31 May 2021 Report

Brad Burkman

27 May 2021

Contents

1 Accuracy, Precision, Recall, Sensitivity, f1, False Alarm Rate 1

2 Train/Test Split for Imbalanced Data Set 2

3 SMOTE: Synthetic Minority Oversampling TEchique 2

4 Negative Samples 2

5 Google Maps Data? 3

6 Feature Selection, Redundant Variables 3

7 Citing Sources for Common Knowledge 4

8 California Statewide Integrated Traffic Records System (SWITRS) 4

9 References 5

1 Accuracy, Precision, Recall, Sensitivity, f1, False Alarm Rate

In a previous report I said I was getting 99% accuracy, and you agreed that it seemed suspicious.; I've now figured out why. I'm getting 99% accuracy because the dataset is so imbalanced. We have 681 fatal crashes out of 160,186, or 0.43%. If the ML model predicts that all of the crashes are non-fatal, it will get 99.57% accuracy. What we want to measure is *recall*, also called *sensitivity* and *detection rate*, which is the proportion of fatal crashes that we have correctly identified.

$$\text{Recall} = \frac{TP}{TP + FN}$$

I'm amazed that most of the articles I'm reading talk primarily about accuracy. They may mention recall/sensitivity in Section 4 or 5, but the abstract, introduction, and conclusion only mention accuracy.

Confusion Matrix in SciKit-Learn.

		Prediction	
		N	P
Actual	N	TN	FP
	P	FN	TP

2 Train/Test Split for Imbalanced Data Set

For balanced data sets, the question in splitting into Train and Test data is whether you want a 50/50, 75/25, 80/20, or 90/10 split.

For this Louisiana crash data set, we have 160,168 records, only 681 of which are fatal crashes. We want to find needles in the haystack, but we don't have many needles. If we want an 80/20 split, should we make sure that the test set has 20% of the needles?

We could do that by first splitting the dataset into Positive and Negative sets, taking an 80/20 split of each set, then combining the splits into the Train and Test set. I've implemented that.

3 SMOTE: Synthetic Minority Oversampling TEchique

Used to balance an imbalanced dataset by creating new data points for the minority class.

"New synthetic data points are created by forming a convex combination of neighboring members." [1]

Might be something I should try.

4 Negative Samples

In Roland [2], in the lit review, they talk about previous authors creating *negative samples.* For each crash record, change one feature from {hour, day, unique road identification}, and see if there's an actual crash record with that set of features. If not, make it another record representing `not_a_crash`. The ideal proportions of positive (crash) and negative (generated non-crash) is a matter of debate.

Is the number of crashes on a stretch of road at a particular time of day proportional to the traffic volume? If we know the traffic volume, should we put in negative samples in such a way that the dataset reflects the traffic volume at each time of day?

For most roads, we don't have much data on traffic volume per hour. What kinds of roads have really good records? Toll roads. Unfortunately, Louisiana doesn't have any, but Texas does.

5 Google Maps Data?

The dataset we have has lots of information that's only available after the crash, not available in real time. What if we combined the following data streams and focused on a particular stretch of road that have frequent traffic slowdowns, like I-10 around the Mississippi River Bridge in Baton Rouge or I-10 through Kenner (near the New Orleans airport)?

- Google Maps (or similar) crash reports, which would give time and location.
- Other Google Maps reports, such as speed traps, slowdowns, construction, lane closures, stalled vehicles, and objects on road.
- Google Maps real-time data on traffic speed.
- Louisiana DOTD traffic cameras, which would give traffic volume and speed. A well positioned camera could give information about a reckless driver, either because of
  - Higher speed than the traffic
  - Tailgating
  - Sudden or multiple lane changes
- Current weather (Python API *Dark Sky*)
- Time of day and day of week.

We expect slowdowns 7-9am and 4-6pm that correlate to high traffic volume, but the interesting parts are when traffic is not heavy then suddenly slows down.

6 Feature Selection, Redundant Variables

I'll start this week to use feature selection algorithms on our Louisiana crash data to narrow the number of features.

In Roland [2], they ran feature selection, but it selected some weather variables that correlate to the time of day, like humidity, uvIndex, temperature, dewPoint, pressure, and visibility, and excluded Rain/cloudy/foggy/snow/clear, Rain in previous hour, and Precipitation intensity. I need to check the correlations between features, not just the correlation of each feature to the dependent variable.

The same article makes a weird argument for including variables it acknowledges are redundant, like Lat/Lon and Grid_Num, because they use different scales. (?)

There are methods to help with feature selection, but do those evaluate whether two features give you the same information? Methods/software mentioned in the literature include ANOVA and Gini.

Could we normalize some of the weather data to be above/below average at that time of day and that time of year? Or use a daily value rather than an hourly value? I find it hard to believe that uvIndex is more predictive than rain. Roland’s article suggests future work making a single variable to represent the weather for the day, as was done in Hébert [3].

The Dark Sky API that Roland used has been bought by Apple and is no longer accepting new signups, but there are others that do the same thing. The remaining documentation gives a forecast by year, month, day, and hour, returning the information in this format:

```
{
’ozone’: 290.06,
’temperature’: 58.93,
’pressure’: 1017.8,
’windBearing’: 274,
’dewPoint’: 52.58,
’cloudCover’: 0.29,
’apparentTemperature’: 58.93,
’windSpeed’: 7.96,
’summary’: ’Partly Cloudy’,
’icon’: ’partly-cloudy-night’,
’humidity’: 0.79,
’precipProbability’: 0,
’precipIntensity’: 0,
’visibility’: 8.67,
’time’: 1476410400
}
```

The documentation says there are other attributes that may or may not exist in a particular record, including uvIndex, windGust, precipAccumulation, and precipType.

7 Citing Sources for Common Knowledge

Many of the papers I’ve read cite sources for common knowledge in the field, like the definitions of recall, precision, and accuracy, and statements like “driving is dangerous.” It seems silly to me.

8 California Statewide Integrated Traffic Records System (SWITRS)

`https://iswitrs.chp.ca.gov/Reports/jsp/index.jsp`
Apparently one can create an account and get data for research purposes.

9 References

[Par+19] Amir Bahador Parsa, Homa Taghipour, Sybil Derrible, et al. “Real-time accident detection: Coping with imbalanced data”. In: *Accident Analysis & Prevention* 129 (2019), pp. 202–210. ISSN: 0001-4575. DOI: https://doi.org/10.1016/j.aap.2019.05.014. URL: https://www.sciencedirect.com/science/article/pii/S0001457519301642.

[Rol+21] Jeremiah Roland, Peter D. Way, Connor Firat, et al. “Modeling and predicting vehicle accident occurrence in Chattanooga, Tennessee”. In: *Accident Analysis & Prevention* 149 (2021), p. 105860. ISSN: 0001-4575. DOI: https://doi.org/10.1016/j.aap.2020.105860. URL: https://www.sciencedirect.com/science/article/pii/S0001457520316808.

[Heb+19] Antoine Hebert, Timothee Guedon, Tristan Glatard, et al. “High-Resolution Road Vehicle Collision Prediction for the City of Montreal”. In: *2019 IEEE International Conference on Big Data (Big Data)* (Dec. 2019). DOI: 10.1109/bigdata47090.2019.9006009. URL: http://dx.doi.org/10.1109/BigData47090.2019.9006009.