Prospectus Report: Imbalanced Data

Brad Burkman

Updated 26 September 2022

Contents

Index

Chapter 0

To Do

0.1 Topics Remaining

0.1.1 Big Items

- Crash Analysis Lit Review
- Data Overview and Analysis
- Research Plan

0.1.2 Details

- Worked through Brad's Reports
- Find where it mentions multiple Tomek runs.
- Work through this tutorial, which will explain Condensed Nearest Neighbor.
  https://machinelearningmastery.com/undersampling-algorithms-for-imbalanced-classification/
- Figure out what Rough Sets Theory and Fuzzy Sets are.
- Learn to use ROC curves.
- Review different types of models and how to implement them in Keras.
- Bagging and Boosting
- Topological Data Analysis
  Giotto-tda is a Python package dedicated to integrating TDA in the machine learning workflow by means of a scikit-learn API

0.2 Plan

- Identify review papers by established experts that give overviews of the field. Use these as a "textbook" for imbalanced data.
- Write a review of the field (similar to my study guide for the Algorithms Comp)
  - Topics
    - Benchmark datasets for imbalanced classification.

    - Sampling methods
    - Metrics
    - Loss functions
    - Bagging and Boosting
    - Math and CS tools, like Principal Component Analysis
  - For each topic
    - Original and significant papers
    - Theory and rationale
    - Examples of usage in the field
    - Example where I implement it
    - If appropriate, implement it on the crash data

0.3 Paper Standards

When I took a 619 with Dr. Raghavan, he rejected my first draft of my paper because it could be summarized as, "I used existing methods to do the same thing other people have done, but on my own data." He expected me to have done more thinking and synthesis.

I had written the paper that way because that's what most of the papers I'd read did. I learned later that I had been reading in a low-ranked journal.

I feel like most of the *Accident Analysis and Prevention* papers are like that, "I did a thing!" If this field had benchmark datasets, it would be easier to see what is actually new in each article. The *AAP* journal is not well ranked. The *Transportation Research: Part C, Emerging Technologies* is better in both content and rankings. I'll use the *TRC* articles as models.

0.4 Paper Outline Idea from 7/19/21 Report

1. Louisiana Crash Dataset and its Challenges
    a. General Description
    b. Some data is missing
    c. Some data is unreliable or obviously wrong
    d. Data is imbalanced
2. Incorporating Other Data
    a. Weather
    b. Urban/Rural
3. Data Cleanup: Methods
    a. Comparison of methods for handling missing data
    b. Methods for dealing with outliers
4. Features: Methods

    a. Engineering New Features
    b. Selecting Features [Incorporate Jing Chen's methods]

5. Imbalanced Data: Comparison of Methods
    a. Oversampling
    b. Understampling
    c. `class_weight = 'balanced'`
    d. Finding balance between methods [pun intended]

6. New Metrics: *Balanced Precision* and *Balanced f1*
    a. Definitions
    b. Justification
    c. Examples

7. Models
8. Balancing Everything
9. Conclusion

Chapter 1

Introduction

1.1 Problem

1.1.1 Application

New (starting in 2022) iPhones and Apple Watches have a feature that will automatically alert the police when involved in an automobile crash. Such systems (like GM OnStar) have existed for years, but now 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 dataset we have, 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 iPhone 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. It may, however, give us additional information like the number of people (number of phones) involved, and speed at time of impact. This new iPhone feature will keep the crash analysis community busy for many years.

The "make a good predicition" part is complicated. We're 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

So far I am looking at two datasets, one of Louisiana crash records 20142-18 ($\approx 800,000$ records) and the US NHTSA (National Highway Transportation Safety Administration) Crash Investigation Sampling System data 2016-2020 data ($\approx 250,000$ 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.

The dataset classifies the crashes as "Fatal," "Severe," "Moderate," "Complaint," and "No Injury," also called PDO, "Property Damage Only." Note that "Fatal" does not mean that everyone involved died, so there may be people moderately or seriously injured who need urgent medical attention. If we are looking to model whether a crash is Fatal or Severe, those together are 1.1%, and Fatal/Severe/Moderate is 6.8%.

1.1.4 Tradeoffs

We would like to predict with good certainty whether a crash has severe injuries, so we can dispatch an ambulance as soon as possible, but we may need to include moderate injuries to have a significant amount of data. We would also like to not send ambulances where one is not needed, because they're expensive and in finite supply. We will need to tolerate some threshold of false positives, and that threshold will ultimately be set by policy makers, not by data scientists. Different techniques will give us a different balance of true positives, false positives, false negatives, and true negatives, and it is likely that this study will only illustrate the choices to be made rather than find a silver bullet that will significantly increase the number of true positives without increasing the number of false positives.

