Commentary on the Performance Measure for the 2022 MEBDI Machine Learning Competition

Robert Winslow

March 28, 2022

Abstract

The measure used for ranking algorithms in the the 2022 MEBDI Machine Learning Competition fails in its stated goal of equally weighting predictions of employment and unemployment. I demonstrate the problem with an example, compare alternate measures, and give recommendations for how the competition description should be changed.

The objective of the 2022 MEBDI Machine Learning Competition is to design an algorithm which predicts whether individuals will be unemployed in the following year. The competition ranks submitted algorithms according to the following Goodness of fit (GF) measure:[1]

$$\text{GF} = \frac{1}{1+\bar{y}} \cdot \left[\frac{\sum_{i\in T}\ \hat{y}_{i,t+1} \cdot y_{i,t+1}}{\sum_{i\in T} y_{i,t+1}}\right] + \frac{\bar{y}}{1+\bar{y}} \cdot \left[\frac{\sum_{i\in T}\ (1-\hat{y}_{i,t+1}) \cdot (1-y_{i,t+1})}{\sum_{i\in T} (1-y_{i,t+1})}\right]$$

This formula is chosen, according to the contest description, to ensure that the measure "puts equal weight on predicting employment and unemployment". However, the given GF measure fails to do so; it puts *far* more weight on correctly predicting unemployment. This misweighting is so extreme that rather silly strategies are expected to outperform more sophisticated approaches.

If the goal is to place equal weight on predicting each category, I suggest using the following Balanced Accuracy (BAcc) measure instead:

$$\text{BAcc} = \frac{1}{2} \cdot \left[\frac{\sum_{i\in T}\ \hat{y}_{i,t+1} \cdot y_{i,t+1}}{\sum_{i\in T} y_{i,t+1}}\right] + \frac{1}{2} \cdot \left[\frac{\sum_{i\in T}\ (1-\hat{y}_{i,t+1}) \cdot (1-y_{i,t+1})}{\sum_{i\in T} (1-y_{i,t+1})}\right]$$

In this document, I start by presenting a degenerate algorithm which nonetheless performs quite well on the given GF measure. Next, I compare the given GF measure to alternate measures by which classfication algorithms could be evaluated. And finally, I discuss why BAcc is a more balanced measure of an algorithm's performance in this competition.

[1]See the appendix for a list of notation explanations.

1 The Degenerate "Yes-Man" Algorithm

I have designed a simple algorithm which performs wonderfully in this competition. It simply predicts that *everyone* will be unemployed. The model is specified very intuitively:

$$\hat{y}_{i,t+1} = 1$$

And the source-code is quite simple, written in Python 3 and requiring no external libraries:

```
print("Everyone in this sample will be unemployed next period. Trust me.")
```

It performs quite well on the Goodness of fit measure, earning a score of 0.9511, regardless of the test sample it is evaluated on:

$$\begin{aligned}
\text{GF} &= \frac{1}{1+\bar{y}} \cdot \left[\frac{\sum_{i\in T} \ \hat{y}_{i,t+1} \cdot y_{i,t+1}}{\sum_{i\in T} y_{i,t+1}}\right] + \frac{\bar{y}}{1+\bar{y}} \cdot \left[\frac{\sum_{i\in T} \ (1-\hat{y}_{i,t+1}) \cdot (1-y_{i,t+1})}{\sum_{i\in T} (1-y_{i,t+1})}\right] \\
&= \frac{1}{1+\bar{y}} \cdot \left[\frac{\sum_{i\in T} \ 1 \cdot y_{i,t+1}}{\sum_{i\in T} y_{i,t+1}}\right] + \frac{\bar{y}}{1+\bar{y}} \cdot \left[\frac{\sum_{i\in T} \ 0 \cdot (1-y_{i,t+1})}{\sum_{i\in T} (1-y_{i,t+1})}\right] \\
&= \frac{1}{1+\bar{y}} \cdot \left[\frac{\sum_{i\in T} \ y_{i,t+1}}{\sum_{i\in T} y_{i,t+1}}\right] + \frac{\bar{y}}{1+\bar{y}} \cdot \left[\frac{0}{\sum_{i\in T} (1-y_{i,t+1})}\right] \\
&= \frac{1}{1+\bar{y}} \approx 0.9511
\end{aligned}$$

A perfect classifier only returns a score of 1.0, so this will be quite the difficult algorithm to beat.

