TDD-ML: Test-Driven Development for ML Systems

Presenter: Rachel Brower-Sinning

Date: January *6*, 2022

FY23 MTP Line Proposal

Softw are Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Problem Statement

ML models often unexpectedly exhibit poor behavior when deployed in operational settings. Recent work has shown that underspecification of the model is a key reason for this decline, particularly in MLOps pipelines. (D'Amour et al., 2020)

We hypothesize that applying principles from software engineering's strategy of test driven development, namely the generation of test cases from requirements (and test data) prior to project development and then applying these tests during the model evaluation, can be used in conjunction with existing statistical tests to help solve the problem of underspecification in MLOps pipelines.

Project Objectives

What we aim to do in this project is develop a methodology and a tool for analysis of test data sets used in ML model development and acquisition processes. We will then integrate into a MLOps pipeline to support maintenance and expansion of the test sets in a production environment.

Specifically, we will develop a methodology and tool were:

1. Representative ML model developers, when using the tool in conjunction with software engineering TDD practices, will better specify the ML model, resulting in an 30% increase in model accuracy, in line with results of Afan, et al. 2022.
2. Use of the tool during ML model evaluation and acquisition will lead to the selection of an ML model that will perform at least 20% better in production, in line with the findings of Oakden-Rayner, et al. 2020.
3. Use of the tool, when integrated into a DoD MLOps pipeline, during ML model evaluation and acquisition will lead to the selection of an ML model that will not experience as large of a performance drop off when in production when compared to baseline.

State of the Art/Practice – Problems

Model underspecification is a known problem, that results in models being developed and deployed in MLOps pipelines that do not adequately address organizational need. Published work indicates that improved specification of the ML model can result in models that outperform their underspecified counterparts

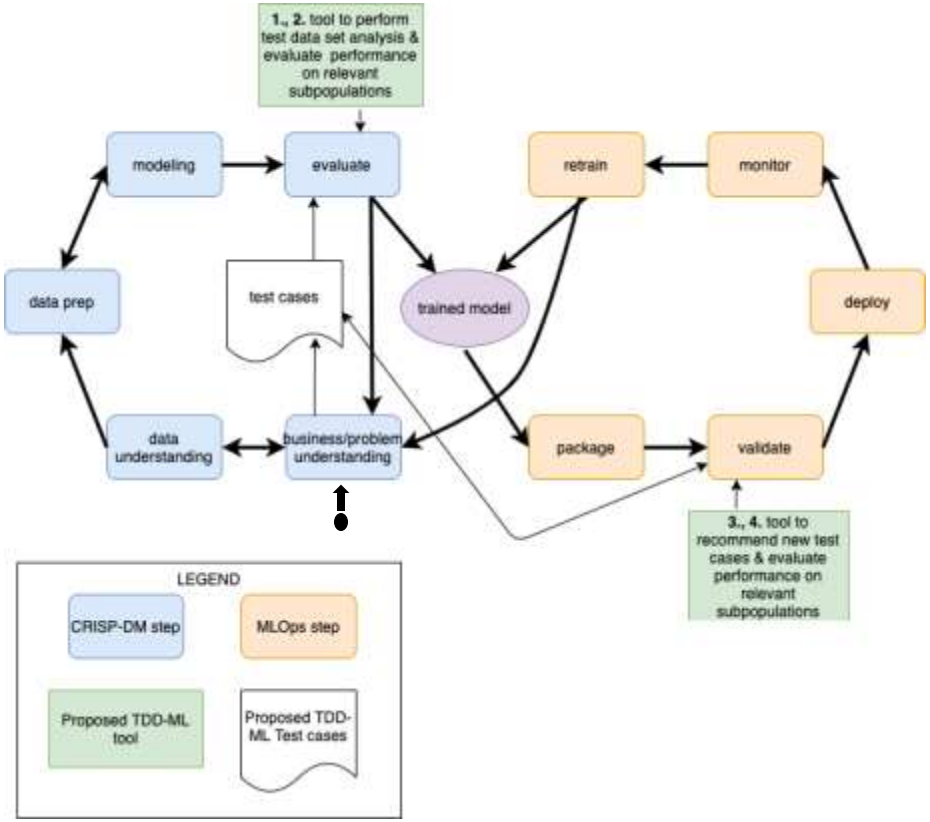
Ideally, a test data set allows the evaluator to determine how well an ML model will perform in production, and is independent from the training data. Current MLOps pipelines allow for retraining of the ML model, but do not generate a recommendations as to how to maintain the test data set over the lifespan of the model.

