PyData Test Code

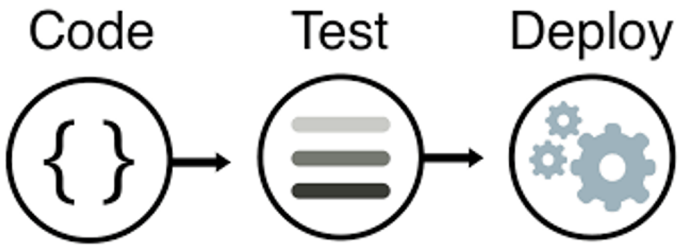
TEAMLAB 박원영
2020.01.25

Test Code를 작성하지 않는 이유

- 시간
- 작성 방법에 대한 무지
- 테스트 코드 중요성 의문

Test Code를 작성해야 하는 이유

- 프로그램의 안전성 보장
- 버그 대응
- 배포 후 시간 절약
- 팀 생산성 향상



HIV 약물 내성 예측

```
# utils.py

def read_protein(filename):
    sequence = ... # stuff happens
    return sequence

# Returns array 5 times the length of sequence.
def featurize(sequence):
    features = ... # stuff happens
    return features

# Return model predictions.
def predict(features):
    model = ... # load scikit-learn model
    return model.predict(features)
```

Test Code

```
from utils import featurize

def test_featurize():
    sequence = "MKALVIELQDPG..."  # something 99 amino acids l
    feats = featurize(sequence)

    assert feats.shape[0] == 1
    assert feats.shape[1] == len(sequence) * 5
```

```
# An integration test for the predict function.
def test_predict():
    sequence = "MKALVIELQDPG..."  # something 99 amino acids l
    feats = featurize(sequence)
    preds = predict(feats)
```

Test Code

```
acceptable_letters = set('ACDEFGHIJKLMNPQRSTVWXY')
def featurize(sequence):
    if not len(sequence) == 99:
        raise ValueError("put informative error here.")
    if not set(sequence).issubset(acceptable_letters):
        raise Exception("put informative error here.")
    features = ... # stuff happens
    return features
```

pytest VS unittest

```python
def test_upper():
    assert "foo".upper() == "FOO"
```

```python
from unittest import TestCase

class UpperTestCase(TestCase):
    def test_upper(self):
        self.assertEqual("foo".upper(), "FOO")
```

- unittest 보다 간결하고 편리
- 테스팅 프레임워크로써 장점 많음(픽스쳐, assert문 활용)

fixture : 테스팅을 하는데 필요한 부분/조건들을 미리 준비해놓은 리소스 혹은 코드

DATA

- 복잡
- 진화
- 데이터 관리 요구사항 변화

테스트 데이터

- column names
- column data types
- nullity
- bounds

```
def test_query_function():
    data = query_function()

    # Column tests:
    expected_columns = [...]
    assert set(expected_columns) == set(data.columns)

    # Null checks: this column __must__ be fully populated
    assert pd.isnull(df[column_name]).sum() == 0
```

Testing in data science

- CI 시스템 만들기
- 테스트 검토 없이는 병합하지 않기
- 개발운영팀과 신뢰 쌓기

Code
Data

Test

감사합니다.

참고

https://ericmjl.github.io/testing-for-data-scientists/#/12

https://www.youtube.com/watch?v=5RKuHvZERLY

https://www.bangseongbeom.com/unittest-vs-pytest.html