Machine Learning Models: How Do You Save Them?

Saving a trained machine learning model is a critical step in the machine learning workflow. This article demonstrates how to save and then load machine learning models that have been trained.

It is advised that your data set be divided into three sections:

Training: 60%

Validation: 20%

Test: 20%

After training the model on the training data, it is validated and tested on the validation and test data.

Frequently, the longest period of time is spent training the model. As a result, we may save time by training the model once and reloading it as needed.

1. For StatsModel Machine Learning Models:

Saving the model:

```
import statsmodels.api as sm
model = sm.tsa.ARIMA([1,5,9,12], order=(1, 0, 1))
my model= model.fit()
my model.save(myfile)
```

Loading The Model:

```
from statsmodels.tsa.arima model import ARIMAResults
loaded = ARIMAResults.load(my file)
```

2. For Scikit-Machine Learn's Learning Models

Saving The Model:

Serialize and save models using Pickle.

```
from sklearn.linear model import LogisticRegression
import pickle
model = LogisticRegression()
model.fit(xtrain, ytrain)
pickle.dump(model, open(model file path, 'wb'))
```

Loading The Model:

Use Pickle to deserialize and save the models

```
model = pickle.load(open(model file path, 'rb'))
result val = model.score(xval, yval)
result test = model.score(xtest, ytest)
```

Saving The Model

Serialize and store the models using JobLib.

```
from sklearn.linear model import LogisticRegression
from sklearn.externals import joblib
model = LogisticRegression()
model.fit(xtrain, ytrain)
joblib.dump(model, model file path)
```

Load The Model

Serialize and store the models using JobLib.

```
model = joblib.load
(model file path)
result val = model.score(xval, yval)
result test = model.score(xtest, ytest)
```

3. For Keras Machine Learning Models:

```
Create and train the model
from keras.models import Sequential
from keras.layers import Dense
model = Sequential()
model.fit(xtrain, ytrain)
```

Saving The Model

```
json file = model.to json()
with open(json file path, "w") as file:
file.write(json file)
model.save weights(h5 file)
```

Loading The Model

```
from keras.models import model from json
file = open(json file, 'r')
model json = file.read()
file.close()
loaded model = model from json(model json)
loaded model.load weights(h5 file)
```