

# How To Save Trained Machine Learning Models?

## Save & Reload Your Trained Machine Learning Models In Python

One of the key steps in the machine learning workflow is to save a trained machine learning model.

Once the model is trained on the training set, the model is validated and tested on the validation and test set.

Training the model often takes the longest amount of time. Hence it can save us time to train the model once and reload it if and when it is required.

### 1. If you are working with StatsModel Machine Learning Models

#### 1.1 Save 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)
```

#### 1.2 Load The Model

```
from statsmodels.tsa.arima_model import ARIMAResults
loaded = ARIMAResults.load(my_file)
```

### 2. If you are working with Scikit-Learn Machine Learning Models

#### 2.1 Save The Model

Use Pickle to serialise and save the models

```
from sklearn.linear_model import LogisticRegression
import pickle
model = LogisticRegression()
model.fit(xtrain, ytrain)
# save the model to disk
pickle.dump(model, open(model_file_path, 'wb'))
```

#### 2.2 Load The Model

Use Pickle to deserialise and save the models

```
model = pickle.load(open(model_file_path, 'rb'))
result_val = model.score(xval, yval)
result_test = model.score(xtest, ytest)
```

## 2.3 Save The Model

Use JobLib to serialise and save the models

```
from sklearn.linear_model import LogisticRegression
from sklearn.externals import joblib
model = LogisticRegression()
model.fit(xtrain, ytrain)# save the model to disk
joblib.dump(model, model_file_path)
```

## 2.4 Load The Model

Use JobLib to deserialise and save the models

```
model = joblib.load(model_file_path)
result_val = model.score(xval, yval)
result_test = model.score(xtest, ytest)
```

# 3. If you are working with Keras Machine Learning Models

## Create and train the model

```
from keras.models import Sequential
from keras.layers import Dense# create model
model = Sequential()
# Fit the model
model.fit(xtrain, ytrain)
```

## 3.1 Save The Model

```
# serialize to JSON
json_file = model.to_json()
with open(json_file_path, "w") as file:
    file.write(json_file)
# serialize weights to HDF5
model.save_weights(h5_file)
```

## 3.2 Load The Model

```
from keras.models import model_from_json# load json and create model
file = open(json_file, 'r')
model_json = file.read()
file.close()loaded_model = model_from_json(model_json)
# load weights
loaded_model.load_weights(h5_file)
```