Prospectus Report: Building a Model to Recommend Dispatching an Ambulance based on Automated Crash Reports from Cell Phones

Brad Burkman

Updated 2 November 2022

One-Page Summary

Problem: Given an automated crash report from a cell phone, use two historical datasets to build and analyze models to recommend whether to immediately dispatch an ambulance. The solution requires data cleaning, imputing unknown values, and handling imbalanced data.

1. I am qualified and on track to graduate in December 2023. See Qualifications.
2. My dissertation will demonstrate competence in the major techniques of research.
    a. Finding an interesting question whose answer requires current and novel methods
    b. Literature review
    c. Finding appropriate datasets
    d. Data cleaning and imputation of missing values
    e. Handling imbalanced data
    f. Building, testing, and comparing models
    g. Analysis of results in terms of the application
    h. Analysis of results in terms of the current and novel methods
3. My dissertation will make novel contributions to the field.
    a. Novel application
    b. Previously unused method for imputing unknown values in a major dataset
    c. New metrics: Balanced precision and balanced F1
    d. New interpretation of class weights as a political and ethical cost-benefit tradeoff
4. My dissertation will demonstrate that I have wrestled with the data in these aspects that are as much art as science.
    a. Imputing missing values
    b. Binning (discretizing, batching) many categories into fewer
    c. Order of operations for imputing and binning
    d. Handling imbalanced data
5. I have reviewed the literature in these areas.
    a. ML metrics for imbalanced data
    b. Dataset balancing techniques
    c. Others' use of the CRSS dataset and how they handled missing data
    d. Use of the metrics and imbalanced data techniques in the crash analysis literature
6. I am preparing a paper for submission to a respected journal. See accompanying draft. I want the paper to be a model of open science, with a technical paper and clean code available on GitHub.
7. I have a detailed and realistic plan for completing the dissertation.
8. This document and the accompanying paper draft illustrate that I can make useful large documents to different specifications.
9. I have more questions than answers.

Chapter 0

Qualifications

0.1 Preparation and Degree Plan

0.1.1 Previous Education

1989 - 1993 Wheaton College (IL)
B.A. in English and Economics (double major)

1998 - 2000 SUNY Buffalo
M.A. Mathematics
Returned 2001-03 for additional coursework (total 60 hours)
Passed first PhD Qualifying Exam

0.1.2 Courses Taken

Transfer - UBuffalo	3	MATH 595	PDE's
Transfer - UBuffalo	3	MATH 555	Numerical Analysis I
Transfer - UBuffalo	3	MATH 556	Numerical Analysis II
Transfer - LSU Shreveport	3	CSCE 502	Bioinformatics
Fall 2018	3	CSCE 515	Graphics
Fall 2018	3	CSCE 553	Software Methodology
Fall 2018	3	CSCE 561	Information Storage and Retrieval
Fall 2018	1	CSCE 595	Seminar
Fall 2018	3	CSCE 669	Raghavan Adv. Topics
Spring 2019	3	CSCE 500	Algorithms
Spring 2019	3	CSCE 509	Pattern Recognition
Spring 2019	3	CSCE 530	Architecture
Spring 2019	1	CSCE 595	Seminar
Fall 2019	3	CSCE 572	Combinatorial and Geometric Algorithms
Fall 2019	1	CSCE 595	Seminar
Spring 2020	3	CSCE 619	Jin Adv. Topics

Spring 2020	1	CSCE 595	Seminar
Fall 2020	3	CSCE 619	Jin Adv. Topics
Fall 2020	1	CSCE 595	Seminar
Spring 2021	3	CSCE 619	Jin Adv. Topics
Summer 2021	3	CSCE 619	Jin Adv. Topics
Fall 2021	3	CSCE 699	Jin Dissertation
Spring 2022	3	CSCE 699	Jin Dissertation
Summer 2022	3	CSCE 699	Jin Dissertation
Total (excluding 595)	57		
Total 595	5		

0.1.3 Examinations

GRE (19 May 2016)
- 170 Quantitative
- 170 Verbal

PhD Comprehensive Exams
- Software Engineering (January 2019)
- Algorithms (August 2019)

