Quick Hacks To Save Machine Learning Model using Pickle and Joblib

Purnendu Shukla — Published On August 18, 2021 and Last Modified On July 26th, 2022

Advanced Libraries Machine Learning Project Python Structured Data Supervised

This article was published as a part of the Data Science Blogathon

TABLE OF CONTENTS

INTRODUCTION

Saving models is a crucial part of the realm of model development. To understand what it means let's understand it with a very simple example:

Suppose you are working on a practice problem related to house rent given lots of *data points and input features.* It's quite common to perform **EDA, Preprocessing(may need to create additional features)**, and **feeding our data to our model**. In this scenario even if we use the simplest **Linear Regression Model** (multiple variables) it may become *huge in size* due to all the input_features and all the parameters which will be time-consuming to re-train again and again for use.

So the simplest thing to do is to **save our model** and later load it for inference or prediction at a later time. While Keras models API provides the [**model.save()**] functionality for saving our deep learning model is limited to the realm of deep learning and for most beginners, in ML it's quite confusing to save their model. Also due to estimators having a huge number of parameters, it is quite advisable to save them. So in this article, we will look into few small hacks to save our model

Loading Dataset And Creating Our Model

We are going to use a house price prediction dataset with a single feature **area(for demonstration purposes)**. Our job will be to predict the price given the area. For keeping things simple we will have only 4-5 data points and the model we will be using will be a ***Linear Regression Model*** which just fits a straight line to our dataset and **calculates the square of predicted difference from actual differences over all data points***

> The square in cost function ensures that negative values are nullified

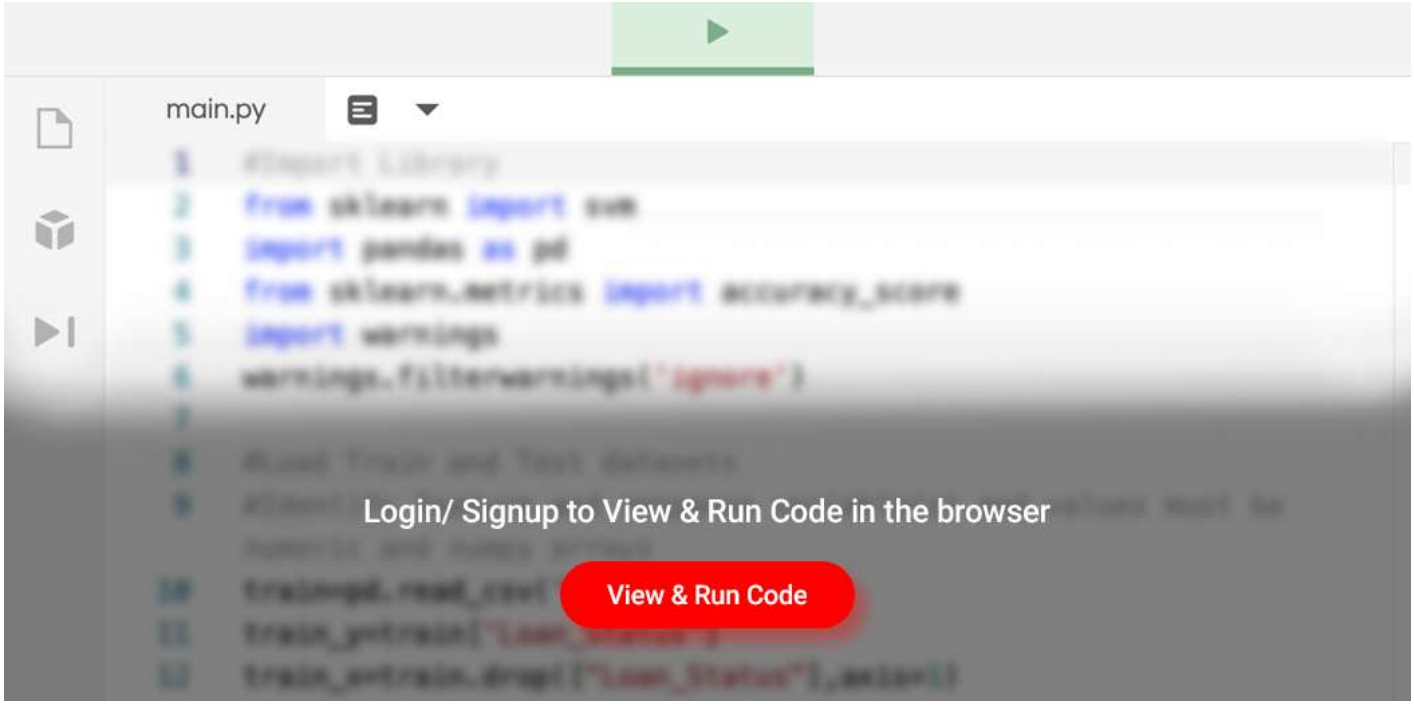
Creating Model Files

We are now quickly going to create our model file in 5 steps which we will be saving for later use.

