Brown Bag: Where to Set the Threshold in Predictive Models

Data Science Team

???Date???

Andrew Wheeler, PhD

andrew.wheeler@hms.com

Overview of the Problem

- Models give a predicted probability or a predicted value
- Predictions do not intrinsically translate to *what* you should do with the information
- How to decide *when to act* given those predictions

Example 1

- Predicted probability *of rain* is 5%
  - would you bring your umbrella?

- Predicted probability *of used car not working* is 5%
  - Would you buy that car?

- What is the difference? **Costs/Benefits** of false negative

Example 2

- Utility of getting fast food for dinner
  - Benefits – fast, cheap
  - Costs – unhealthy, not as good as home cooked meals

- Costs/Benefits don’t change, so why do you get fast food sometimes and not others?
  - Kids have event after school, more likely to get fast food

- What is the difference? **Constraints**

Cost/Benefits & Constraints

- Better Example: When to audit a claim to determine fraud, based on predictive model probability

- Costs = time it takes to review claim
- Benefits = % of claim capture
- Constraints -> whether you have someone to review the claim

Hypothetical Example

- Score 1,000 claims for probability it is fraud
  - Accuracy of model is 90% (True Positive Rate)
  - False Positive Rate is 10%
  - Prevalence of fraud is ~10%

- Benefits of finding a fraud case is $10
- Costs to review a case is $2
- How much money are we making?