0.1.4 PhD Degree Requirements

- √ CSCE 500
- √ Breadth Requirement
  - One 500-level course in hardware
    - CSCE 530
  - Two 500-level courses in software
    - CSCE 553 Software Methodology
    - CSCE 561 Information Storage and Retrieval
  - One 500-level course in theory
    - CSCE 500 Algorithms
  - One other 500-level course in areas not listed above
    - CSCE 515 Graphics
  - Any accepted 500-level course
    - CSCE 509 Pattern Recognition
- 7, 3.85 √ Six 500-level courses in CACS with a GPA of at least 3.5
- 12 √ At least 9 hours of CSCE 6x9 research courses
- √ PhD Comprehensive Exam
  - Software Engineering (January 2019)
  - Algorithms (August 2019)
- × PhD Prospectus Exam

	×	PhD Dissertation Defense
9	×	Exactly 24 hours of CSCE 699 (dissertation credit)
48	√	48 other hours
5	√	5 semesters of CSCE 595

0.1.5 Fall 2022 Plan

Fall 2022	3	CSCE 699	Dissertation
		Prospectus Exam	(tentatively Friday 4 November 2022)

0.1.6 Remaining Requirements

12 hours of 699

PhD Dissertation Defense

0.1.7 Plan for Completing Degree

Six years from Fall 1998

Spring 2023	3 hours 699
Summer 2023	6 hours 699
Fall 2023	3 hours 699
	PhD Dissertation Defense (December 2023)

0.1.8 Committee Members

Dr. Henry Chu	CACS	Chair
Dr. Xiaoduan Sun	Civil Engineering	
Dr. Aminul Islam	CACS	
Dr. Mehmet Tozal	CACS	

0.2 Previous Work

Two documents accompany this prospectus report to show the variety of work I have done.

1. A partial draft of the paper I plan to submit to *Transportation Research Part C: Emerging Technologies* in January 2023, using the journal's LaTeXtemplate.
2. My study guide for the 2019 Algorithms qualifying exam.
   `https://github.com/bburkman/Algorithms_Comp_Prep/blob/2fabe0e05bb13118a58a83e55016f4158de19c9c/CSCE_500_Comps_Prep/Algorithms_Comp.pdf`

Chapter 1

Introduction

1.1 Problem

1.1.1 Application

New (starting in 2022) Google Pixel phones have a feature that will automatically alert the police when involved in an automobile crash. Apple says the feature is coming to iPhones and Apple Watches soon; those products already have a feature that detects a person falling, calls the person, and if no response, calls a neighbor, a friend, or the police. One of my friends with multiple sclerosis uses this app.

Such systems (like GM OnStar) , built into vehicles, have existed for years, but soon they will become ubiquitous. When the police receive a notification, based on the information they have, should they automatically deploy an ambulance? In an accident with severe (but not instantly fatal) injuries, a few minutes' delay may have serious consequences, but sending an ambulance is expensive, and their supply is limited. Can we develop a model that will, from the limited information the police can hope to have, from the datasets we have chosen, build a model to make a good prediction of whether an ambulance is needed?

I am using "police" as a shorthand for "the decision makers at the emergency call center."

This new cell phone feature will not be perfect; it will give many false positives and may not detect crashes with small objects, like pedestrians, that do not cause severe deceleration but are most likely to have severe injury. The automated reports may, however, give us additional information like the number of people (number of phones) involved, and speed at time of impact. This new phone feature will keep the crash analysis community busy for many years.

The "make a good predicition" part is complicated. We are not going to get 100% accuracy. What would we mean by "good," and what would we use as a basis of comparison? The current system relies mostly on phone calls from eyewitnesses who can give more information than the police will have in an automated notification. These are thorny questions that we must address.

1.1.2 Datasets

I am looking at two datasets, the US Department of Transportation (DOT) National Highway Transportation Safety Board (NHTSB) Crash Report Sampling System (CRSS) data 2016-2020 data ($\approx 250,000$ records), and a census of Louisiana crash records 2014-18 ($\approx 800,000$ records).

1.1.3 Imbalanced Data

In the 2014-2018 Louisiana data, we have over eight hundred thousand crash records. If we are just looking for fatal crashes, about 3500 were fatal, 0.42%. If we built a model to predict whether a crash is fatal, and the model predicted that all crashes were nonfatal, that model would have correctly classified 99.58% of crashes, or have 99.58% *accuracy*. In most contexts, that level of accuracy would be amazing, but in this context, such a model would be useless.