1.2 Rationale

Novel Aspects of this Work

- New Real-World Problem: Newly emerging problem of what to do with greatly increased volume of automated crash notification data.
- New Dataset: This particular dataset (Louisiana crash reports) has not been used in this way before.
- New Combinations of Methods: The data is very incomplete, dirty, and imbalanced. Off-the-shelf methods will not give the level of confidence needed for life-and-death decisions.

Chapter 2

Lit Review: Crash Analysis

2.1 Journals

Journal	CiteScore	Impact Factor
Accident Analysis and Prevention	7.8	4.993

Accident Analysis & Prevention provides wide coverage of the general areas relating to accidental injury and damage, including the pre-injury and immediate post-injury phases. Published papers deal with medical, legal, economic, educational, behavioral, theoretical or empirical aspects of transportation accidents, as well as with accidents at other sites. Selected topics within the scope of the Journal may include: studies of human, environmental and vehicular factors influencing the occurrence, type and severity of accidents and injury; the design, implementation and evaluation of countermeasures; biomechanics of impact and human tolerance limits to injury; modelling and statistical analysis of accident data; policy, planning and decision-making in safety.

American Journal of Emergency Medicine	3.2	2.469

A distinctive blend of practicality and scholarliness makes the American Journal of Emergency Medicine a key source for information on emergency medical care. Covering all activities concerned with emergency medicine, it is the journal to turn to for information to help increase the ability to understand, recognize and treat emergency conditions. Issues contain clinical articles, case reports, review articles, editorials, international notes, book reviews and more. The American Journal of Emergency Medicine is recommended for initial purchase in the Brandon-Hill study, Selected List of Books and Journals for the Small Medical Library (2001 Edition).

Decision Support Systems	10.5	5.795

The common thread of articles published in Decision Support Systems is their relevance to theoretical and technical issues in the support of enhanced decision making. The areas

addressed may include foundations, functionality, interfaces, implementation, impacts, and evaluation of decision support systems (DSSs). Manuscripts may draw from diverse methods and methodologies, including those from decision theory, economics, econometrics, statistics, computer supported cooperative work, data base management, linguistics, management science, mathematical modeling, operations management, cognitive science, psychology, user interface management, and others. However, a manuscript focused on direct contributions to any of these related areas should be submitted to an outlet appropriate to the specific area.

Examples of research topics that would be appropriate for Decision Support Systems include the following:

1. DSS Foundations e.g. principles, concepts, and theories of enhanced decision making; formal languages and research methods enabling improvements in decision making. It is important that theory validation be carefully addressed.

2. DSS Functionality e.g. methods, tools, and techniques for developing thefunctional aspects of enhanced decision making; solver, model, and/or data management in DSSs; rule formulation and management in DSSs; DSS development and use in computer supported cooperative work, negotiation, research and product.

3. DSS Interfaces e.g. methods, tools, and techniques for designing and developing DSS interfaces; development, management, and presentation of knowledge in a DSS; coordination of a DSS's interface with its functionality.

4. DSS Implementation - experiences in DSS development and utilization; DSS management and updating; DSS instruction/training. A critical consideration must be how specific experiences provide more general implications.

5. DSS Evaluation and Impact e.g. evaluation metrics and processes; DSS impact on decision makers, organizational processes and performance.

Journal of Safety Research 5.0 3.487

The Journal of Safety Research is a multidisciplinary publication that provides for the exchange of scientific evidence in all areas of safety and health, including traffic, workplace, home, and community. While this research forum invites submissions using rigorous methodologies in all related areas, it focuses on basic and applied research in unintentional injury and illness prevention. Affiliated with the National Safety Council, it seeks to engage the global scientific community including academic researchers, engineers, government agencies, policy makers, corporate decision makers, safety professionals and practitioners, psychologists, social scientists, and public health professionals.

Transportation Research Part C: Emerging Technologies 14.0 8.089

The focus of Transportation Research: Part C (TR_C) is high-quality, scholarly research that addresses development, applications, and implications, in the field of transportation systems and emerging technologies . The interest is not in the individual technologies per se, but in their ultimate implications for the planning, design, operation, control, maintenance and rehabilitation of transportation systems, services and components. In other words, the intellectual core of the journal is on the transportation side, not on the technology side. The integration of quantitative methods from fields such as operations research, control systems, complex networks, computer science, artificial intelligence are encouraged.

Of particular interest are the impacts of emerging technologies on transportation system performance, in terms of monitoring, efficiency, safety, reliability, resource consumption and the environment. Submissions in the following areas of transportation are welcome: multimodal and intermodal transportation; on-demand transport; intelligent transportation systems; traffic and demand management; real-time operations; connected and autonomous vehicles; logistics; railways; resource and infrastructure management; aviation; pedestrians and soft modes.