2 Comparing Performance Measures for Classifiers

Why does this degenerate algorithm perform so well? In the original data, unemployment is much less common than employment. The GF measure attempts to compensate for this, but ends up overcompensating.

2.1 Accuracy

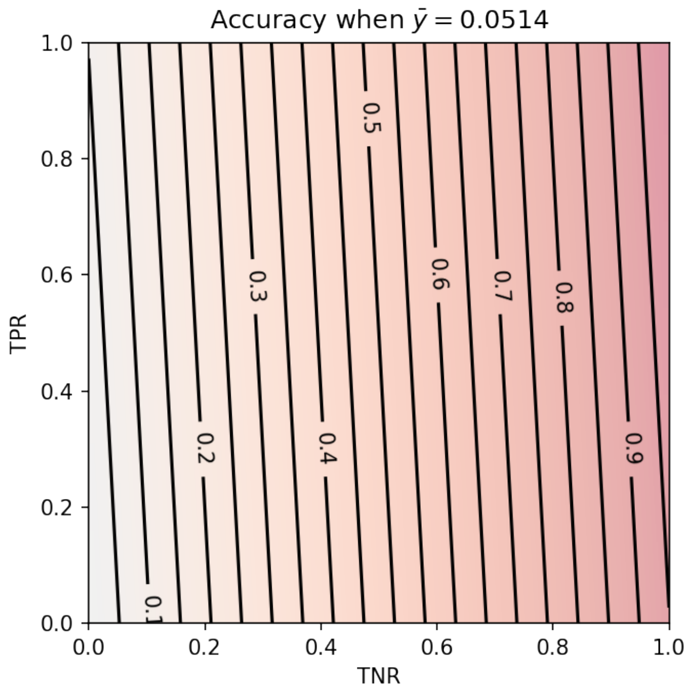
The naive approach to evaluating a classification algorithm is to simply measure its accuracy, defined as the portion of individuals for whom the algorithm makes a correct prediction:

$$\begin{aligned}\text{ACCURACY} &\equiv \frac{\sum_{i\in T}(\hat{y}_{i,t+1}\cdot y_{i,t+1}) + \sum_{i\in T}((1-\hat{y}_{i,t+1})\cdot(1-y_{i,t+1}))}{\sum_{i\in T}1} \\ &= \frac{\text{TP}+\text{TN}}{|\text{T}|} \\ &= \frac{\sum_{i\in T} y_{i,t+1}}{|\text{T}|}\cdot\text{TPR} + \frac{\sum_{i\in T}(1-y_{i,t+1})}{|\text{T}|}\cdot\text{TNR}\end{aligned}$$

If we assume for simplicity that the portion of to-be-unemployed individuals is roughly the same in the testing and training samples (that is, that $\frac{\sum_{i\in T} y_{i,t+1}}{|\text{T}|} \approx \bar{y}$), then we can rewrite the formula for accuracy like so:

$$\begin{aligned}\text{ACCURACY} &\approx \bar{y}\cdot\text{TPR} + (1-\bar{y})\cdot\text{TNR} \\ &\approx 0.0514\cdot\text{TPR} + 0.9486\cdot\text{TNR}\end{aligned}$$

As mentioned in the contest description, the data used for this contest is unbalanced, with only a small fraction $\bar{y}$ being unemployed in the following period. As such, it is far easier to boost the accuracy of a classification algorithm by correctly predicting that people won't be unemployed (increasing the TNR) than by correctly predicting that they will (incrasing the TPR).



