Desh Iyer | 500081889 | Experiment - 7

April 2, 2023

1 Experiment 7 - Use a Classifier and Create an API to it

1.1 Import Libraries

[ ]:
```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
import pickle

import numpy as np
import requests
import json

from flask import Flask, jsonify
from flask_restful import Api, Resource, reqparse
```

1.2 Train the Model

[ ]:
```
# Load Iris data
iris = load_iris()

# Split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(iris['data'],␣
 ↪iris['target'], random_state=12)

# Train the model
clf = RandomForestClassifier(random_state=12)
clf.fit(X_train, y_train)

# Make prediction on the test set
y_predict = clf.predict(X_test)
print(y_predict)

# Save model
with open('model.pickle', 'wb') as f:
    pickle.dump(clf, f)
```

```
[0 2 0 1 2 2 2 0 2 0 1 0 0 0 1 2 2 1 0 1 0 1 2 1 0 2 2 1 0 0 0 1 2 0 2 0 1
 1]
```

1.3 Test the Model

[ ]:
```
# Load data
iris = load_iris()

# Split into train and test sets using the same random state
X_train, X_test, y_train, y_test = train_test_split(iris['data'],␣
 ↪iris['target'], random_state=12)

# Serialize the data into json and send the request to the model
payload = {'data': json.dumps(X_test.tolist())}
y_predict = requests.post('http://127.0.0.1:5000/iris', data=payload).json()

# Make array from the list
y_predict = np.array(y_predict)
print(y_predict)
```

1.4 Deploy the Model to a RESTfulAPI

[ ]:
```
app = Flask(__name__)
api = Api(app)

# Create parser for the payload data
parser = reqparse.RequestParser()
parser.add_argument('data')

# Define how the api will respond to the post requests
class IrisClassifier(Resource):
    def post(self):
        args = parser.parse_args()
        X = np.array(json.loads(args['data']))
        prediction = model.predict(X)
        return jsonify(prediction.tolist())

api.add_resource(IrisClassifier, '/iris')

if __name__ == '__main__':
    # Load model
    with open('model.pickle', 'rb') as f:
        model = pickle.load(f)

    app.run(debug=True)
```

```
 * Serving Flask app '__main__'
```

```
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with watchdog (inotify)
/home/volt/.local/lib/python3.10/site-packages/traitlets/traitlets.py:2547:
FutureWarning: Supporting extra quotes around strings is deprecated in traitlets
5.0. You can use 'hmac-sha256' instead of '"hmac-sha256"' if you require
traitlets >=5.
  warn(
/home/volt/.local/lib/python3.10/site-packages/traitlets/traitlets.py:2498:
FutureWarning: Supporting extra quotes around Bytes is deprecated in traitlets
5.0. Use 'ec5dfbcb-f82d-4d70-b78d-fa520930b867' instead of
'b"ec5dfbcb-f82d-4d70-b78d-fa520930b867"'.
  warn(
Traceback (most recent call last):
  File "/home/volt/.local/lib/python3.10/site-packages/ipykernel_launcher.py",
line 17, in <module>
    app.launch_new_instance()
  File "/home/volt/.local/lib/python3.10/site-
packages/traitlets/config/application.py", line 1040, in launch_instance
    app.initialize(argv)
  File "/home/volt/.local/lib/python3.10/site-
packages/traitlets/config/application.py", line 113, in inner
    return method(app, *args, **kwargs)
  File "/home/volt/.local/lib/python3.10/site-packages/ipykernel/kernelapp.py",
line 665, in initialize
    self.init_sockets()
  File "/home/volt/.local/lib/python3.10/site-packages/ipykernel/kernelapp.py",
line 309, in init_sockets
    self.shell_port = self._bind_socket(self.shell_socket, self.shell_port)
  File "/home/volt/.local/lib/python3.10/site-packages/ipykernel/kernelapp.py",
line 246, in _bind_socket
    return self._try_bind_socket(s, port)
  File "/home/volt/.local/lib/python3.10/site-packages/ipykernel/kernelapp.py",
line 222, in _try_bind_socket
    s.bind("tcp://%s:%i" % (self.ip, port))
  File "/home/volt/.local/lib/python3.10/site-packages/zmq/sugar/socket.py",
line 232, in bind
    super().bind(addr)
  File "zmq/backend/cython/socket.pyx", line 568, in
zmq.backend.cython.socket.Socket.bind
  File "zmq/backend/cython/checkrc.pxd", line 28, in
zmq.backend.cython.checkrc._check_rc
zmq.error.ZMQError: Address already in use
```

```
An exception has occurred, use %tb to see the full traceback.

SystemExit: 1
```

```
/home/volt/.local/lib/python3.10/site-
packages/IPython/core/interactiveshell.py:3441: UserWarning: To exit: use
'exit', 'quit', or Ctrl-D.
  warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```