



Productionizing machine learning models with flask & cookiecutter



Hello!

I am **Katie Simmons**

The GitHub repo for this talk is here:

<https://github.com/katie-simmons/productionizing-ml-models>



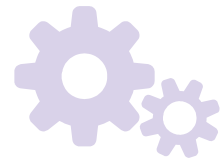
Flask

- Python web framework
- Create REST API
- Predict endpoint
POSTed JSON, returns
JSON w/prediction



COOKIECUTTER

- Python command line utility
- Generate project boilerplate
- Jinja2 templating



Pre-cookiecutter: Getting ML Models into Production



Data Scientist experiments to create a machine learning model 👍



Data Scientist trains and tunes the machine learning model 👍



Data Scientist creates a new application from scratch over a few days to a week 🤖



Data Scientist manually deploys, troubleshooting over a day or two 😬

Post-cookiecutter: Getting ML Models into Production



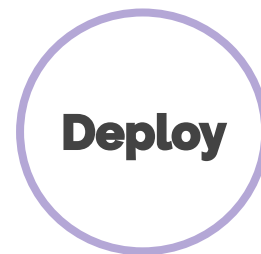
Data Scientist experiments to create a machine learning model 👍



Data Scientist trains and tunes the machine learning model 👍

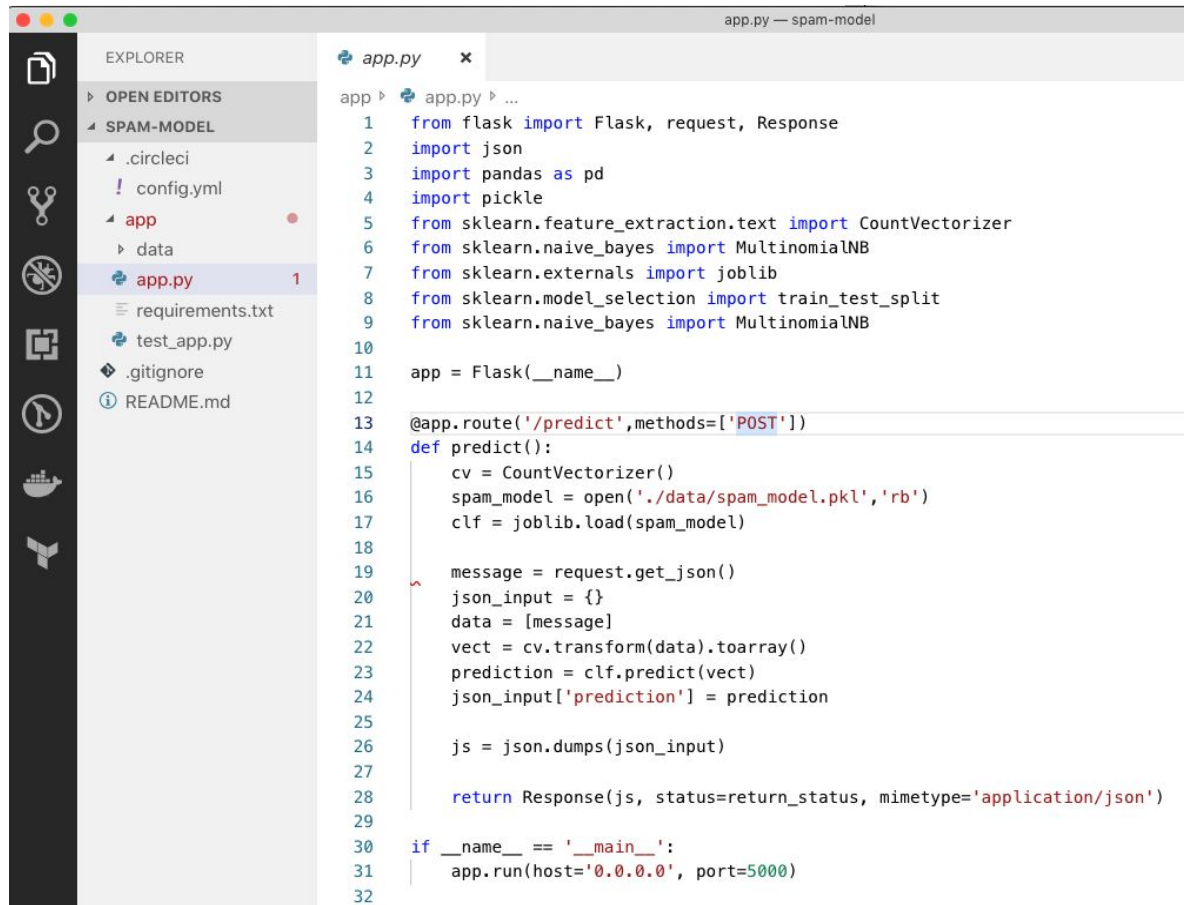


Data Scientist uses cookiecutter, drops in some code from their Jupyter notebook and tests locally 👍



DevOps Engineer confirms configuration and Data Scientist smoke tests prediction 👍

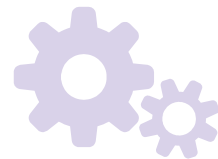
Flask app



```
app.py — spam-model
app.py
1 from flask import Flask, request, Response
2 import json
3 import pandas as pd
4 import pickle
5 from sklearn.feature_extraction.text import CountVectorizer
6 from sklearn.naive_bayes import MultinomialNB
7 from sklearn.externals import joblib
8 from sklearn.model_selection import train_test_split
9 from sklearn.naive_bayes import MultinomialNB
10
11 app = Flask(__name__)
12
13 @app.route('/predict', methods=['POST'])
14 def predict():
15     cv = CountVectorizer()
16     spam_model = open('./data/spam_model.pkl', 'rb')
17     clf = joblib.load(spam_model)
18
19     message = request.get_json()
20     json_input = {}
21     data = [message]
22     vect = cv.transform(data).toarray()
23     prediction = clf.predict(vect)
24     json_input['prediction'] = prediction
25
26     js = json.dumps(json_input)
