

**Testing in ML**



Chandan Agrawal

**Effective ML Workshop**

September 20-21, 2022

**www.wadhwaniai.org**



**AI FOR SOCIAL IMPACT**

Contents

* Why should we test?
* What all should we test?
* How should we test?

Why Should We Test?

* Easy to add new features
* To avoid a wasted training job.
* Better to find shortcomings on your own than from the clients.

What to Test?

1. Pre-train Tests
   1. Testing the code and the data
   2. Similar to software testing and the same principles can be used.
2. Post-train Tests
   1. Testing the model (testing model’s skills)
   2. Well tested code and data are not sufficient to ensure the creation of a good model.
      1. Evaluation of validation set
      2. Testing the model for some expected behaviour.
   3. These tests contain a set of inputs on which model’s performance is calculated.

# Pre-train Tests

* Testing the code and the data
* Similar to software testing and the same principles can be used.
* Types

a. Unit tests:

■ To test small components of the code and the data.

■ Any new function/class added, should have unit test testing it’s functionality. b. Regression tests:

■ To test bugs which have been encountered before and fixed. ■ Can be added as a new unit test.

c. Integration tests:

■ To test execution of multiple small components of the code.

■ (find this out)

# Unit Tests

* Unit tests in ML

○ Data

■ Check for Nan values at unexpected places.

■ Check for leakage in your splits (by intersection).

■ Check if features values are in the expected range. Similarly for categories.

○ DataLoader

■ Check sample’s shapes (correct or not)

■ Check data loading (possible or not)

■ Check augmentations (applied or not, shapes) ■ Check everything separately for train and test samples.

* Ex: Augmentations differ for train and test.

# Unit Tests

● Unit tests in ML

○ Model

■ Check output shapes, forward pass.

■ Check cpu to gpu to cpu transfer.

■ Check sample independence.

■ Check gradient’s existence for all the expected parameters.

○ Loss

■ Check output type, shape.

■ Check expected behaviour.

○ Trainer

■ Ensure fitting (by overfitting a single batch).

# Post-train Tests

Model evaluation on validation set

* We usually do this.
* Effective to compare between two models of 70% and 90% accuracy.
* But comparing models of 90% and 91% accuracy gets tricky.
* Better models could have unexpected failure modes such as bad performance on small objects, gender bias, sensitivity to low lighting conditions etc.

Behavioural testing

* Manual error analysis should be done to identify failure modes of the model.
* Tests should be added to quantify these failure modes.
* These tests would be problem, domain and dataset specific.
* Broadly classified into three types

○ Invariance test

○ Directional expectation test

○ Minimum functionality test

Source

# Invariance Test

* Apply label preserving perturbations to inputs and expect model predictions to remain the same.

○ Replace places names in sentiment analysis.

■ Examples:

* @AmericanAir thank you we got on a different flight to [ Chicago → Dallas ].
* @VirginAmerica I can’t lose my luggage, moving to [ Brazil → Turkey ] soon, ugh.

○ Typos in sentiment analysis. ■ Examples:

* The flight was great → The flght ws great.
* Why are we getting so lazy → Why are we gettnig so lazy.

# Invariance Test

● Apply label preserving perturbations to inputs and expect model predictions to remain the same.

○ Rotating images in object detection.



Image: Farmers ensuring they get the same alert for the same trap catch.

# Directional Expectation Test

* Apply perturbations to the inputs and expect labels to change in a certain way.

○ Keeping everything else the same, if house built up area is reduced, does the price increase?

○ Adding a more negative sentence to the end should not make outputs more positive.