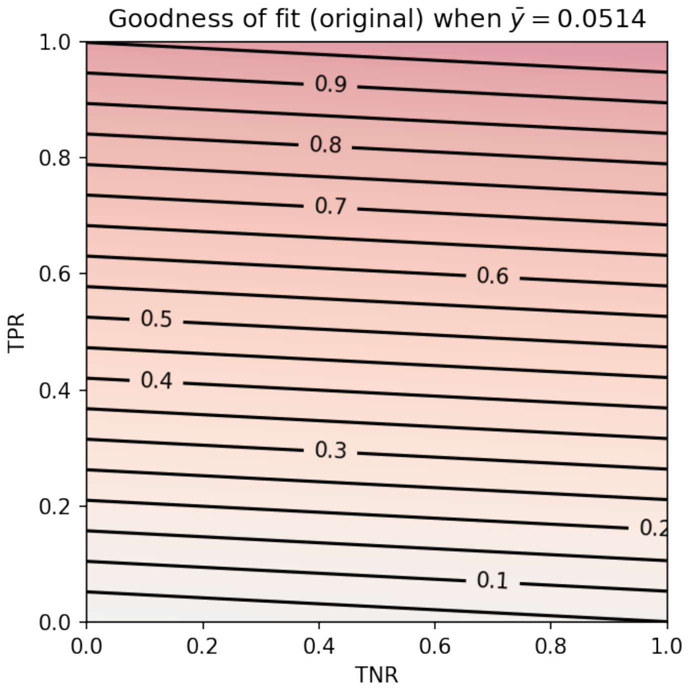
The degenerate algorithm of always predicting that no one will be unemployed ($\text{TNR} = 1, \text{TPR} = 0$) manages to score quite well on accuracy, with a score of $1-\bar{y} \approx 0.9486$ in the training sample. To improve on this degenerate case would require some signal in the data which predicts future unemployment with greater than 95% confidence. It seems dubious that such a signal can be found in CPS data. And in fact, when I fit a simple logistic regression with a threshold of $\frac{1}{2}$ to a subset of the training data for this competition, my regression did essentially converge to this degenerate all-zeros prediction.

2.2 Goodness of Fit

If we want a more compelling classification scheme, it seems prudent to increase the weight which is placed on accurate predictions of unemployment. The competition's GF measure does so, but to too great of an extent.

$$\begin{aligned}
\text{GF} &\equiv \frac{1}{1+\bar{y}} \cdot \left[\frac{\sum_{i \in T} \ \hat{y}_{i,t+1} \cdot y_{i,t+1}}{\sum_{i \in T} y_{i,t+1}} \right] + \frac{\bar{y}}{1+\bar{y}} \cdot \left[\frac{\sum_{i \in T} \ (1-\hat{y}_{i,t+1}) \cdot (1-y_{i,t+1})}{\sum_{i \in T} (1-y_{i,t+1})} \right] \\
&= \frac{1}{1+\bar{y}} \cdot \text{TPR} + \frac{\bar{y}}{1+\bar{y}} \cdot \text{TNR} \\
&\approx 0.9511 \cdot \text{TPR} + 0.0489 \cdot \text{TNR}
\end{aligned}$$

Whereas Accuracy places nearly twenty times as much weight on true predictions of non-unemployment, this Goodness of Fit metric makes the opposite mistake, placing nearly twenty times as much weight on true predictions of unemployment. And to beat the strategy would require finding some incredibly strong signal of future employment.