Hypothetical Example

	No Fraud	Fraud
Predicted No Fraud	810	10
Predicted Fraud	90	90

Prevalence: 0.1
Accuracy: 0.9
False Positive Rate: 0.1

Benefits

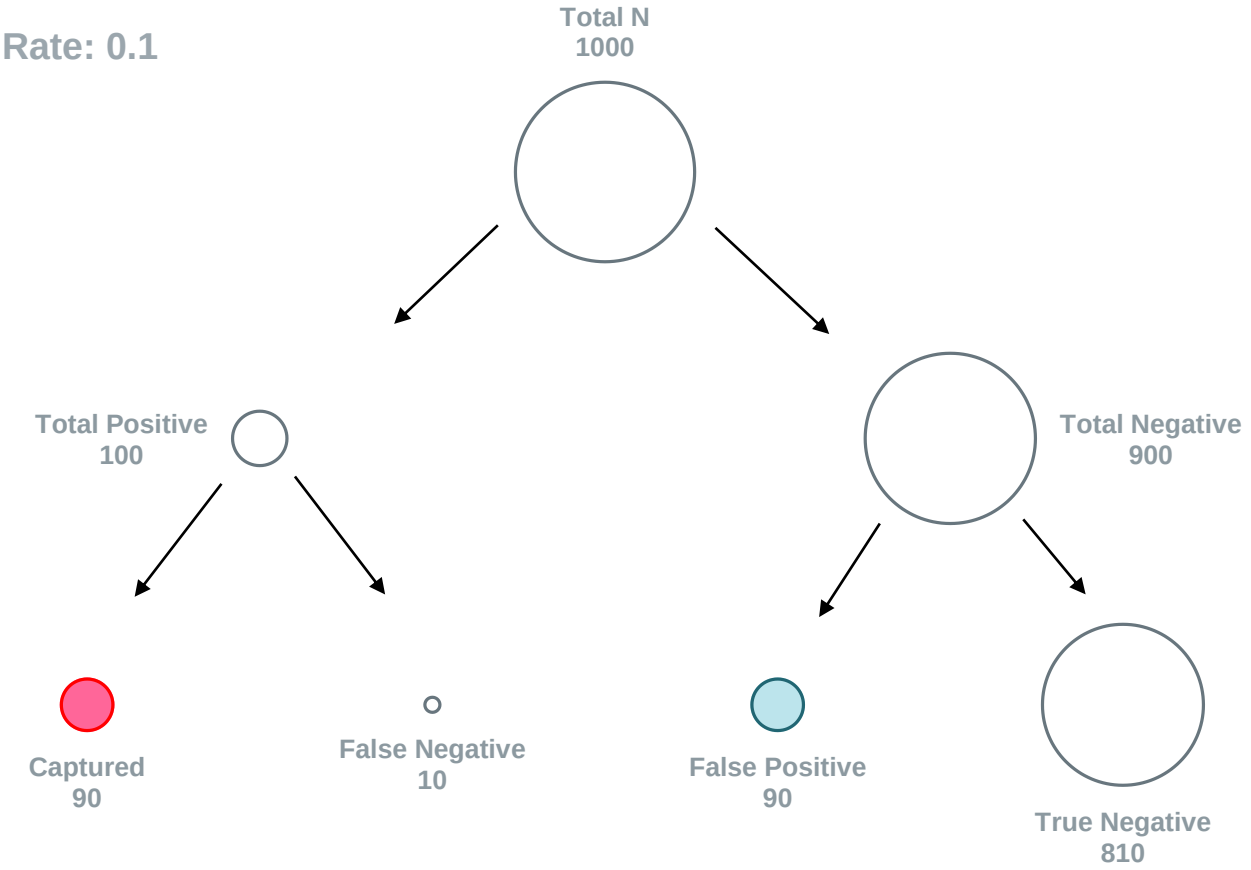
90*10=900

Costs

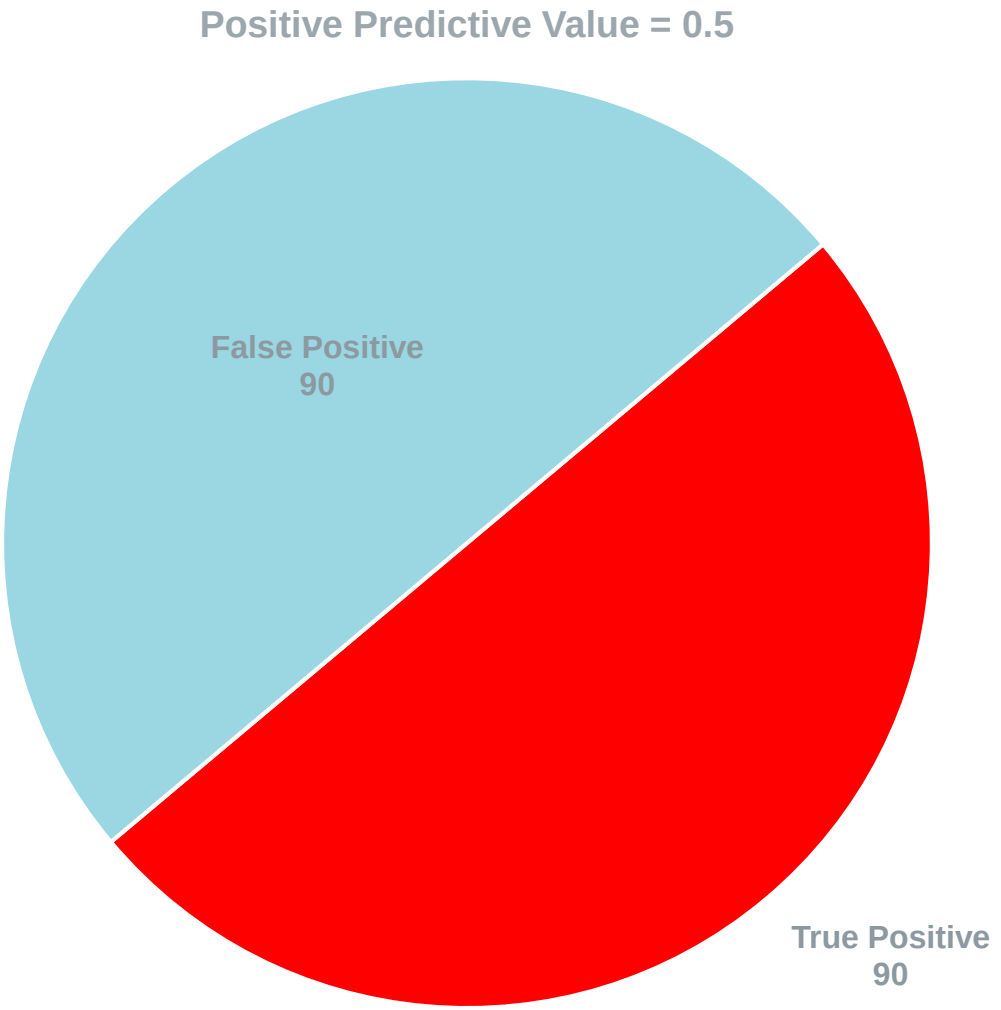
180*2=360

Net

900-360=540



Hypothetical Example



Generalizing to different thresholds

$$\text{Net} = [N \cdot \text{Prev} \cdot \text{TPR} \cdot (B - C)] + [N \cdot (1 - \text{Prev}) \cdot \text{FPR} \cdot C]$$