Special emphasis is given in open science initiatives and promoting the opening of large-scale datasets for papers published in TR_C that can support transferability and benchmarking of different approaches. The realization of data opportunities that arise from emerging technologies and new sensors in transportation can revolutionize how this data reshape our understanding of congestion mechanisms and can contribute in efficient and sustainable mobility management.

2.2 Articles using Similar Datasets

- Rahim 2021 [**RAHIM2021106090**] LSU faculty, similar dataset to what we have.
- Jiang 2020 [**JIANG2020105520**] used similar data and addressed the challenges we'll have with it.

2.3 Articles on Imbalanced Crash Data

- Schlogl 2020 [**SCHLOGL2020105398**] uses imbalanced data.

2.4 Ambulances

2.4.1 Ambulance Response Time

From 11/29/21 Report.

- Found standard for emergency medical service (EMS) response time, from The National Fire Protection Association. "1710 NFPA Standard for the Organization and Deployment of

Fire Suppression Operations, Emergency Medical Operations, and Special Operations to the Public by Career Fire Departments, 2020" §4.1.2.1

  - 60-second turnout time
  - 240 seconds or less travel time for the arrival of a unit with first responder with automatic external defibrillator (AED) or higher-level capability at an emergency medical incident
  - 480 seconds or less travel time for the arrival of an advanced life support (ALS) unit at an emergency medical incident, where this service is provided by the fire department provided a first responder with an AED or basic life support (BLS) unit arrived in 240 seconds or less travel time.
  - Lots of papers, like Liu (2016)[**Liu˙2016**] cite Rafael Sa'adah (2004), which I think is a response to the NFPA standards, but I can't find it online or in the library database.

2.5 iPhone to Automatically Detect Crash and Call Emergency Services

From 11/29/21 Report.

- iPhones and Apple Watches will soon automatically call police when the accelerometer detects a car crash.
- Several articles dated 11/1/21, including in the Wall Street Journal.
- Available in 2022
- What data would that provide, and what data would the police already have to complement it? These are just my guesses.
- Data from Apple
  - Registered owner of the phone (or phones) in the car
  - Typical users of that phone (Apple knows!)
  - GPS location
  - Perhaps a rough idea of how fast the car was going and how suddenly it stopped
  - If more than one phone sends signal, do these people know each other, or are they likely in different vehicles?
  - Accelerometer signature of a pedestrian or bicyclist getting hit?
- Complementary data from police database
  - Type of roadway and speed limit
  - Was it at an intersection?
  - Time of day, day of week
  - Type of vehicle registered to that person
  - Driving record of user of phone (History of DUI?)
  - Weather

2.6 Weather

From 11/29/21 Report.

- Wang et al [**WANG2020102619**] studied the data of a ride-hailing company, DiDi Chuxing, and looked for how resilient the system was during "extreme weather events."
  - They defined such weather to be "hurricane, flooding, and rainstorm." (page 2) I suspect that "rainstorm," which is really vague in English, is a poor translation of a more specific Chinese word.
  - Because these extreme events are rare, they have a sample imbalance problem (page 13). They solve the problem in an interesting way, by ignoring it and watering down their data set. "The characteristics of urban transportation resilience under catastrophic events have generally similar patterns to those under general precipitation events. Thus we incorporated the rainstorm and usual prediction events data into [sic] data set to strengthen the model training." So, as I understand it, they had an imbalanced data problem modeling extreme weather, so they just modeled ordinary weather.
  - I like how the authors started their methodology section with a page of definitions.

2.7 Lagniappe

- Osman 2019 (LSU) [**OSMAN2019274**] looked much more deeply at the data than other studies, looking for correlations between sets of variables.
- Ziakopoulos 2020 [**ZIAKOPOULOS2020105323**] is a good overview of the field and its jargon.
- Guimmarra 2020 [**GIUMMARRA2020105333**] is interesting for its text mining of crash reports.
- Park 2019 [**Park˙2019**] has a full-page table categorizing studies of ambulance location, relocation, and dispatching using different optimization methods.

2.8 Significant Authors

From 11/15/21 Notes:

Reviewed all of the 66 articles from 2021 with the word "crash" in *Transportation Research Part C: Emerging Technologies*.

- Most of the articles are about autonomous vehicles.
- Mohammed Abdel-Aty at the U of Central Florida is a major author in this journal, but not in this year. In previous years, if there was an article from UCF, his name was on it. His website does not say that he has retired.
- When I write, I want to include more examples than many authors give.