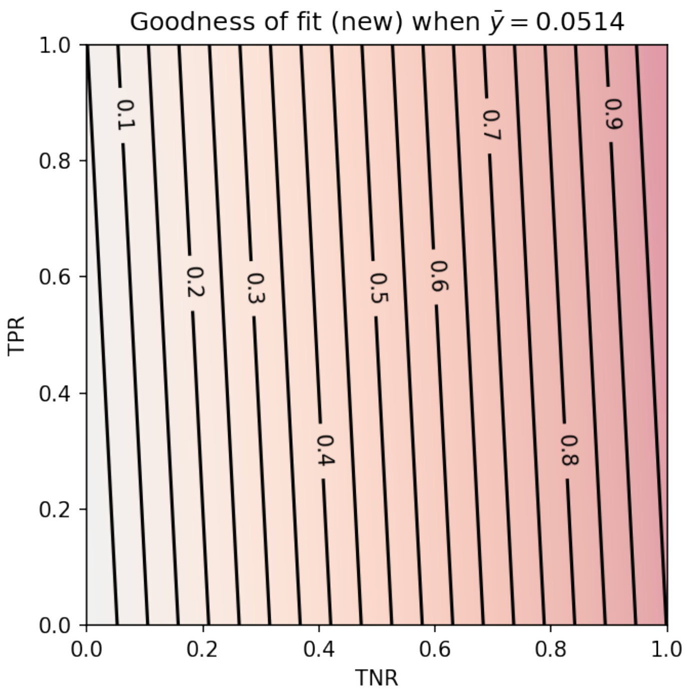
My original plan for this competition, which I will use if the success criterion is not changed, was to start with the degenerate strategy described in Section 1 above, use decision trees to try to find such a strong signal of future non-unemployment, and deviate from the degenerate strategy only for those rare individuals who provide that strong signal. But I don't have great confidence that such a thing can be found in CPS data.

2.3 Goodness of Fit, version 2

Very temporarily, the Goodness of Fit measure was updated to swap the weights on TPR and TNR:

$$\begin{aligned}
\mathrm{GF}_{v2} &\equiv \frac{\bar{y}}{1+\bar{y}} \cdot \left[\frac{\sum_{i\in T}\ \hat{y}_{i,t+1}\cdot y_{i,t+1}}{\sum_{i\in T} y_{i,t+1}}\right] + \frac{1}{1+\bar{y}} \cdot \left[\frac{\sum_{i\in T}\ (1-\hat{y}_{i,t+1})\cdot(1-y_{i,t+1})}{\sum_{i\in T}(1-y_{i,t+1})}\right] \\
&= \frac{\bar{y}}{1+\bar{y}} \cdot \mathrm{TPR} + \frac{1}{1+\bar{y}} \cdot \mathrm{TNR} \\
&\approx 0.0489 \cdot \mathrm{TPR} + 0.9511 \cdot \mathrm{TNR}
\end{aligned}$$

But the end result is a measure which is qualitatively similar to Accuracy. Worse even, for if the training and test samples have similar rates of future unemployment, then this measure actually puts even *more* weight on true predictions of non-unemployment than the Accuracy measure does. As such, the degenerate *all-zeros* classifier would be expected to perform even better under this measure than it would under the Accuracy measure.

