



Hyperparameter Tuning

Apply grid search cross-validation to XGBoost models.

Chapter Goals:

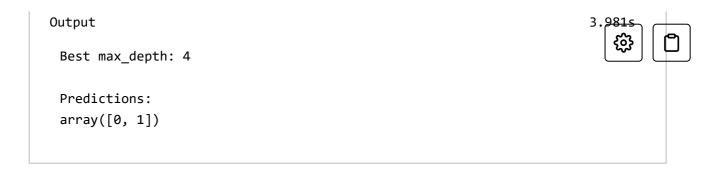
• Apply grid search cross-validation to an XGBoost model

A. Using scikit-learn's GridSearchCV

One of the benefits of using XGBoost's scikit-learn style models is that we can use the models with the actual scikit-learn API. A common scikit-learn object used with XGBoost models is the GridSearchCV (https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV .html#sklearn.model_selection.GridSearchCV) wrapper. For more on GridSearchCV see the **Data Modeling** section.

The code below applies grid search cross-validation to a binary classification XGBoost model.

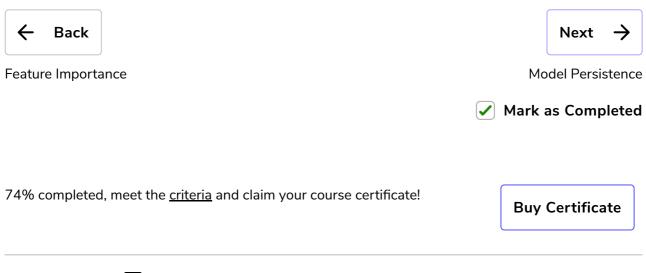
```
model = xgb.XGBClassifier()
    params = {'max_depth': range(2, 5)}
 3
 4 from sklearn.model_selection import GridSearchCV
5 cv model = GridSearchCV(model, params, cv=4, iid=False)
 6
7 # predefined data and labels
8 cv_model.fit(data, labels)
9 print('Best max_depth: {}\n'.format(
10
      cv_model.best_params_['max_depth']))
11
12 # new data contains 2 new data observations
    print('Predictions:\n{}'.format(
13
14
      repr(cv_model.predict(new_data))))
                                                              ני
\triangleright
```



In the code above, we applied grid search cross-validation to a binary classification XGBoost model to find the optimal 'max_depth' parameter (in the range from 2 to 4, inclusive). The K-fold cross-validation (the default for grid search) uses 4 folds. Note that the cross-validation process works the same for an XGBRegressor object.

After calling fit on data and labels, cv_model represents the cross-validated classification model trained on the dataset. The grid search cross-validation automatically chose the best performing 'max_depth' parameter, which in this case was 4. The best_params_ attribute contains the best performing hyperparameters after cross-validation.

The official XGBoost documentation provides a list (https://xgboost.readthedocs.io/en/latest/parameter.html) of the possible parameters we can tune for in a model. A couple commonly tuned parameters are 'max_depth' and 'eta' (the learning rate of the boosting algorithm).



(!) Report an Issue

? Ask a Question

(https://discuss.educative.io/tag/hyperparameter-tuning__gradient-boosting-with-xgboost__machine-learning-for-software-engineers)



