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`, which saves the model's binary data into an input file.

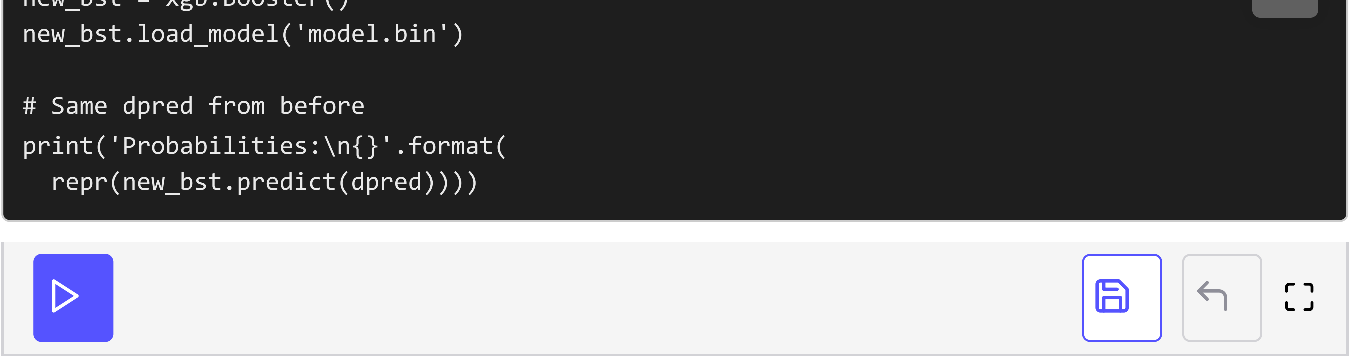
The code below saves a trained `Booster` object, `bst`, into a binary file called *model.bin*.

```
# predefined data and labels
dtrain = xgb.DMatrix(data, label=labels)
params = {
  'max_depth': 3,
  'objective':'binary:logistic'
}
bst = xgb.train(params, dtrain)

# 2 new data observations
dpred = xgb.DMatrix(new_data)
print('Probabilities:\n{}'.format(
  repr(bst.predict(dpred))))

bst.save_model('model.bin')
```

We can restore a `Booster` from a binary file using the `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*.

```
# Load saved Booster
new_bst = xgb.Booster()
```



```
new_bst = xgb.Booster()
new_bst.load_model('model.bin')

# Same dpred from before
print('Probabilities:\n{}'.format(
  repr(new_bst.predict(dpred))))
```