Using the web service

A simple web service is launched on <u>pythonanywhere.com</u> following the instruction in:

https://github.com/nindate/ml-zoomcamp-exercises/blob/main/how-to-usepythonanywhere.md

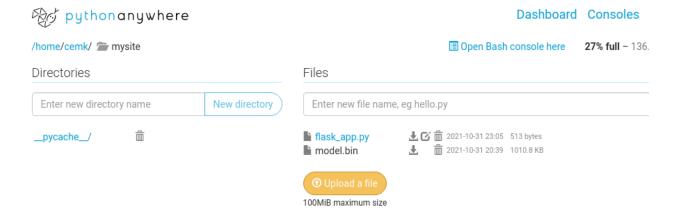
The code running on the host is given in file: <code>flask_app.py</code> (under <code>web_service</code> folder) It looks like below on the pythonanywhere editor:



```
from flask import Flask, request, jsonify
2 import pickle
3 import numpy as np
5 app = Flask(__name__)
7 print("dd")
9 - with open('mysite/model.bin', 'rb') as f_in:
10
        dv, model = pickle.load(f_in)
11
12
13 @app.route('/predict', methods=['POST'])
14 - def predict():
15
        building = request.get_json()
16
        X = dv.transform([building])
17
        y pred = model.predict(X)
18
19
20 -
        result = {
            'heating': round(float(np.exp(y_pred[0,0])), 2),
21
            'cooling': round(float(np.exp(y_pred[0,1])), 2)
22
23
        return jsonify(result)
24
25
```

Note that, the model file is also uploaded:

Using the web service 1



In order to reach it, use the *cloud-test.ipynb* file in *web_service* folder as below:

```
+ Code + Markdown | D≫ Run All ···
                                           Python 3.9.5 64-bit ('midterm_project-TBMLFfwK': pipenv)
          import requests
                                                                                   Python
         0.8s
        url = 'http://cemk.pythonanywhere.com/predict'
         0.2s
                                                                                   Python
           building = {"compactness": 0.9,
        2
                   "surface area": 563.5,
                                                     Set these
        3
                   "wall area": 318.5,
                                                      parameters
        4
                   "roof area": 122.5,
        5
                   "height": 7.0,
        6
                   "orientation": 5.0,
        7
                   "glazing_area": 0.4,
        8
                   "glazing_distribution": 4.0}
                                                                                   Python
      ✓ 0.1s
> <
           response = requests.post(url, json=building).json()
        2
        3 print(response)
         0.6s
                                                                                   Python
                                             ← Results
    {'cooling': 35.88, 'heating': 36.23}
```

Using the web service 2

Alternatively, you can modify the *cloud-test.py* file (in the same folder) by setting building parameters in your editor as you wish and run it with:

python cloud-test.py

Using the web service 3