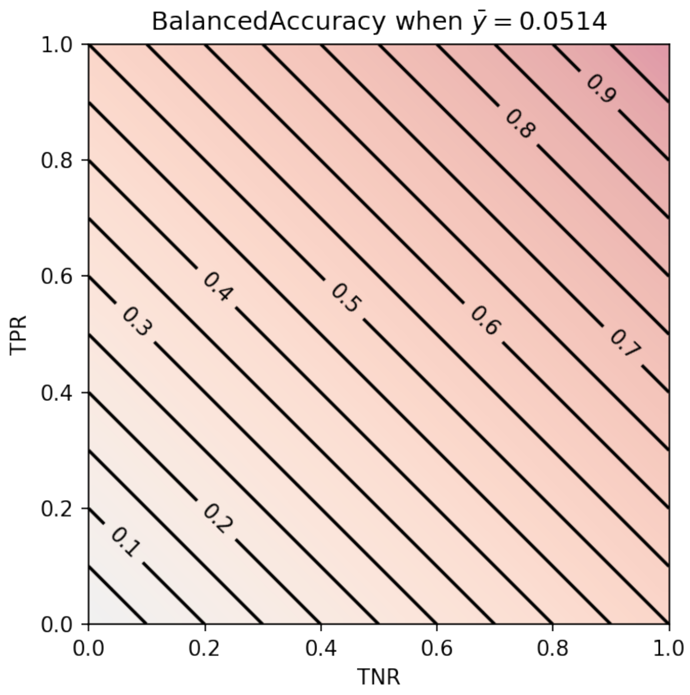


2.4 Balanced Accuracy and Informedness

The simplest way to fix the Goodness of Fit Measure would simply be to remove the coefficients $\frac{1}{1+\bar{y}}$ and $\frac{\bar{y}}{1+\bar{y}}$; and to instead *equally weight* the rate at which unemployment and non-unemployment are correctly predicted. This approach gives us what is sometimes called the "Balanced Accuracy" of the classifier:

$$\begin{aligned}\text{BAcc} &\equiv \frac{1}{2}\cdot\left[\frac{\sum_{i\in T}\ \hat{y}_{i,t+1}\cdot y_{i,t+1}}{\sum_{i\in T} y_{i,t+1}}\right] + \frac{1}{2}\cdot\left[\frac{\sum_{i\in T}\ (1-\hat{y}_{i,t+1})\cdot(1-y_{i,t+1})}{\sum_{i\in T}(1-y_{i,t+1})}\right]\\ &= \frac{1}{2}\cdot\text{TPR} + \frac{1}{2}\cdot\text{TNR}\end{aligned}$$

This measure no longer allows for a degenerate strategy to reign supreme.



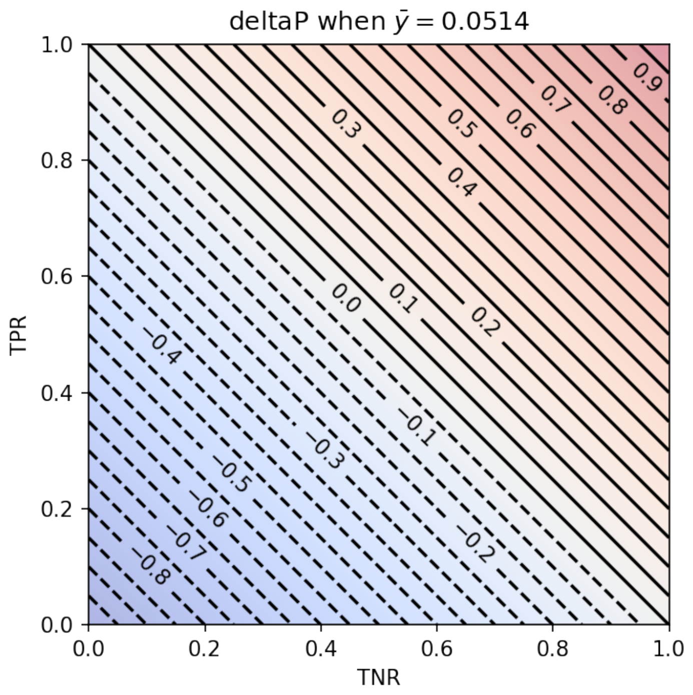
Both the *always-zero* algorithm and the *always-one* algorithm will earn a BAcc score of 0.5, and this measure has desirable properties:

- As with the GF measure, the scores for BAcc range from 0 to 1, with a perfect classifier attaining the upper bound.
- *Any* degenerate classifier, meaning any classifier which ignores the data when making its predictions, will earn the same expected score of BAcc=0.5

A very similar measure can be found in Informedness", also called "deltaP", which is defined as follows:

$$\begin{aligned}\text{BAcc} &\equiv \left[\frac{\sum_{i\in T}\ \hat{y}_{i,t+1}\cdot y_{i,t+1}}{\sum_{i\in T} y_{i,t+1}}\right] + \left[\frac{\sum_{i\in T}\ (1-\hat{y}_{i,t+1})\cdot(1-y_{i,t+1})}{\sum_{i\in T}(1-y_{i,t+1})}\right] - 1\\ &= \text{TPR} + \text{TNR} - 1\end{aligned}$$

Informedness is an affine transformation of Balanced Accuracy, and for the sake of ranking algorithms in this competition, the two are essentially equivalent. But the Informedness measure assigns a score of 0 to degenerate classifiers and a score of -1 to a classifier which somehow manages to always make the wrong prediction.