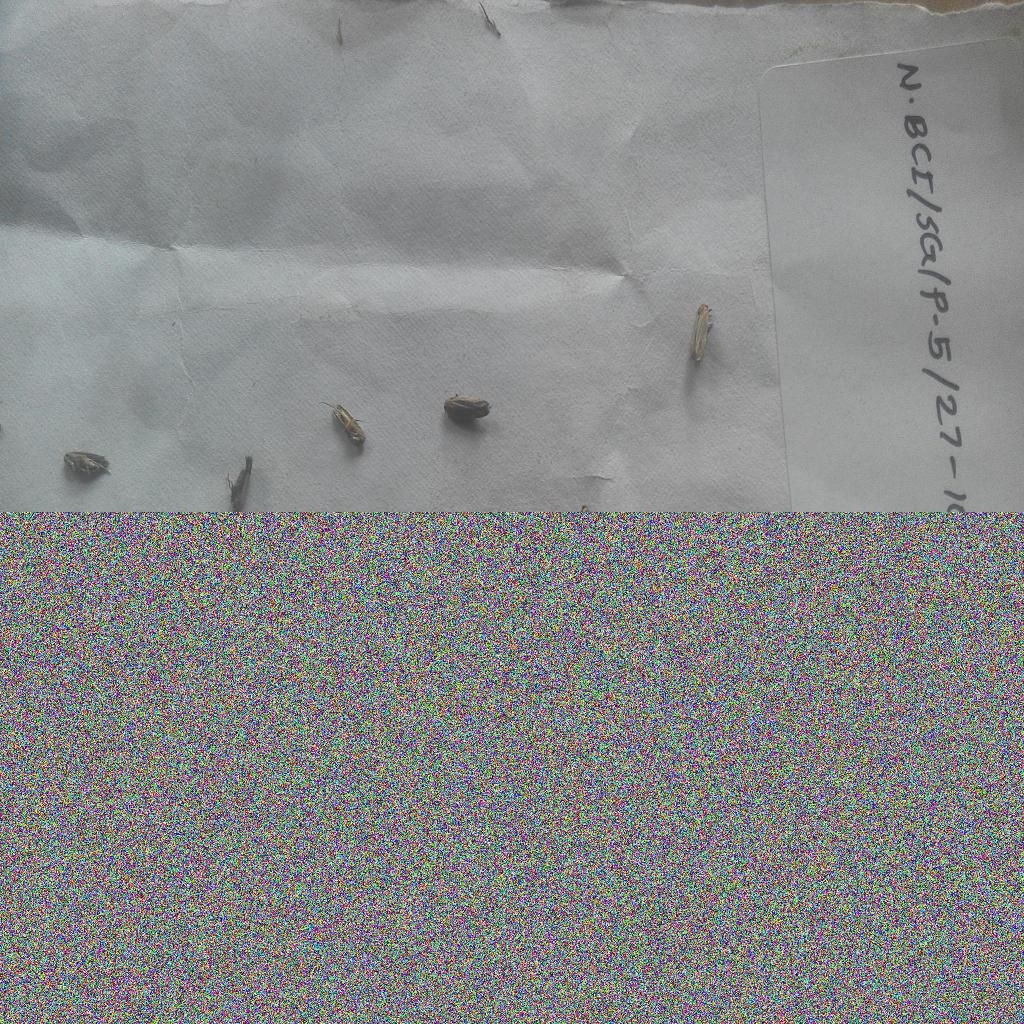
■ Examples:

* @USAirways your service sucks. → @USAirways your service sucks. You are lame.
* @JetBlue why won't YOU help them? → @JetBlue why won't YOU help them? I dread you.

# Directional Expectation Test

● Apply perturbations to the inputs and expect labels to change in a certain way.

○ Replacing lower half of the image, increase the count?



# Minimum Functionality Test

* Simple test cases designed to test a specific behaviour.
* Analogous to unit test where we test the model on a small part of the data.
* Failure modes discovered in error analysis can be tested with MFT.

○ Examples:

■ Testing on short, long sentences separately in sentiment analysis.

■ Testing on small, large objects in object detection.

■ Testing on each region’s samples separately.

■ Testing on simple sentences created with certain template in sentiment analysis:

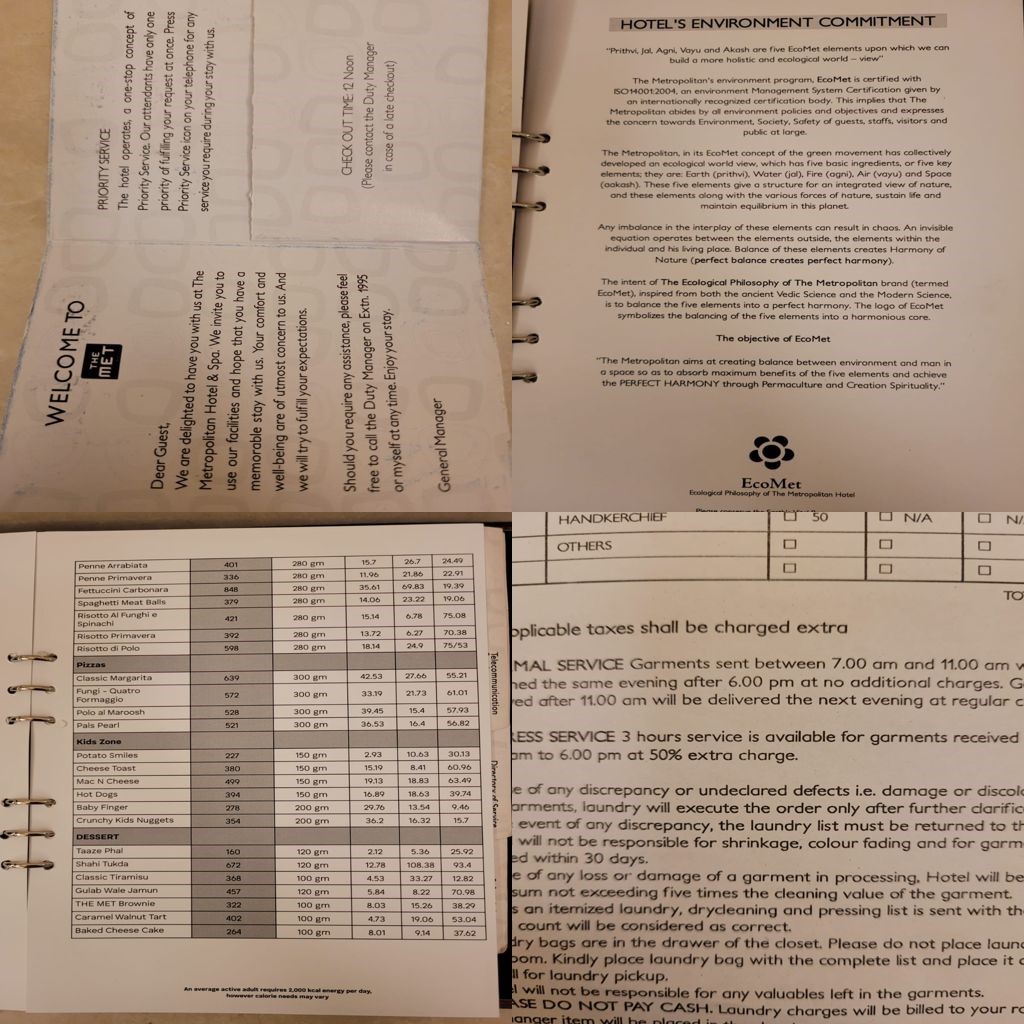
* Template: “I {negation} {positive verb} the {thing}”.
* Sentences: I can’t say I recommend the book, I didn’t love the flight etc.

# Minimum Functionality Test

○ Examples:

■ In pest management’s trap based solution, one of the failure modes is accepting ‘white printed paper’ as a valid trap image.