1. We will start by loading all the required dependencies.

```
# loading dependencies
import pandas as pd
import numpy as np
from sklearn import linear_model
```

2. Now we will be loading our data using **pd.read_csv()** function into a pandas dataframe(**train_df**) and use **df.head()** method to print first 5 rows.



3. To create our model we will be first creating a model object which will be actually a **LinearRegression** classifier and then fit our model with our training samples and training labels for which our model job will be to find the best straight line fit.

```
# creating the model object
model = linear_model.LinearRegression() # y = mx+b
```

```
# fitting model with X_train - area, y_train - price
model.fit(train_df[['area']],train_df.price)
```

After executing the above code output will look a bit like this

```
>> LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

4. As we know a straight line has a coefficient and an intercept in the equation, so we should check out those values as sklearn provides some handy attributes. These can be checked as

```
# checking coeffiecent - m
model.coef
```

```
model.intercept_
```

>> 180616.43835616432

5. Finally for completeness sake one can test the model for predicting the price for a 5000sqft area house.

```
# predict model values - area = 5000
model.predict([[5000]])
```

>> array([859554.79452055])

Saving Model

It's now time to save our created model. We are going to look into 2 quick hacks for the saving model. Also as a bonus, I will be providing guidelines on where to use which method.

Method 1 – Pickle – 2 Steps

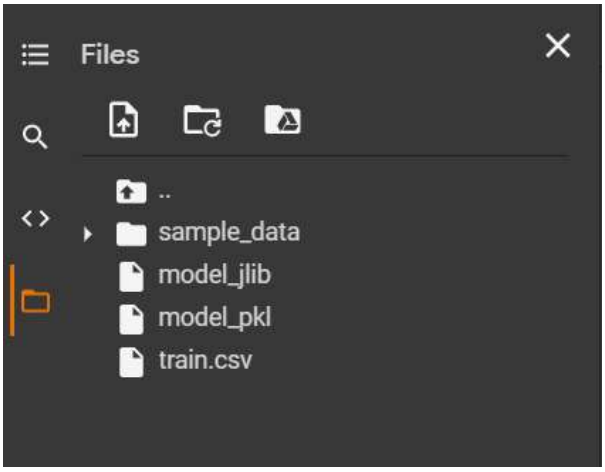
Many of you will be familiar with the pickle module, however, if not it's good to know that the pickle module allows you to pickle a file using de-serialization which means simply **breaking down an object into its constituting components.** For e.g, our model files attribute like the one we saw.

To save a file using pickle one needs to **open** a file, load it under some **alias name** and **dump** all the info of the model. This can be achieved using below code:

```
# loading library
import pickle
```

```
# create an iterator object with write permission - model.pkl
with open('model_pkl', 'wb') as files:
    pickle.dump(model, files)
```

After the above steps, one can see a file with the name **model_pkl** in the directory, and opening it will show something like this:



Directory As Shown In Google Collab

Quick Hacks To Save Machine Learning Model using Pickle and Joblib

inside model_pkl file

One can load this file back again into a model using the same logic, here we are using the **lr** variable for referencing the model and then using it to **predict** the price for 5000sqft:

```
# load saved model
with open('model_pkl' , 'rb') as f:
    lr = pickle.load(f)
```