2.9 TR_C Articles on Machine Learning

2.9.1 Application of articles whose keywords contain *machine learning*, *deep learning*, or *reinforcement learing*

- Autonomous Vehicles
  - Control of Autonomous Vehicles [**AN2020102777**], [**BAUTISTAMONTESANO2022103662**], [**CAO2022103656**], [**DONG2021103192**], [**DU2022103489**], [**GUO2021102980**], [**KALATIAN2021102962**], [**LAZAR2021103258**], [**LI2022103452**], [**SHI2021103421**], [**WANG2022103478**], [**WEGENER2021102967**], [**WU2020102649**], [**YE2019155**], [**ZHANG2021103140**], [**ZHU2020102662**],
  - Preferences for Autonomous Vehicles [**ZHANG2020102774**]
- Lagniappe
  - Anomalous Event Prediction [**YANG2021102862**],
  - Origin/Destination [**MA2020102747**], [**SUN2021103114**],
  - Variable Speed Limits [**WU2020102649**]
  - Dynamic Pricing [**GENSER2022103485**], [**HAN2022103584**], [**PANDEY2020102715**]
  - Parking [**MANTOUKA2021103173**], [**YANG2019248**], [**ZHANG2022103624**]
  - Traffic Signal Optimization [**LEE2019117**], [**LI2021103059**], [**WANG2021103046**], [**WANG2022103670**], [**WU2019246**], [**YOON2021103321**],
  - Perimeter Metering (?) [**ZHOU2021102949**]
  - Energy Consumption [**QI201967**], [**YAO2019276**]
  - Vehicle Idenfification [**DABIRI2020102644**], [**LI2021102946**],
  - Trip Purpose [**FAROQI2021103131**]
- Traffic
  - Traffic Prediction [**BACHIR2019254**], [**BOGAERTS202062**], [**CUI2020102620**], [**DAI2019142**], [**DO201912**], [**DRCHAL2019370**], [**KUMAR2021103432**], [**LI2021102977**], [**LI2021103185**], [**LI2021103389**], [**MA2020352**], [**ROY2021103339**], [**WANG2020102763**], [**WONG2020247**], [**YANG2021103228**], [**ZHANG2021103372**], [**ZHANG2022103659**],
  - Traffic Speed Prediction [**NAIR2020269**], [**REMPE2022103448**], [**WANG2019372**], [**ZHANG2019297**]
  - Traffic in Extreme Weather [**WANG2020102619**]
  - Traffic Signals [**GUO2021103416**], [**MAHMOUD2021102930**], [**ZHAO2022103522**]
  - Dynamic Traffic Control [**SHOU2022103560**]
- Individual Driver
  - Vehicle Behavior Modeling [**CHEN2020102646**], [**MA2020102785**], [**MO2021103240**], [**RAHMAN2021103310**], [**YAO2021103415**], [**ZHANG2019287**]
  - Classifying Driving Styles [**MOHAMMADNAZAR2021102917**]
  - Driver's Visual Environment [**CAI2022103541**], [**LI2019132**], [**MA2019317**]

  - Driver Behavior [**MOHAMMADNAZAR2021102917**], [**XING2021103288**],
  - Driver Distraction [**CAI2022103541**] This article is interesting, perhaps relevant to me, for correlating crashes with something else.

- Delivery
  - Delivery Times [**HUGHES2019289**],
  - Vehicle Routing Problem [**XU2022103620**], [**ZHANG2020102861**]
  - Fleet Management [**TURAN2020102829**]
  - Transportation Systems [Survey article] [**WANG2019144**]

- Public Transit
  - Taxis [**CHEN2021103272**], [**JIAO2021103289**], [**KE2021102858**], [**MAO2020102626**], [**QIN2021103239**], [**SHOU2020102738**], [**TANG2020102844**], [**YU2022103640**],
  - Public Transit [**CHOW2021103264**], [**FENG2022103611**], [**LIU201918**], [**MULLERHANNEMA**
    [**TANG2021102951**], [**WANG2019387**], [**WANG2020102661**], [**ZHANG2021102928**]

- Pedestrians and Passengers
  - Pedestrians [**BUSTOS2021103018**], [**HIDAKA2019115**]
  - Bicycles and Scooters [**ITO2021103371**], [**LV2021103404**], [**ZUNIGAGARCIA2022103660**],
  - Travel Demand Modeling [**HAFEZI2021102972**], [**KIM2022103523**], [**LI2021102921**], [**LIU2021103361**], [**PANG2020102706**]

- Planes, Trains, and Boats
  - Railway Maintenance [**ALLAHBUKHSH201935**]
  - Railway Traffic Control [**GHASEMPOUR202091**], [**TANG2022103679**]
  - Train Delays [**LI2022103606**], [**NAIR2019196**]
  - Air Traffic Management [**BAO2021103323**], [**ALBABA2021103417**], [**CORRADO2021103331**], [**DESHMUKH2021103036**], [**DHIEF2022103704**], [**DU2021103122**], [**KHAN2021103225**], [**OLIVE2020102737**], [**PANG2021103326**], [**PHAM2022103463**], [**SCHULTZ2021103119**], [**VERDONKGALLEGO2019356**], [**WANG2021102892**], [**ZHU2021103179**],
  - Ships [**GUMUSKAYA2021103383**], [**LU2021102970**],

