



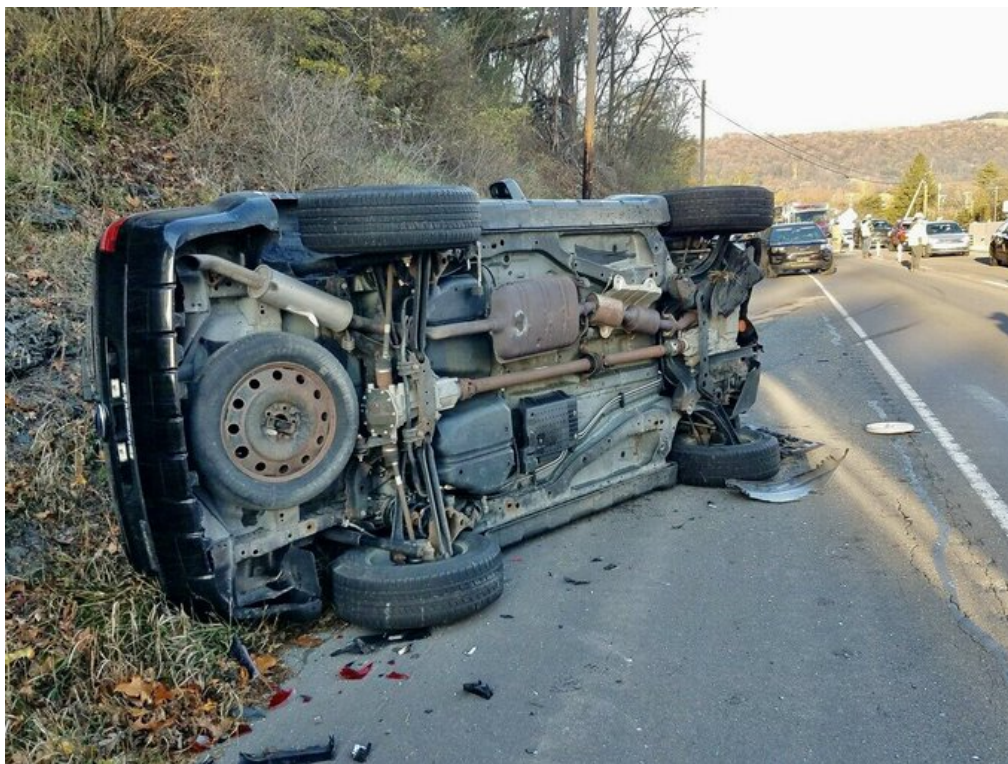
Santiago @svpino

6h

I built a model to predict whether you'll be involved in a crash next time you get in a car.

And it's 99% accurate!

Allow me to show you... 🖱️



💬 11 🔄 72 📄 24 ❤️ 432



Santiago @svpino

6h

Here is the model:



```
def am_I_crashing_today_in_a_car():  
    return False
```

💬 4 🔄 1 📄 0 ❤️ 60



Santiago @svpino

6h

The National Safety Council reports that the odds of being in a car crash in the United States are 1 in 102.

That's a probability of 0.98% of being involved in a crash.

Therefore, my silly model is accurate 99% of the time!

See? I wasn't joking before.





Santiago @svpino

6h

By now, it is probably clear that using "accuracy" as the way to measure the predictive capability of a model is not always a good idea.

The model could be very accurate... and still, give you no useful information at all.

Like right now.



12:16 PM · Feb 4, 2021

💬 1 ↻ 1 🗑️ 0 ❤️ 53



Santiago @svpino

6h

Determining whether you are crashing on a car is an "imbalanced classification problem."

There are two classes: you crash, or you don't. And one of these represents the overwhelming majority of data points.

Takeaway: Accuracy is not a great metric for this type of problem.



💬 1 ↻ 0 🗑️ 0 ❤️ 33



Santiago @svpino

6h

Crashing a car is a little bit too morbid, so here are a few more problems that could be framed as imbalanced classification tasks as well:

- Detecting fraudulent transactions
- Classifying spam messages
- Determining whether a patient has cancer



💬 1 ↻ 0 🗑️ 0 ❤️ 25



Santiago @svpino

6h

We already saw that we can develop a "highly accurate" model if we classify every credit card transaction as not fraudulent.

An accurate model, but not a useful one.

How do we properly measure the model's effectiveness if accuracy doesn't work for us?



💬 1 ↻ 0 🗑️ 0 ❤️ 23



Santiago @svpino

6h

We care about *positive* samples (those transactions that are indeed fraudulent,) and we want to maximize our model's ability to find them.

In statistics, this metric is called "recall."

[Recall – Ability of a classification model to identify all relevant samples]



💬 1 ↻ 0 🗑️ 0 ❤️ 29



Santiago @svpino

6h

A more formal way to define Recall is through the attached formula.

- True Positives (TP): Fraudulent transactions that our model detected.
- False Negatives (FN): Fraudulent transactions that our model missed.

$$\text{RECALL} = \frac{\text{TRUE POS.}}{\text{TRUE POS.} + \text{FALSE NEG.}}$$

💬 1 ↻ 0 🗑️ 0 ❤️ 20



Santiago @svpino

6h

Imagine that we try again to solve the problem with the attached (useless) function. We are classifying every instance as negative, so we are going to end up with 0 recall:

- $\text{recall} = \text{TP} / (\text{TP} + \text{FN}) = 0 / (0 + \text{FN}) = 0$



```
def is_transaction_fraudulent(t):  
    return False
```

💬 2 ↻ 0 🗑️ 0 ❤️ 16



Santiago @svpino

6h

That's something!