In the CRSS dataset, which over represents severe crashes, 81.15% of people involved in a crash were not transported to the hospital, and 16.75% went to the hospital (the remaining 2.10% unknown). This nearly 5:1 imbalance is not as severe as the example with fatalities above, but still will be a challenge for our usual model building algorithms to give us the insights we seek.

The problem of imbalanced data appears in many applications, including spam detection and credit card fraud detection, and over the past decades the community has built many tools for addressing the problem. Applying those tools is as much art as science, and the best combination of methods depends on the dataset and desired outcome. The desired outcome is a moral, ethical, and political question as well as a technical one.

1.1.4 Tradeoffs

Balancing false positives and false negatives in this application is additionally problematic because they have different costs. The cost of a false positive (sending an ambulance when one is not needed) is measured in dollars, but the cost of a false negative (not sending an ambulance when one is needed) is measured in lives. It is likely that this study will only illustrate the choices to be made rather than find a "best" solution that will significantly increase the number of true positives without increasing the number of false positives.

1.2 Novel Contributions of this Work (Knowledge Gap)

Novel Aspects of this Work

- New Real-World Problem: Newly emerging problem of how to use the greatly increasing volume of automated crash notification data.
- New Dataset: The Louisiana dataset has not appeared significantly in the literature.
- New Imputation of Unknown Values in Well Known Dataset (CRSS)
- New Metrics: Balanced Precision and Balanced F1
- Interpretation of Class Weights as a Political/Ethical Cost-Benefit Tradeoff

- New Combinations of Methods: The Louisiana data is very incomplete, dirty, and imbalanced, and the CRSS data is imbalanced. Off-the-shelf methods will not give the level of confidence needed for life-and-death decisions.

Chapter 10

Research Plan

10.1 Progress To Date

- Reviewed literature (ongoing process)
- Developed problem
- Chose datasets (Crash Report Sampling System (CRSS) [**CRSS**] and Louisiana)
- Learned how to build custom loss functions in Keras. (It's not really an option in scikit-learn.)
- Understood a wide variety of methods for handling imbalanced data. Many of them are available in Keras, and some I had to implement myself.
- Learned to use Imputation and Variance Estimation Software (IVEware) [**IVEware**] and used it to impute unknown values in CRSS

10.2 Goals

10.2.1 CRSS Data Set

Crash Report Sampling System (CRSS) [**CRSS**]

- Answer question about whether binning or imputing should come first
- Some of the binning is not consistent; make a clear rationale for binning and apply it
- Finish preparing the data
- Apply imbalanced data techniques, testing individually and in combination
- Analyze results

10.2.2 Paper for *Transportation Research Part C: Emerging Technologies*

- Reread papers from this journal that are models of good writing
- Read and reread the submission policies
- Revise paper
- Make a list of opportunities for future research
- Write and post technical paper

- Submit
- Get feedback
- Respond to feedback

10.2.3 Louisiana Data Set

- Select features to match/complement what I did with CRSS data
- Clean
- Discretize data. This will be different from what I did with CRSS, because some of the data is continuous
- Impute missing values
- Apply imbalanced data in a way that matches/complements what I did with the CRSS data
- Analyze results

10.2.4 Write Dissertation

- Review the literature again
- Find and review good examples of dissertations
- Write
- Revise
- Repeat

10.3 Timeline

October 2022	Answer question for CRSS about order of operations of binning and imputing unknown values
	Finish preparing CRSS data
November 2022	Test imbalanced data techniques (and combinations thereof) on CRSS data
December 2022	Analyze results
January 2023	Submit paper to *Transportation Research Part C: Emerging Technologies*
February 2023	Clean Louisiana database
	Respond to reviews from TR_C
March 2023	Wrestle with the data: Figure out how to use Louisiana and CRSS data together
April 2023	Test imbalanced data techniques (and combinations thereof) on the Louisiana data
May 2023	Write first draft of dissertation
June 2023	Get feedback, Read papers, Rework, Write, and Revise
July 2023	Get feedback, Read papers, Rework, Write, and Revise
August 2023	Get feedback, Read papers, Rework, Write, and Revise
September 2023	Get feedback, Revise dissertation
October 2023	Submit Dissertation
December 2023	Dissertation Defense
15 December 2023	Graduation