- Crashes
  - Inferring Pre-Crash Impact Data [**CHEN2021103009**],

- Back End (No Application)
  - Generative Modeling [**BORYSOV201973**], [**GARRIDO2020102787**]
  - Preference Learning [**ZHU2020102849**]
  - Extracting Economic Information (?) [**WANG2020102701**]
  - Graphs [**RODRIGUEZDENIZ2022103556**]
  - Discrete Choice Models [**SFEIR2022103552**]
  - Fairness in Artificial Intelligence [**ZHENG2021103410**]
  - Discrete Choice Modeling [**WONG2021103050**]

2.9.2 Articles whose abstracts refer to imbalanced data

Chen [**CHEN2020102646**] talks about resampling using SMOTE and Tomek. Used LightGBM classifier.

Cai [**CAI2020102697**] used the deep convolutional generative adversarial network (DCGAN). Compared four models, logistic regression model, support vector machine, artificial neural network, and convolutional neural network.

Emarani Abou Elassad [**ELAMRANIABOUELASSAD2020102708**] works with several imbalanced methods. Use this paper as a model.

Yu [**YU2020102740**] used focal loss for real-time crash prediction.

Shi [**SHI2021103414**] uses the Grey Wolf Optimizer and SMOTE to balance the data.

Khan [**KHAN2021103225**] used SMOTE and "average balanced recall accuracies,"

Chen [**CHEN2022103709**] uses bagging.

Anomalous events might also use imbalanced data.

2.9.3 Crashes

Twenty-one articles in TR_C have 'crash', 'accident', 'ambulance', 'hospital', 'fatal', or 'injury' in the keywords. Another forty have them in the abstract. I'm really only interested in ones that use real data, not simulation.

Kalatian [**KALATIAN2021102962**] studies interactions between pedestrians and autonomous vehicles.

Cai and Abdel-Aty [**CAI2020102697**] do similar work to ours with machine learning.

Emarani Abou Elassad [**ELAMRANIABOUELASSAD2020102708**] was mentioned above as a model paper. Also applied to crashes.

Yu [**YU2020102740**] mentioned above.

Chapter 3

Lit Review: Data Cleaning

3.1 Cleaning Techniques Used in Crash Analysis Studies

In "A deep learning based traffic crash severity prediction framework" by Rahim (LSU) [**RAHIM2021106090**], they just deleted any records with missing or inconsistent data. The *Titanic* Kaggle sites you showed me use several other methods for filling in incomplete data. Write a paper where I compare different methods for dealing with missing data, and their effect on different metrics (precision, recall, accuracy, sensitivity, f1, false alarm rate) of the classification of the test set.

Rahim's article took out 37% of the records for missing or inconsistent data, but only 21% of the fatal crashes; could that imbalance in the data cleaning skew the model prediction? It makes sense that police would be more meticulous in their record keeping for fatal crashes, but 21% and 37% are huge.

Would we get a better model if we found a good way to fill in missing data?

Chapter 4

Lit Review: Methods for Imbalanced Data

4.1 Algorithm Level Approaches

4.1.1 Some Papers

- Recognition-based: Learning from one class rather than discrimination-based, doing unsupervised learning on the minority class. [**CHAWLA˙2004**]
- Fuzzy rule-based classification systems (what is this?) [**CHABBOUH˙2019**] [**DABLAIN˙2021**] [**MAHMUDAH˙2021**] [**ZHAI˙2020**] [**ZHAI˙2020˙D2GAN**]
- In decision trees, using evolutionary/genetic methods instead of greedy search [**CHABBOUH˙2019**] [**WEISS˙2000**]
- Clustering and Subspace Modeling [**CHEN˙2011**]

4.1.2 Genetic Algorithms

In this short 2000 paper, Weiss [**WEISS˙2000**] used a genetic algorithm to predict rare events. Borrowing from simulated annealing, they varied the relative importance of precision and recall at each step of the genetic algorithm.

4.1.3 Subspace Model

Chen 2011 [**CHEN˙2011**]

This was fascinating and entirely different from anything I've seen.

1. Separate the training data Tr into negative (majority) and positive (minority) classes TrN and TrP.
2. Let K be the ratio of negative to positive samples, in my case about 100, so that if you divide the majority class TrN into K groups, each will have about the same number of samples as the minority class.

3. Use K-means clustering to separate the negative (majority) class TrN into K groups; each of the groups is a cluster of the negative (majority) class.
4. For each of the K groups TrN_i:
   - Combine the negative elements of the group with the entire positive (minority) class TrP to form a balanced subspace.
   - Train the model for the subspace
5. Recombine the K subspace models with a model trained on the entire data set to build an integrated model.