```
# check prediction

lr.predict([[5000]]) # similar
```

>> array([859554.79452055])

Benefits:

- **The pickle module keeps track of the objects it has already serialized, so that later references to the same object won't be serialized again, thus allowing for faster execution time.**
- **Allows saving model in very little time.**
- **Good For small models with fewer parameters like the one we used.**
-

Method 2 – Joblib – 2 Steps

Joblib is an alternative to model saving in a way that it can operate on objects with large NumPy arrays/data as a backend with many parameters. It can be used as an individual module(refer here) or using the Sci-Kit Learn library. For simplicity's sake, we will be using the second method.

-> First, we will import **joblib** from **sklearn's external class**

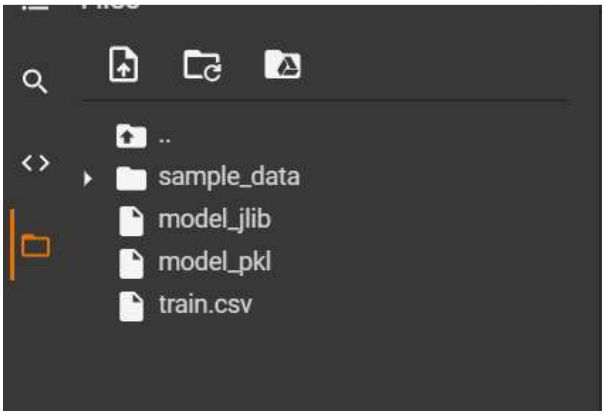
```
# loading dependency
from sklearn.externals import joblib
```

To save the model we will use its **dump** functionality to save the **model** to the **model_jlib** file.

```
# saving our model # model - model , filename-model_jlib
joblib.dump(model , 'model_jlib')
```

After running the above code a file will be created with a filename and contents will be similar to the pickle file.

Quick Hacks To Save Machine Learning Model using Pickle and Joblib



The directory

inside model_jlib file

> **Note: We didn't use an iterator as the module saves the data onto disk rather than string-names. However, it accepts file-like objects.**

To load the model we will be providing **file-path** or **file object** to the **load** function and storing it in the **m_jlib** variable, which we can later use for prediction.

```
# opening the file- model_jlib
m_jlib = joblib.load('model_jlib')
```

Finally for predicting we can call **predict** method on **m_jlib** and pass it a 2d array with values as 5000.

```
# check prediction
m_jlib.predict([[5000]]) # similar
```

>> array([859554.79452055])

> Note predict methods assumes you provide data in a 2d format so we used [[5000]] meaning 5000 as an 2d array

Benefits:

- **Ideal for the large models having many parameters and can have large NumPy arrays in the backend.**

Conclusion

Due to the time complexity involved in training large models, saving is becoming a crucial part of the data-science realm and with this article, I tried to introduce few quick ways to save them. However, it must be noted both the process works on the same concept of serialization(saving of data into its component form) and deserialization(restoring of data from the serialized chunks), thus it is advised to pickle or joblib the model from a trusted source.

Also for simplicity sake, we have used a Linear Regression model, but the same can be used to save models of different types like **Logistic Regression**, **Decision Trees**, **SVM's,** and a lot more:)

Hope you have enjoyed reading the article and learned something in the process. Those who want to dive deeper can refer to the reference section and work along.

References

Collab Notebook: – File containing all the codes can be found here.

Job Lib:- For those who want to learn more about joblib, refer here.

Inspiration:- A humble and respectful thanks to code basics which inspire me to write the content.

CSV File – The training data used can be downloaded from here

The media shown in this article are not owned by Analytics Vidhya and are used at the Author's discretion.

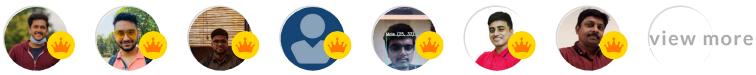
blogathon joblib pickle save machine learning model

About the Author

Purnendu Shukla

Hey All✋, My name is Purnendu Shukla a.k.a Harsh. I am a passionate individual who likes exploring & learning new technologies, creating real-life projects, and returning to the community as blogs. My Blogs range from various topics, including Data Science, Machine Learning, Deep Learning, Optimization Problems, Excel and Python Guides, MLOps, Cloud Technologies, Crypto Mining, Quantum Computing.

Our Top Authors



Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name*

Email*

Website

Notify me of follow-up comments by email.

Notify me of new posts by email.

Submit

Top Resources

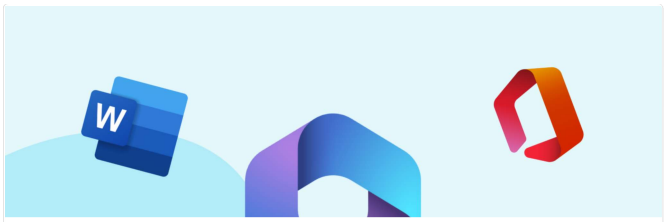
How to Read and Write With CSV Files in Python:..

Harika Bonthu - AUG 21, 2021

Understand Random Forest Algorithms With Examples (Updated 2023)

Sruthi E R - JUN 17, 2021

Feature Selection Techniques in Machine Learning (Updated 2023)



Microsoft Power Platform Copilot: No Coding Era Is Coming

Quick Hacks To Save Machine Learning Model using Pickle and Joblib