Save Model

Outlines

- Save & load model using pickle
- Save & load model using joblib

Solving a problem using a machine learning cosist of two steps:
1. Training your model using your training dataset.
2. Ask your questions to the train model that will give you your answers.

Often the size of dataset is pretty huge. Because as the size increases your model becomes more accurate. When your training dataset is so huge the training step becomes more time consuming. If you save the train model to a file you can later on use that same model to make the actual prediction.

Problem:

Load homeprice.csv file and use Linear Regression to make the actual prediction about area=5000.

In [6]:
```
import numpy as np
import pandas as pd
from sklearn import linear_model
```

In [7]:
```
df = pd.read_csv('D:/Data_Science/My Github/Machine-Learning-with-Python/4. Save M
df.head()
```

Out[7]:

	area	price
0	2600	550000
1	3000	565000
2	3200	610000
3	3600	680000
4	4000	725000

In [8]:
```
model = linear_model.LinearRegression()
model.fit(df[['area']],df.price)
```

Out[8]:
```
LinearRegression()
```

In [9]:
```
model.coef_
```

Out[9]:
```
array([135.78767123])
```

In [10]:
```
model.intercept_
```

Out[10]:
```
180616.43835616432
```

In [12]:
```
model.predict([[5000]])
```

Out[12]:
```
array([859554.79452055])
```

Save Model with pickle

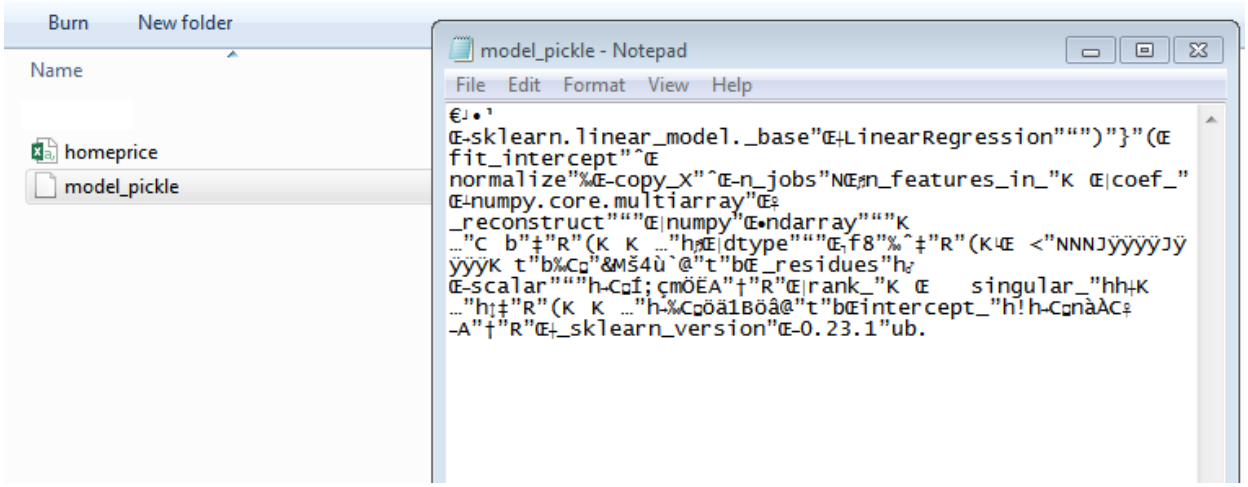
Pickle is the standard way of serializing objects in Python.
serialize (verb) = to print or broadcat a story in several separate parts.
You can use the pickle operation to serialize your machine learning algorithms and save the serialized format to a file.

Later you can load this file to deserialize your model and use it to make new predictions.

In [13]:
```
import pickle
```

In [15]:
```
with open('D:/Data_Science/My Github/Machine-Learning-with-Python/4. Save Model/Ex
    pickle.dump(model,f)
```

In my working directory it created this model_pickle file look like this.



Your model is save into a file. You can use this model to make prediction.

In [19]:
```
# This time open the file with read mode 'rb' & load model from a file into a memo
with open('D:/Data_Science/My Github/Machine-Learning-with-Python/4. Save Model/Ex
    mp = pickle.load(f)
```

In [20]:
```
# If I use this object to predict it will give me the same answer
mp.predict([[5000]])
```

Out[20]:
```
array([859554.79452055])
```

You can send this trained model to a friend of yours and he can use it.

Save Model with Sklearn Joblib

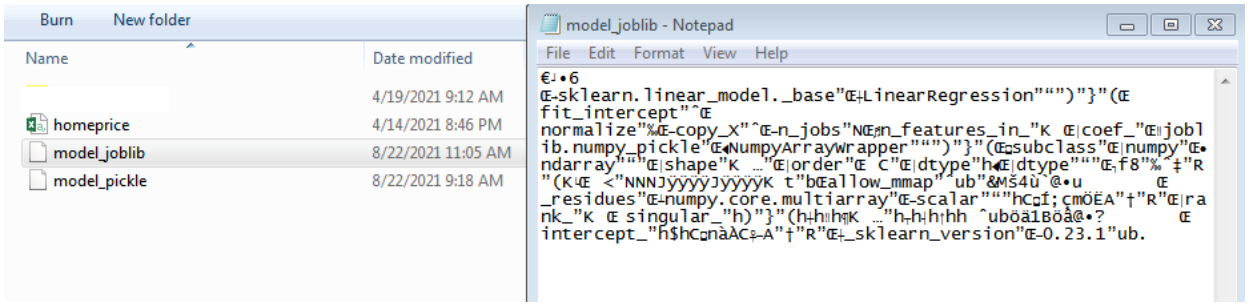
In the specific case of scikit-learn, it may be better to use joblib's replacement of pickle (dump & load), which is more efficient on objects that carry large numpy arrays internally as is often the case for fitted scikit-learn estimators, but can only pickle to the disk and not to a string.

In [28]:
```
import joblib
```

In [29]:
```
joblib.dump(model,'D:/Data_Science/My Github/Machine-Learning-with-Python/4. Save
```

Out[29]:
```
['D:/Data_Science/My Github/Machine-Learning-with-Python/4. Save Model/Example/m
odel_joblib']
```

The result is something like this:



In [31]:
```
# Load The Model
mj=joblib.load('D:/Data_Science/My Github/Machine-Learning-with-Python/4. Save Mod
```

In [32]:
```
# Make actual prediction
mj.predict([[5000]])
```

Out[32]:
```
array([859554.79452055])
```

Everything in the model is just like the actual model. coef, intecept, ...

In [33]:
```
mj.coef_
```

Out[33]:
```
array([135.78767123])
```

In [34]:
```
mj.intercept_
```

Out[34]:
```
180616.43835616432
```

Use this link for more information about model saving (https://scikit-learn.org/stable/modules/model_persistence.html)

Date	Author
2021-08-22	**Ehsan Zia**