4.2 Metrics

4.2.1 The Problem: Imbalanced Data Set

In an unbalanced data set, the number of actual negatives ($N = TN + FP$) is much different from the number of actual positives ($P = FN + TP$). In our case, if our independent variable is fatal crashes, the negatives are 99.574714% of the data set, and the positives are just 0.425286%.

The standard metrics get thrown off by the imbalance. If we predict that every crash is nonfatal, we have accuracy of 99.57%, which sounds really impressive.

The recall (true positive rate) is not thrown off by an imbalanced data set, because it only works with TP and FN, the actual positives. Similarly for specificity (true negative rate).

The precision is thrown off by an imbalanced data set, because it works with both a subset of the actual positives (TP) and a subset of the actual negatives (FP).

4.2.2 Standard Metrics

		Prediction	
		N	P
Actual	N	TN	FP
	P	FN	TP

$$\text{Accuracy} = \frac{TN + TP}{TN + FP + FN + TP}$$

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

$$\text{Specificity, Selectivity, or TNR} = \frac{TN}{TN + FP}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

4.2.3 Balanced Precision and Balanced f1 in the Penalty Function

Most ML algorithms work using a *penalty function* that measures how bad the current solution is, then iteratively improving the solution in the direction that minimizes the penalty. We should be able to write a custom penalty function.

Update: How-to instructions for changing the metrics in `scikit-learn`. The example is how to use recall instead of accuracy.

`https://stackoverflow.com/questions/54267745`

Recall only deals with the minority class, so the balance of the data set doesn't matter. *Precision*, on the other hand, takes results from both classes, so we can balance it by scaling the count of False Positive results, giving a *Balanced Precision* metric. From Recall and Balanced Precision we can get a *Balanced f1* metric.

If our penalty function uses balanced precision and balanced f1, it may not matter that our data set is imbalanced, and we can use all of, and only, the original data to build our model.

4.2.4 Balanced Precision in the Literature

Balanced Accuracy frequently appears in the literature. I have not found *balanced precision* in the literature. Two possible reasons. Either nobody has thought of it, or they did, found it not useful, and abandoned the idea.

Update: `imbalanced-learn` has more metrics than *scikit-learn*, but still no balanced precision.

`https://imbalanced-learn.org/dev/metrics.html`

4.2.5 Balanced Accuracy

There is a metric called *balanced accuracy*. You get it from the definition of *accuracy* by multiplying the actual negative elements (TN and FP) by the ratio of the positives to negatives,

$$\frac{P}{N} = \frac{FN + TP}{TN + FP}$$

so that the total number of actual negatives and total number of actual positives in the sample are equal.

[I suppose you could also get it by multiplying the actual positive elements (FN and TP) by the reciprocal.]

I got this derivation by intuiting about what I would want *balanced accuracy* to mean, and it matches the definition I found in Wikipedia.

`https://en.wikipedia.org/wiki/precision_and_recall#Imbalanced_data`

Wikipedia says [I'm sure I can find a more authoritative source.]

$$\text{Balanced Accuracy} = \frac{TPR + TNR}{2}$$

$$\begin{aligned}
\text{Recall or TPR} &= \frac{TP}{TP + FN} \\
\text{Specificity or TNR} &= \frac{TN}{TN + FP} \\
\text{Accuracy} &= \frac{TN + TP}{TN + FP + FN + TP} \\
\text{Balanced Accuracy} &= \frac{TN \cdot \frac{P}{N} + TP}{TN \cdot \frac{P}{N} + FP \cdot \frac{P}{N} + FN + TP} \\
&= \frac{TN \cdot P + TP \cdot N}{TN \cdot P + FP \cdot P + FN \cdot N + TP \cdot N} \\
&= \frac{TN \cdot P + TP \cdot N}{(TN + FP) \cdot P + (FN + TP) \cdot N} \\
&= \frac{TN(FN + TP) + TP(TN + FP)}{(TN + FP)(FN + TP) + (FN + TP)(TN + FP)} \\
&= \frac{TN(FN + TP) + TP(TN + FP)}{2(TN + FP)(FN + TP)} \\
&= \frac{TN(FN + TP)}{2(TN + FP)(FN + TP)} + \frac{TP(TN + FP)}{2(TN + FP)(FN + TP)} \\
&= \frac{TN}{2(TN + FP)} + \frac{TP}{2(FN + TP)} \\
&= \frac{TNR + TPR}{2}
\end{aligned}$$

4.2.6 Balanced Precision

I haven't found *balanced precision* in a brief Google search, although Google knows the kind of stuff I look up and sent me to articles on balanced accuracy. Finding it will take some work, because "balanced precision" has different meanings in other tech fields.

We can make balanced precision the same way we made balanced accuracy, by taking the actual negative results (TN and FP) and scaling them so that the total number of actual negatives equals the total number of actual positives, by multiplying by $\frac{P}{N} = \frac{FN+TP}{TN+FP}$.