27
28     return Response(js, status=return_status, mimetype='application/json')
29
30 if __name__ == '__main__':
31     app.run(host='0.0.0.0', port=5000)
32
```



Cookiecutter project



cookiecutter gh:katie-simmons/productionizing-ml-models

```
EXPLORER
└─ OPEN EDITORS
  └─ PRODUCTIONIZING-ML-MODE...
    └─ .vscode
      └─ {{cookiecutter.project_slu...
        └─ .circleci
          └─ app
            └─ data
              └─ app.py
                └─ requirements.txt
                  └─ test_app.py
                    └─ .gitignore
                      └─ Dockerfile
                        └─ README.md
                          └─ .gitignore
                            └─ cookiecutter.json
                              └─ README.md

app.py
{{cookiecutter.project_slug}} > app > app.py > ...
You, a few seconds ago | 1 author (You)
1 from flask import Flask, request, Response
2 import json
3 import pandas as pd
4 import pickle
5
6 app = Flask(__name__)
7
8 @app.route('/predict', methods=['POST'])
9 def predict():
10     {{cookiecutter.project_slug}} = open('./data/{{cookiecutter.proje
11     clf = joblib.load('{{cookiecutter.project_slug}}')
12
13     message = request.get_json()
14     json_input = {}
15     data = [message]
16     prediction = CHANGE
17     json_input['prediction'] = prediction
18
19     js = json.dumps(json_input)
20
21     return Response(js, status=return_status, mimetype='application/j
22
23 if __name__ == '__main__':
24     app.run(host='0.0.0.0', port=5000)
25

cookiecutter.json
{{cookiecutter.project_slug}} > ...
You, a day ago | 1 author (You)
1 {
2     "author": "Your name",
3     "email": "your-name@gmail.com",
4     "github_username": "Your github username",
5     "project_name": "Required unless you want your project to be name
6     "repo_name": "{{ cookiecutter.project_name.lower().replace(' ',
7     "project_slug": "{{ cookiecutter.project_name.lower().replace('
8     "project_short_description": "A short description of the project"
9     "release_date": "{{% now 'local' %}}",
10    "version": "1.0.0",
11    "_extensions": ["jinja2_time.TimeExtension"]
12 }
13
```