Trap Images Non-Trap Images Test case designed with failure mode images



Summary

* Pre-train test
  1. Unit test

○ Regression test

○ Integration test

* Post-train test
  1. Model evaluation on validation set

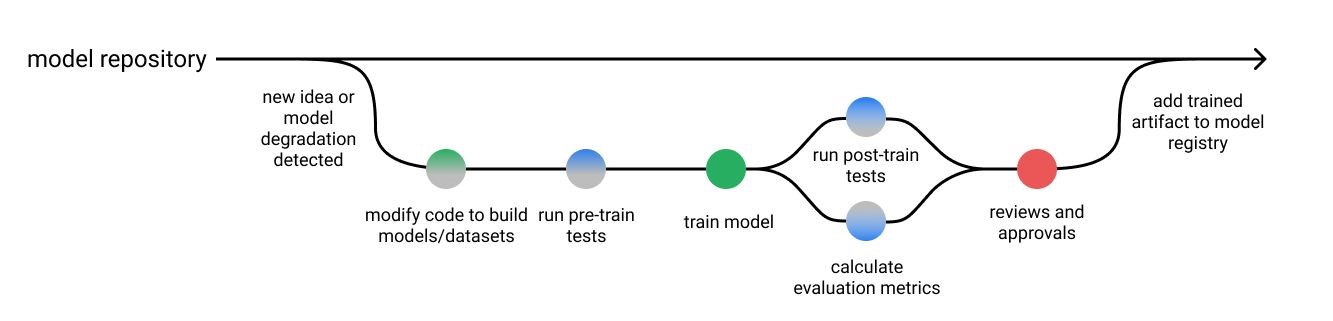
○ Behavioural testing

■ Invariance test

■ Directional expectation test

■ Minimum functionality test

# Final ML Pipeline



References

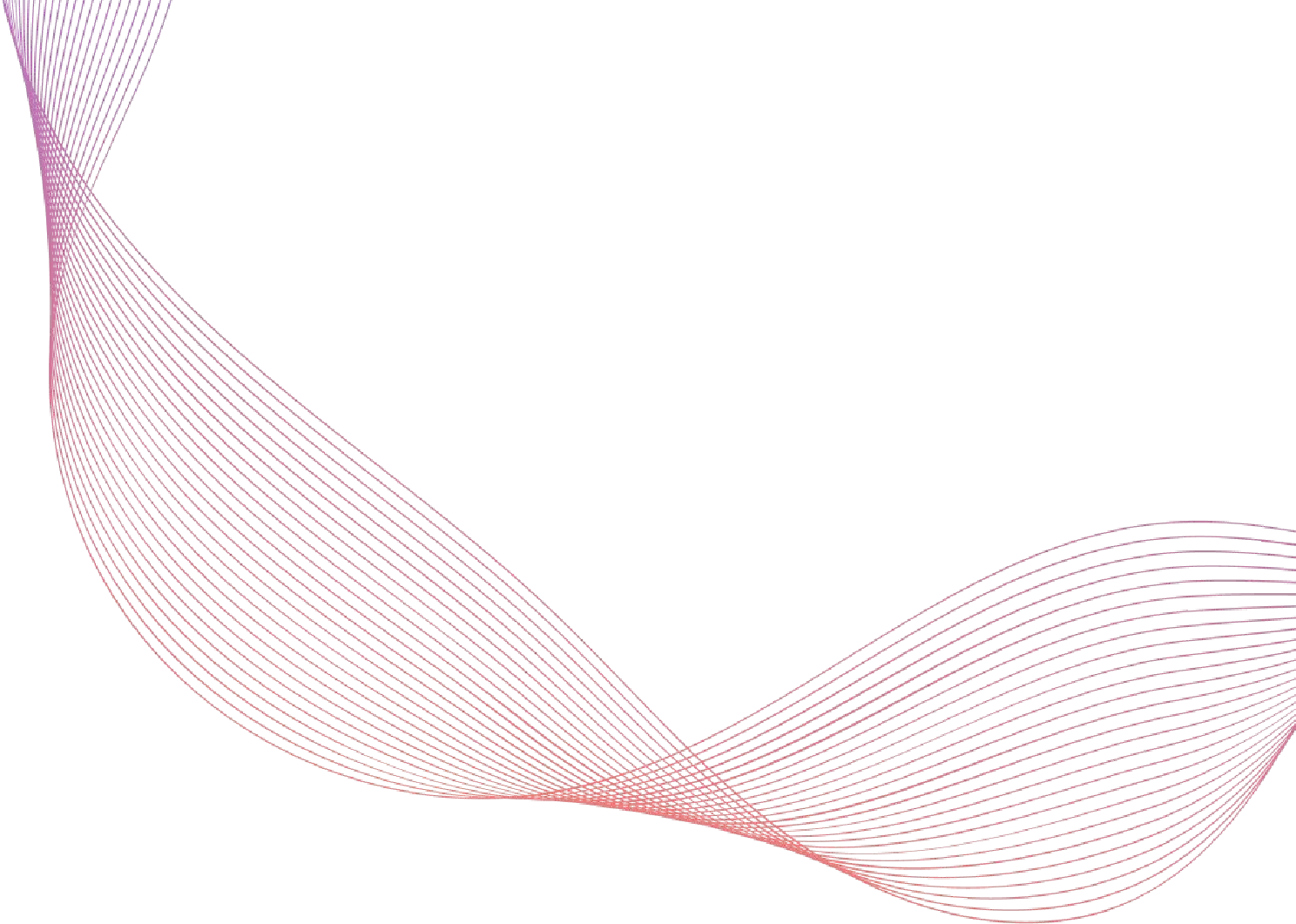
* Must read
  1. How to Trust Your Deep Learning Code (Writing Unit Tests)

○ Paper on Testing in NLP

* Nice to read
  1. Testing for ML Systems

○ How to Test ML Models

Effective ML Workshop-22



**Thank you!**

**www.wadhwaniai.org**