Is this related to the G-mean? [No]

$$\text{G-mean} = \sqrt{\text{Precision} \times \text{Specificity}}$$

$$
\begin{aligned}
\text{Precision} &= \frac{TP}{TP + FP} \\
\text{Balanced Precision} &= \frac{TP}{TP + FP \cdot \frac{P}{N}} \\
&= \frac{TP \cdot N}{TP \cdot N + FP \cdot P} \\
&= \frac{TP(TN + FP)}{TP(TN + FP) + FP(FN + TP)} \\
&= \frac{TP(TN + FP)}{TP(TN + FP) + FP(FN + TP)} \\
&= \dots
\end{aligned}
$$

Giving up here on finding some nice, concise connection between Balanced Precision and other metrics.

4.2.7 Balancing Two Metrics: F1 and Gmean

From Elassad 2020: [**ELAMRANIABOUELASSAD2020102708**]

> F1 score, is a highly informative measure as it considers both precision and recall measures, which makes it very suitable for imbalanced classification (Qian et al., 2014; Sun et al., 2018); it's deemed to be a special measure that conveys the balance between the precision and recall in order to find an effective and efficient trade-off. Another useful metric is G-mean, which is considered as a metric of stability between correct classification of positive class and negative class viewed independently. It is usually adopted in order to resist the imbalances in the dataset (Kubat et al., 1997).

F1 Metric

F1 is the harmonic mean of Precision and Recall.

$$
\text{F1} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}
$$

Gmean

Gmean is the geometric mean of Precision and Specificity (TNR).

$$\begin{aligned}
\text{Precision} &= \frac{TP}{TP+FP} \\
\text{Specificity, Selectivity, or TNR} &= \frac{TN}{TN+FP} \\
\text{Gmean} &= \sqrt{\text{Precision} \times \text{Specificity}} \\
&= \sqrt{\frac{TP}{TP+FP} \times \frac{TN}{TN+FP}}
\end{aligned}$$

4.3 Loss Functions

4.3.1 Binary Cross-Entropy Loss Function

Let's say we have an imbalanced data set with 100 negative samples for each positive sample.

For binary classification, the first three (class weights, weighted loss function, and naive oversampling) are effectively the same in the training phase. The cross-entropy loss function,

$$loss = \sum_{i=1}^{n} y_i \log(p_i) + (1 - y_i) \log(1 - p_i)$$

for binary classification is

$$loss = \sum_{y_i=1} \log(p_i) + \sum_{y_i=0} \log(1 - p_i)$$

which is the sum of the logs of the errors in predictions for the negative class plus the sum of the logs of the errors in predictions for the positive class.

4.3.2 Class Weights and α-weighted Loss

If the classes are imbalanced, like there are 100 times as many samples with $y = 0$ as samples with $y = 1$, then the loss function is mostly summing how bad the predicting probability is for the majority class and largely ignoring the minority class. Both the class weights parameter and a weighted loss function fix this by multiplying one or the other by some compensating factor.

$$loss = 100 \times \sum_{y_i=1} \log(p_i) + \sum_{y_i=0} \log(1 - p_i)$$

This multiple gives the two classes equal weight in the loss.

In the α-weighted cross entropy,

$$loss = \sum_{i=1}^{n} \alpha y_i \log(p_i) + (1 - \alpha)(1 - y_i) \log(1 - p_i)$$

let $\alpha = \frac{100}{100+1}$ and you'll get the same thing, within a positive constant multiple.

$$loss = \sum_{i=1}^{n} \frac{100}{101} y_i \log(p_i) + \frac{1}{101}(1 - y_i) \log(1 - p_i)$$

$$loss = \frac{1}{101} \sum_{i=1}^{n} 100 y_i \log(p_i) + (1 - y_i) \log(1 - p_i)$$

The only difference I can ascertain between the class weights parameter and a weighted loss function is that the class weights aren't used with the validation set.

4.3.3 Oversampling

4.3.4 Naive Oversampling

Naive oversampling would be to create 99 copies of each of the positive samples, so that the two sets are balanced. That would have exactly the same effect on the loss function, because there would now be 100 times as many samples with $y_i = 1$.

4.3.5 Class Weights v/s Naive Oversampling: They're the Same

I had an insight on why these things are the same. Let's say you have an imbalanced data set, with 100 times as many negative samples as positive samples.

In Naive Oversampling, you make 100 copies of each of the positive samples and run regular cross-entropy loss.

In weighted Class Entropy, you multiply the positive-class losses by 100.

$$loss = 100 \times \sum_{y_i=1} \log(p_i) + \sum_{y_i=0} \log(1 - p_i)$$

These two approaches different in execution but the same in result because, as I often remind my students, multiplying something by 100 is the same as adding it to itself 100 times.