Cookiecutter

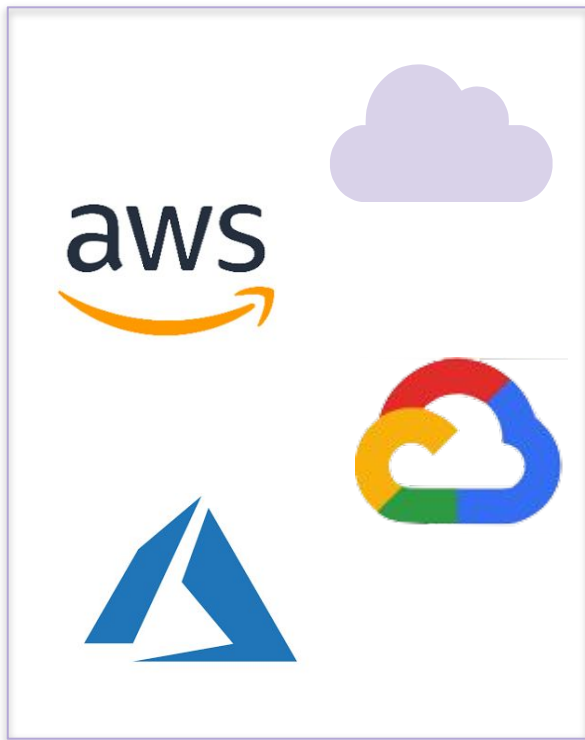
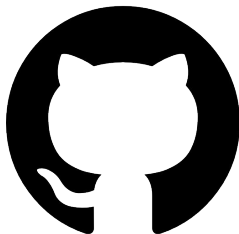
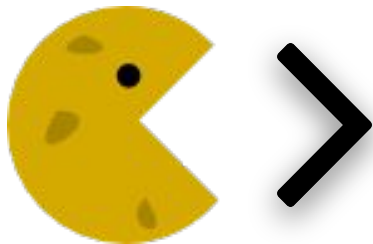
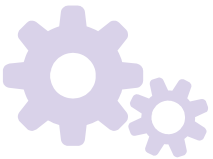
result =
Flask app



The screenshot shows a VS Code editor window with a project named 'TEST-SPAM-MODEL'. The Explorer sidebar on the left shows the file structure: `.circleci`, `config.yml`, `app` (selected), `requirements.txt`, `test_app.py`, `.gitignore`, `Dockerfile`, and `README.md`. The main editor area shows the `app.py` file with the following code:

```
1 from flask import Flask, request, Response
2 import json
3 import pandas as pd
4 import pickle
5 from sklearn.feature_extraction.text import CountVectorizer
6 from sklearn.naive_bayes import MultinomialNB
7 from sklearn.externals import joblib
8 from sklearn.model_selection import train_test_split
9 from sklearn.naive_bayes import MultinomialNB
10
11 app = Flask(__name__)
12
13 @app.route('/predict', methods=['POST'])
14 def predict():
15     cv = CountVectorizer()
16     test_spam_model = open('./data/test-spam-model.pkl', 'rb')
17     clf = joblib.load(test_spam_model)
18
19     message = request.get_json()
20     json_input = {}
21     data = [message]
22     vect = cv.transform(data).toarray()
23     prediction = clf.predict(vect)
24     json_input['prediction'] = prediction
25
26     js = json.dumps(json_input)
27
28     return Response(js, status=return_status, mimetype='application/json')
29
30 if __name__ == '__main__':
31     app.run(host='0.0.0.0', port=5000)
32
```


Into production





Thanks!

Any questions?

You can find me at katie-simmons or @katiecodes11 or
katie@katie.codes

You can find cookiecutter here:

<https://cookiecutter.readthedocs.io/en/latest/readme.html>