- *N* is the total number of cases
- Prev is the prevalence of true cases
- TPR is the true positive rate
- FPR is the false positive rate
- *B* is the benefits, and *C* is the cost

- Can factor out N to make it a per-record utility estimate
- Can include false negatives as additional revenue that we lose out on

Simple Python Function

```python
#May want to expand to include the full cost matrix (true negatives/false negatives)
def util_cut_point(tp,fp,th,pr,be,co,curve=False):
    """
    Returns the optimal cut point given costs/benefits and prevalence population
    estimates

    Parameters
    ----------

    # Take these straight from sklearn.metrics.roc_curve()
    tp : numpy array of true positive rates
    fp : numpy array of false positive rates
    th : numpy array of thresholds based on prediction model

    # These you need to figure out for the local problem
    pr : scaler (float) of prevalence estimate in pop
    be : scaler (float/int) benefit of identifying true positive case (should be positive)
    co : scaler (float/int) cost of false positive (should be negative)

    # Optional
    curve : Boolean whether to return entire utility curve, default False

    Returns
    -------
    cut_point : float of the optimal cut point
    util_scores : optional numpy array of utility curve along entire trp/fpr/thresholds
    """
    pr_min = 1-pr
    util_scores = pr*tp*be + pr_min*fp*co
    cut_point = th[np.argmax(util_scores)]
    #return the full utility curve if curve=True
    if curve:
        return cut_point, util_scores
    else:
        return cut_point
```

Sensitivity Analysis with Different Costs/Benefits

Cost	Benefit	CutPoint	TN	FP	FN	TP	Utility
-1	4	0.093844	62	6	0	120	474
-1	9	0.093844	62	6	0	120	1074
-1	19	0.093844	62	6	0	120	2274
-2	3	0.093844	62	6	0	120	348
-2	8	0.093844	62	6	0	120	948
-2	18	0.093844	62	6	0	120	2148
-3	2	0.326861	63	5	1	119	223
-3	7	0.093844	62	6	0	120	822
-3	17	0.093844	62	6	0	120	2022

Notes

- Threshold may vary depending on benefits of case
  - Cases with higher $$ claims should have lower threshold
- For very rare outcomes, need very high benefit/cost ratio (will always have many false positives to only a few true positives)
- Other uses besides setting threshold
  - Can pre-emptively say how much total audit may cost (e.g. to get a true positive rate of 95% need to conduct X audits)
  - Monitoring of audits (e.g. should identify around X% of positive cases)

Potential Issues

- Prevalence estimates may vary sample to sample
  - Case/control designs (or oversampling fraud cases) will make prevalence seem higher than it really is (going from small sample to large database)
  - Solutions? Bounds on prevalence estimates, worst case analysis
- Costs & Benefits may not be a single number for any claim
  - Solutions? Model these as well, then monte carlo/convolutions to get range of cost/benefit estimates
- Using threshold estimated on test data may be too optimistic
  - Solutions? K-fold cross-validation in test sample to estimate error

Applying in the Future

- Need estimates of costs, benefits, and prevalence for each project
  - This is model agnostic, so getting a good model is a separate issue
  - AUC is still a good first hand metric (shows it does well under many thresholds)

- Did not consider constraints here. Will illustrate in future how to optimally distribute resources under different constraints (linear programming)
  - Even if there are hard constraints (e.g. number of cases that can be reviewed by a human), we can still estimate cost/benefit

Links to Resources

- Jupyter notebook giving example use, https://github.com/hmsholdings/data-science-utils/blob/master/where_cut_point.ipynb

Future Topics?

- My ideas:
  - Setting constraints with *linear programming*
  - Overview of *interpretable machine learning summaries*
  - *Survival analysis* with time to event data
  - Hierarchical models

Send me topics *you* want to cover

Questions?

Brown Bag: Where to Set the Threshold in Predictive Models

Data Science Team

???Date???

Andrew Wheeler, PhD

andrew.wheeler@hms.com