



Storing Boosters

Save and load Booster objects using XGBoost binary files.

Chapter Goals:

• Learn how to save and load Booster models in XGBoost

A. Saving and loading binary data

After finding the best parameters for a Booster and training it on a dataset, we can save the model into a binary file. Each Booster contains a function called save_model

(https://xgboost.readthedocs.io/en/latest/python/python_api.html#xgboost.Booster.save_model), which saves the model's binary data into an input file.

The code below saves a trained Booster object, bst, into a binary file called *model.bin*.

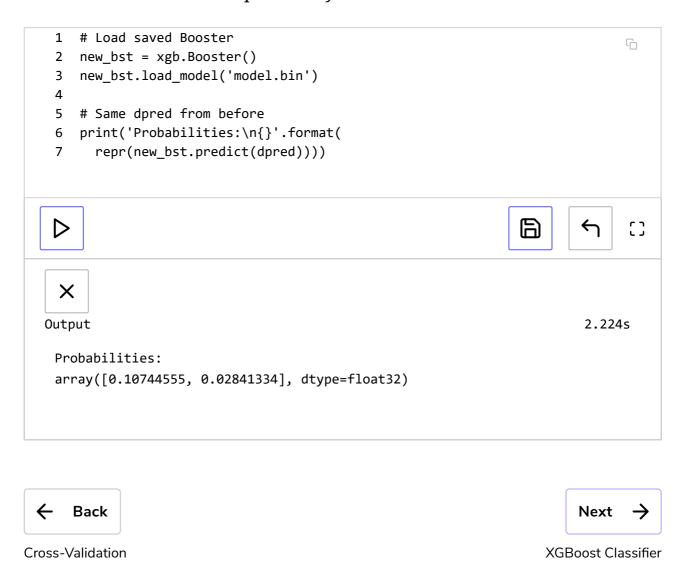
```
1 # predefined data and labels
 2 dtrain = xgb.DMatrix(data, label=labels)
 3 params = {
      'max_depth': 3,
      'objective': 'binary:logistic'
 6
7 bst = xgb.train(params, dtrain)
9 # 2 new data observations
10 dpred = xgb.DMatrix(new_data)
11 print('Probabilities:\n{}'.format(
12
      repr(bst.predict(dpred))))
13
14 bst.save_model('model.bin')
                                                              \triangleright
```

```
Output

[08:54:40] /workspace/src/tree/updater_prune.cc:74: tree pruning end, 1 ro
```

We can restore a Booster from a binary file using the load_model (https://xgboost.readthedocs.io/en/latest/python/python_api.html#xgboost.Booster.load_model) function. This requires us to initialize an empty Booster and load the file's data into it.

The code below loads the previously saved Booster from *model.bin*.



✓ Mark as Completed



Report an Issue

? Ask a Question

 $(https://discuss.educative.io/tag/storing-boosters__gradient-boosting-with-xgboost__machine-learning-for-software-engineers)\\$