Now we know that our model is completely useless by using "recall" as our metric. Since it's 0, we can conclude that the model can't detect any fraudulent transactions.

Ok, we are done!

Or, are we?



💬 1 ↻ 0 🗑️ 0 ❤️ 15



Santiago @svpino

6h

How about if we change the model to the attached function?

Now we are returning that every transaction is fraudulent, so we are maximizing True Positives, and our False Negatives will be 0:

- $\text{recall} = \text{TP} / (\text{TP} + \text{FN}) = \text{TP} / \text{TP} = 1$

Well, that seems good, doesn't it? 😞



```
def is_transaction_fraudulent(t):
```

💬 1 ↺ 0 🗑️ 0 ❤️ 13



Santiago @svpino

6h

A recall of 1 is indeed excellent, but again, it just tells part of the story. Yes, our model now detects every fraudulent transaction, but it also misclassifies every normal transaction! Our model is not too *precise*.



💬 1 ↺ 0 🗑️ 0 ❤️ 14



Santiago @svpino

6h

As you probably guessed, "precision" is the other metric that goes hand in hand with "recall."

[Precision – Ability of a classification model to identify only relevant samples]



💬 1 ↺ 0 🗑️ 0 ❤️ 17



Santiago @svpino

6h

A more formal way to define Precision is through the attached formula.

- True Positives (TP): Fraudulent transactions that our model detected.
- False Positives (FP): Normal transactions that our model misclassified as fraudulent.



$$\text{PRECISION} = \frac{\text{TRUE POS.}}{\text{TRUE POS.} + \text{FALSE POS.}}$$

💬 1 ↺ 0 🗑️ 0 ❤️ 17



Santiago @svpino

6h

Let's compute the precision of our latest model (the one that classifies every transaction as fraudulent):

- TP = just a few transactions, so a small number
- FP = (1 - a small number) = large number
- precision = TP / (TP + FP) = small / large ≈ 0



💬 1 ↺ 0 🗑️ 0 ❤️ 14



Santiago @svpino

6h

The precision calculation wasn't that clean, but hopefully, it is clear that the result will be very close to 0.

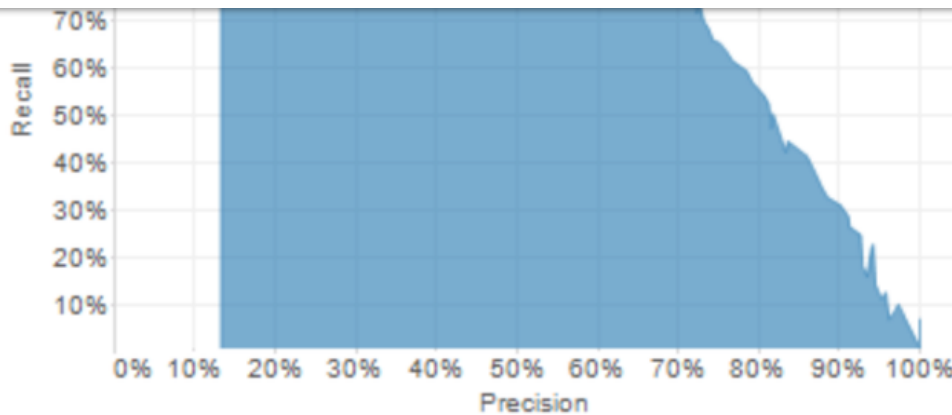
So we went from one extreme to the other!

Can you see the relationship?

As we increase the precision of our model, we decrease the recall and vice-versa.



100%



💬 1 ↻ 0 🗨️ 0 ❤️ 16



Santiago @svpino

6h

Alright, so now we know a few things about imbalanced classification problems:

- Accuracy is not that useful.
- We want a high recall.
- We want high precision.
- There's a tradeoff between precision and recall.

There's one more thing that I wanted to mention.



💬 1 ↻ 0 🗨️ 0 ❤️ 22



Santiago @svpino

6h

There may be cases where we want to find a good balance between precision and recall.

For this, we can use a metric called "F1 Score," defined with the attached formula.

[F1 Score — Harmonic mean of precision and recall]



$$F1\ SCORE = 2 * \frac{Precision * Recall}{Precision + Recall}$$

💬 1 ↻ 0 🗨️ 0 ❤️ 21



Santiago @svpino

6h

The F1 Score gives equal weight to both precision and recall and punishes extreme values.

This means that either one of the dummy functions we discussed before will show a very low F1 Score!

My models suck, and they won't fool the F1 Score.



💬 2 ↻ 0 🗨️ 0 ❤️ 20



Santiago @svpino

6h

So that's it for this story.

If you want to keep reading about metrics, here is an excellent, more comprehensive





Alejandro Piad Morffis @AlejandroPiad

Feb 2

This is a Twitter series on [#FoundationsOfML](#). Today, I want to talk about another fundamental question:

? What makes a metric useful for Machine Learning?

Let's take a look at some common evaluation metrics and their most important caveats...  

[Show this thread](#)

 6  1  0  25

