Fa

"loss fxn" for that purpose. If, however, you are trying to tune your whole model to perform well on, lets say, a recall test - then you need a recall optimizer to be part of the training process. It's tricky, but you can do it...

- 1) Open up your classifier. Let's use an RFC for example: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- 2) click [source]
- 3) See how it's inheriting from ForestClassifier? Right there in the class definition. Click that word to jump to it's parent definition.
- 4) See how this new object is inheriting from ClassifierMixin? Click that.
- 5) See how the bottom of that ClassifierMixin class says this?

```
from .metrics import accuracy_score
return accuracy_score(y, self.predict(X), sample_weight=sample_weight)
```

That's your model being trained on accuracy. You need to inject at this point if you want to train your model to be a "recall model" or a "precision model" or whatever model. This accuracy metric is baked into SKlearn. Some day, a better man than I will make this a parameter which models accept, however in the mean time, you gotta go into your sklearn installation, and tweak this accuracy_score to be whatever you want.

Best of luck!

Share Improve this answer Follow

answered Oct 24 '19 at 22:53





The documentation for make_scorer goes like this:



```
sklearn.metrics.make_scorer(score_func, greater_is_better=True,
needs_proba=False,
needs_threshold=False, **kwargs)
```



So, it dosen't need you to pass arguments while calling the function. Is this what you were asking?

Share Improve this answer Follow



I mean while defining the fucnction, my_custom_func_loss, (the first line of my code), I need to pass arguments right? Without arguments, how can I write the body of the function? I was asking about those arguments. Here I passed y_true and y_pred. - Moonzarin Esha Jan 19

yea! you only need those 2 arguments. However, if you want to pass some additional arguments you could do something like this: score_func(y, y_pred, **kwargs) where **kwargs are the extra parameters that you'd want to pass - Abhishek Jan 19 '19 at 15:52 🎤

I mean by default y will be assigned the label matrix and y_pred the predicted values of the model? don't I need to define those values in the code? I have seen people writing truths, preds. So can we write anything as the argument and scikit learn will be able to make out? It seems a bit weird. - Moonzarin Esha Jan 19 '19 at 16:03

See, If you pass them in the order it would take the arguments as you have defined within the function. ex: Let's suppose we have a function like this costFunc(y, y_pred). Now, if you pass values like costFunc(labels, predictions) then labels would be passed to y and predictions would be passed to y_pred. However, you could do an alternative like this: costFunc(y_pred = predictions, y = labels). As you can see the order is no longer required if you mention the name and pass. - Abhishek Jan 19 '19 at 17:12

I mean by default scikit learn will assume the first argument is the true label and the second argument corresponds to the predicted model output? If i write only y and y_pred, without explicitly mentioning anywhere what is y and what is y_pred, it will still work? –

Moonzarin Esha Jan 19 '19 at 18:43



The arguments of your my_custom_func_loss, does not have any connection with your true labels, which is labm. You can keep the way as it now.





Internally GridSearchCV will call the scoring function hence your true labels does not conflict there. y_pred would be the predicted values, generated from the model's output. y_true will be assigned with the values of labm.



Share Improve this answer Follow

answered Jan 19 '19 at 14:57



Venkatachalam

13.2k