An Introduction to Honest_ML - Why Confidence Intervals Matter

By Eric Schles

Contact

Twitter - @EricSchles

Linkedin - https://linkedin.com/in/ericschles

What are statistical models?

- Distributions & Data
- Descriptive Statistics as a first approximation
- Honest-ML introduction
  - The problem
  - API examples
- Linear Regression
  - Introduction to linear regression
  - Linear Regression in Honest-ML
- Tree Based Models
  - Introduction to Trees, Gradient boosting & Random Forests
  - Tree Based Models in Honest-ML
- Neural Networks
  - Introduction to Neural Networks
  - Neural Networks in Honest-ML
- Extras
  - Genetic Algorithms & Honest-ML
  - Gradient Boosting & Honest-ML
  - Gradient Boosted Genetic Algorithms

Distributions & Data

Introduction to distributions and Data:

https://github.com/EricSchles/datascience_book/blob/master/1/Statistical%20Tests%20-%2001.ipynb

Claim

In fact, descriptive statistics are so powerful you can simulate data knowing only the descriptive statistics and the underlying distribution that are faithful representations of the data generating process:

https://github.com/EricSchles/honest_ml/blob/main/conference_talks/Specifying%20multiple%20parameters.ipynb

Goals Of Statistical Modeling

- Representation - how well does the model represent the data you have available?
- Generalization - how does the model perform on out of sample data?

Representativeness of a sample?

Considerations:

- Underlying distribution
- Sampling process
- Size of population (approximately)

Problems with Statistical models

- The issue with point statistics in metrics
  - Random seed problems
    - Side effects
  - Model specific issues

Evidence:

https://github.com/EricSchles/randomness_experiments/blob/master/scikit-learn-experiments-breast_cancer.ipynb

- Varying seed in train test split
- Varying seed but keeping it constant in train test split

My Solution - Honest_ml

- Library motivation
    - Problems this solves
- Introductory example
- Why everything should have confidence intervals

demo

https://github.com/EricSchles/honest_ml/blob/main/examples/simple_example.ipynb

https://github.com/EricSchles/honest_ml/blob/main/examples/pipeline_example.ipynb

https://github.com/EricSchles/honest_ml/blob/main/examples/confusion_matrix_testing.ipynb

Linear Regression & Logistic Regression

https://github.com/EricSchles/datascience_book/blob/master/2/An%20Introduction%20to%20Regression%20-%2003.ipynb

- MLE
- Linear regression picture

https://github.com/EricSchles/datascience_book/blob/master/3/An%20Introduction%20to%20Classification%20-%2004.ipynb

- Model implementation
- Interpreation

Linear Regression & Logistic Regression in Honest-ML

One of the ways to be confident about your model is by looking at the p-values on your coefficients. This is unfortunately unique to linear regression (in implementation in Python Libraries), but can be done for all sorts of models. With Honest-ML you can be confident, not only of your coefficients, but your predictions!!

https://github.com/EricSchles/honest_ml/blob/main/conference_talks/linear%20regression%20%26%20Logistic%20Regression%20in%20Honest-ML.ipynb

Trees

Tree based models answer a different question - What if you just forget about trying to figure out the underlying distribution and just put data into buckets. Does this work?

https://github.com/EricSchles/datascience_book/blob/master/4/An%20Introduction%20To%20Information%20Theory%20-%2005.ipynb

- Algorithm
- Trees picture

(Yes, sometimes)

Trees & Honest-ML

Trees are fairly interpretable models, yet they typically do not come with p-values attached. And yet, by visualizing the tree, you can usually get a sense of what the tree is ‘thinking’ by looking at how it splits the data into buckets. Unfortunately, this doesn’t really give much sense of how useful each feature is, since the trees can be quiet spread out, with many branches and leaves. One way around this is to look at feature importances, which scikit-learn implements. The reason to look at these is to get a sense of the ‘magnitude’ of each feature, similarly to how the size of a coefficient effects it’s importance in a linear model. We can use honest-ml to get a sense of how much these feature importances shift from model to model:

https://github.com/EricSchles/honest_ml/blob/main/conference_talks/Decision%20Trees%20%26%20Honest%20ML.ipynb

Neural Networks

https://github.com/EricSchles/datascience_book/blob/master/5/An%20Introduction%20to%20Neural%20Networks%20-%2007.ipynb

Vanilla Neural Networks:

Neural Networks answer a similar question to linear regression, except with backpropagation, which is model stacking in the forward pass and parameter update in the backpass. Additionally a non-linearity is (often) added on top of the model at each step.

Structure aware Neural Networks:

- Convolutional Neural Networks - add a learned kernel of ‘correlations’ (sort of correlations because a convolution is a cross correlation flipped about the y axis)
- LSTM Recurrent Neural Networks - add a notion of time dependence in the data, account for state and then prune to the ‘best’ representation (based on the metric)
- Transforms - sample the data _very_ intelligently during training

Neural Networks and Honest-ML

No machine learning framework would be complete without Neural Networks, as they are the pinnacle of many machine learning problems:

https://github.com/EricSchles/honest_ml/blob/main/conference_talks/Neural%20Network%20in%20Honest-ML.ipynb

Extras

Stochastic Bagging

https://github.com/EricSchles/honest_ml/blob/main/honest_ml/trainer/trainer.py#L264 – regression

https://github.com/EricSchles/honest_ml/blob/main/honest_ml/trainer/trainer.py#L315 – classification

Stochastic Gradient Boosting

https://github.com/EricSchles/honest_ml/blob/main/honest_ml/trainer/trainer.py#L454 – regression

Stochastic Genetic Algorithm

https://github.com/EricSchles/honest_ml/blob/main/honest_ml/trainer/trainer.py#L536 – both

Stochastic Genetic Boosting Algorithm

https://github.com/EricSchles/honest_ml/blob/main/honest_ml/trainer/trainer.py#L927 – WIP