State of the Art/Practice – Part of a Possible Solution

Software engineering Test Driven Development (TDD) relies on using the organizational requirements to generate test cases *prior to* the software being developed.

Stratified Performance Evaluation and Shifted Performance Evaluations are techniques that are currently used as methods to understand how performance of the ML model is affected by the characteristics of the data set. Additionally, the can be employed add

Project Summary



Task 1: Develop a TDD-based methodology for test data set analysis during ML model evaluation

Task 2: Develop tool support for the TDD-based methodology during ML model evaluation

Task 3: Extend the TDD-based methodology into operations

Task 4: Extend the TDD-ML tool into operations

Task 1: Develop a TDD-based methodology for test data set analysis during ML model evaluation

Evaluating ML model performance on a single test set using a single metric can lead to am underspecified model that may perform poorly in operation. Additional model specification, through additional testing of organizationally relevant subpopulations, can work to further specific the ML model.

Goal: Develop a TDD-based methodology to analyze the test data set

Approach: Build the TDD-based methodology by evaluate existing techniques to:

1. Analyze the test set for user specified subpopulations and determine, using statistical power calculations, if the subpopulations are sufficient
2. Analyze model performances in each user specified subpopulation using Stratified Performance evaluation techniques (following procedures similar to Oaken-Rayner et al., 2019, D'Amour et al., 2020)
3. Generate a report that documents subpopulations, parameters used in the calculations, and the results

Validation: Validate the TDD-based methodology on existing data sets:

1. On publicly available image data (e.g. Stanford car data set) and text data (e.g.

Task 2: Develop tool support for the TDD-based methodology during ML model evaluation

Goal: Implement and integrate the TDD-based method developed in Task 1 into ML model evaluation workflows.

Approach: Build the tool support for TDD-based methodology (tool is named TDD-ML for simplification)

Validation: Validate the the new tool on existing data sets, with existing personnel:

1. On publicly available image data (e.g. Stanford car data set) and text data (e.g. CERT Coordination Center Vulnerability Data set)
2. On DoD data sets, by DoD personnel

Task 3: Extend the TDD-based methodology into operations

Goal: Extend the TDD-methodology developed and validated in Task 1 into an MLOps pipeline to support continuous update of tests in the operational environment

Approach: Extend TDD-based methodology into the operational environment by:

1. Splitting the operational data into train and test data sets
2. Identifying operational test cases that align with the previously specified, organizationally relevant subpopulations
3. Identifying operational test cases that should be labeled or reviewed (following procedures similar to those in Groce et al., 2014)

Validation: Validate the TDD-based methodology on existing data sets:

1. On publicly available image data (e.g. Stanford car data set) and text data (e.g. CERT Coordination Center Vulnerability Data set)
2. On DoD data sets

Task 4: Extend the TDD-ML tool into operations

Goal: Implement and integrate the TDD-based method developed in Task 3 into MLOps pipelined.

Approach: Extend the TDD-ML tool from Task 2 into MLOps pipelines by adding in the extended methodology from Task 3.

Validation: Validate the the new tool on existing data sets, with existing personnel:

1. On publicly available image data (e.g. Stanford car data set) and text data (e.g. CERT Coordination Center Vulnerability Data set)
2. On DoD data sets, by DoD personnel

Questions?