This reflects how the output of degenerate classifier gives us zero information. And how any classifier that scores below Informedness=0 (BAcc=0.5) actually *misinforms* us, providing worse predictions than a blind guess. (If your classifier has both TPR and TNR close to zero, then either you are extraordinarily unlucky, or you've accidentally inverted the labels on your training data.)

3 Inside the Algorithm - Why Does Balanced Accuracy fix things?

If the above graphs don't convince you of the merits of the BAcc measure, then consider the following thought experiment.

Imagine *you* are given the job of classifying individuals into two categories: employment and unemployment, 0 and 1. Little balls come rolling down the line. Each ball has information about a person written on it, and your job is to put each ball into either the employment bucket or the unemployment bucket. You also know that the balls meant to go into the employment bucket are 19 times as common as the balls meant for the unemployment bucket.

How should you go about this task? Well, it depends on how you are rewarded for your work.

3.1 Classifying when you are rewarded for accuracy.

Let's start by supposing that you are rewarded 1 dollar for each ball that you correctly sort. As such, you can maximize your earnings by maximizing your accuracy.

Your prior for whether a ball belongs in the unemployment bucket is only $\bar{p} \equiv \frac{1}{20}$, so if the balls don't have any useful information written on them, your best best is to just chuck everything into the employment bucket and call it a day.

On the other hand, suppose that there *is* useful information written on each ball. After inspecting a ball, your posterior beliefs are that there is a $\hat{p}$ chance that it belongs in the unemployment bucket, and a $1-\hat{p}$ chance it belongs in the employment bucket. If you toss it into the unemployment bucket, your expected earnings are $\hat{p}\cdot 1+(1-\hat{p})\cdot 0 = \hat{p}$. Likewise, if you toss it into the employment bucket, your expected earnings are $\hat{p} \cdot 0 + (1-\hat{p}) \cdot 1 = 1-\hat{p}$. And tossing the ball into the unemployment bucket only becomes a good option when if $\hat{p} \geq \frac{1}{2}$.

But this means that there may be many situations where you pick up a ball, inspect it, learn information which increases your confidence that the ball belongs in the **un**employment bucket, and then toss the ball into the employment bucket anyways. In Bayesian terms, your prior odds ratio for unemployment is $\frac{1}{19}$, and you should only put a ball into the unemployment bucket if your posterior odds ratio is at least 1. As such, you require a likelihood ratio of at least 19 for you to deviate from the lazy strategy of always tossing the balls into the employment bucket. And a likelihood ratio of 19 is *a lot;* it's the kind of information you get from high-quality medical diagnostics. So unless the information written on the balls is really really predictive, you'll probably just be ignoring most of what you read.

When rewarding you for your accuracy, you just ended up tossing the balls into the same bin. So what's the point of even paying you to inspect the things?

3.2 Classifying when you are rewarded for *Balanced* accuracy.

Now suppose we change the incentive structure, so that you now earn r_u for correctly sorting a ball into the unemployment bucket and r_{nu} for correctly sorting a ball into the employment bucket.

If the balls have no useful information written on them, then the expected earnings from placing a ball into the unemployment bucket is $\bar{p} \cdot r_u$ and the expected earnings from placing a ball into the employment bucket is $(1-\bar{p}) \cdot r_{nu}$. To make you indifferent between blindly sorting into either of the two buckets, we set the rewards so that $\bar{p} \cdot r_u = (1-\bar{p}) \cdot r_{nu}$. One way we can accomplish this is by setting $r_u = \frac{1}{\bar{p}}$ and $\frac{1}{1-\bar{p}}$, giving you 20 dollars for each ball correctly sorted into the unemployment bucket and only a bit more than 1.05 dollars for each ball correctly sorted into the employment bucket.

To maximize your earnings under this new scheme, you must maximize your Balanced accuracy. In the formula for BAcc,

$$\text{BAcc} \equiv \frac{1}{2} \cdot \left[\frac{\sum_{i\in T} \ \hat{y}_{i,t+1} \cdot y_{i,t+1}}{\sum_{i\in T} y_{i,t+1}} \right] + \frac{1}{2} \cdot \left[\frac{\sum_{i\in T} \ (1-\hat{y}_{i,t+1}) \cdot (1-y_{i,t+1})}{\sum_{i\in T} (1-y_{i,t+1})} \right]$$

the terms $\frac{1}{\sum_{i\in T} y_{i,t+1}}$ and $\frac{1}{\sum_{i\in T}(1-y_{i,t+1})}$ are exactly the scaling factors we need to make a classifier "indifferent" between sorting individuals into the two different prediction categories, and the factor of $\frac{1}{2}$ is used to simply normalize the formula to a maximum of 1.

Back to the thought experiment. In this scheme, if you don't get any useful information from inspecting the ball, you can put it in either bin and get the same expected earnings. But if you read information written on the ball which increases your belief *at all* that the ball belongs in one of the two buckets, you'll increase your earnings by putting it into that bucket. The likelihood ratio no longer needs to be enormous to make you care about the information you have learned. And you can now improve the sorting of the balls (compared to random sorting) , even if your inspection abilities aren't medical-grade.

4 In Conclusion

The current GF measure encourages degenerate strategies because the formula overcompensates for the low prevalence of people who will be unemployed next period. The terms in the denominators of the brackets already properly adjust for this difference in prevalence, and so the coefficients outside of the brackets result in unemployed predictions being overemphasized.

Appendix: Notation Used in Formulas

- $y_{i,t+1}$ is a binary variable describing whether an individual in the sample is unemployed in the following year. $y_{i,t+1} = 1$ if individual i will be unemployed, and $y_{i,t+1} = 0$ otherwise.
- $\hat{y}_{i,t+1}$ is the classifier's prediction for whether individual i will be unemployed in the following year. It is also a binary variable, with $\hat{y}_{i,t+1} = 1$ corresponding to a prediction of unemployment.
- $\bar{y} \approx 0.0514$ is the portion of individuals in the training sample for whom $y_{i,t+1} = 1$
- T is the set of individuals in the test sample, and $|\mathrm{T}|$ is the number of individuals in the test sample.
- $\mathrm{TP} \equiv \sum_{i \in T} \ \hat{y}_{i,t+1} \cdot y_{i,t+1}$ is the number of *true positives,* the number of individuals in the test sample which the classifier correctly predicts as being unemployed in the following year.
- $\mathrm{TN} \equiv \sum_{i \in T} \ (1 - \hat{y}_{i,t+1}) \cdot (1 - y_{i,t+1})$ is the number of *true negatives,* the number of individuals in the test sample which the classifier correctly predicts as *not* being unemployed in the following year.
- $\mathrm{TPR} \equiv \frac{\mathrm{TP}}{\sum_{i \in T} y_{i,t+1}}$ is the *true positive rate,* which is the empirical counterpart to $\Pr\left(\hat{y}_{i,t+1} = 1 \mid y_{i,t+1} = 1\right)$.
- $\mathrm{TNR} \equiv \frac{\mathrm{TNR}}{\sum_{i \in T} (1 - y_{i,t+1})}$ is the *true negative rate,* which is the empirical counterpart to $\Pr\left(\hat{y}_{i,t+1} = 0 \mid y_{i,t+1} = 0\right)$.