4.3.6 Focal Loss

Introduced by Lin in 2017. [**LIN˙2020**]

Yu 2020 [**YU˙2020**] adapts α-weighted cross entropy and focal loss to crash analysis.

In the focal loss function,

$$loss = \sum_{i=1}^{n} \alpha(1 - p_i)^{\gamma_1} y_i \log(p_i) + (1 - \alpha) p_i^{\gamma_2} (1 - y_i) \log(1 - p_i)$$

$$loss = \sum_{y_i=1} \alpha(1 - p_i)^{\gamma_1} \log(p_i) + \sum_{y_i=0} (1 - \alpha) p_i^{\gamma_2} \log(1 - p_i)$$

if $\gamma_1 = \gamma_2 = 0$, then it's the same as the α-weighted loss function.

In the original focal loss paper by Lin [**LIN˙2020**], γ_1 and γ_2 are the same.

For samples with $y_i = 1$, the minority class, here are values of $(1 - p_i)^{\gamma_1} \log(p_i)$ for different values of p_i and different values of γ_1. I got the range of values of $\gamma_1 \in \{0, 0.5, 1, 2, 5\}$ from Lin's 2018 paper that proposed focal loss.

		γ_1				
$(1-p_i)^{\gamma_1}\log(p_i)$		0	0.5	1	2	5
	0.1	-3.32	-3.15	-2.99	-2.69	-1.96
	0.3	-1.74	-1.45	-1.22	-0.85	-0.29
p_i	0.5	-1	-0.71	-0.5	-0.25	-0.03
	0.7	-0.51	-0.28	-0.15	-0.05	0
	0.9	-0.15	-0.05	-0.02	0	0

If $\gamma_1 > 0$, then for samples in the positive class, the loss is negligible for good predictions (p_i close to 1), so it focuses the loss on poor predictions.

Yu applied focal loss in the crash literature.[**YU˙2020**]

4.3.7 Optimizing F_β

Loss functions for gradient-based learning need to be differentiable (?), and the F_β score is not differentiable, so this 2021 article by Lee [**LEE˙2021**] proposes a differentiable surrogate loss function that optimizes the F_β score.

With imbalanced data, using a loss function that optimized F_β instead of accuracy would let you balance precision and recall, fixing one aspect of the imbalance problem.

$$F_\beta = \frac{(1+\beta^2)\cdot \text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Precision}) + \text{Recall}} = \frac{1}{\frac{\lambda_\beta}{\text{Recall}} + \frac{1-\lambda_\beta}{\text{Precision}}}, \qquad \lambda_\beta = \frac{\beta^2}{1+\beta^2}$$

The article takes a deep dive into loss functions. I should master it.

4.3.8 Tree-Based Methods

Pendault [**PEDNAULT˙2000**] has a 2000 article on insurance risk modeling that incorporates "a domain-specific optimization criterion... to identify suitable splits during tree building." It assigns different weights to *claim* and *nonclaim* records. Because that strategy helps but does not entirely solve the imbalanced data problem, they also have a split criterion that prevents splits of really small branches, "splinter groups," that are unlikely to contain any elements of the minority class because the minority class is so sparse.

4.3.9 α-weighted Binary Cross-Entropy Loss Function as Ethical Tradeoff

From `Brads_Report_10_25_21`

I made a [perhaps paper-worthy?] connection between the loss function I want and the α-weighted binary cross-entropy loss function, which is widely known and widely implemented, but, according to Yu's paper, not before used in crash-related modeling.

Matrix

	Do Not Send Ambulance $h_\theta(x_i) < 0.5$	Send Ambulance $h_\theta(x_i) > 0.5$
Do Not Need Ambulance $y_i = 0$	TN	FP
Need Ambulance $y_i = 1$	FN	TP

Switching between Binary and Continuous

In the binary cross-entropy loss function,

$$J = -\sum_{i=1}^{N} y_i \log(h_\theta(x_i)) + (1 - y_i) \log(1 - h_\theta(x_i))$$

the y_i are binary, $y_i \in \{0, 1\}$, but the model predictions, $h_\theta(x_i)$, are a probability, $h_\theta(x_i) \in (0, 1)$.

If we treat the model predictions as binary, replacing

$$\log(h_\theta(x_i)) \to \begin{cases} 0 & \text{if } h_\theta(x_i) <= 0.5 \\ 1 & \text{if } h_\theta(x_i) > 0.5 \end{cases}$$

and

$$\log(1 - h_\theta(x_i)) \to \begin{cases} 0 & \text{if } 1 - h_\theta(x_i) <= 0.5 \\ 1 & \text{if } 1 - h_\theta(x_i) > 0.5 \end{cases}$$

then

$$TP = \sum_{i=1}^{N} y_i \log(h_\theta(x_i))$$

$$TN = \sum_{i=1}^{N} (1 - y_i) \log(1 - h_\theta(x_i))$$

and the loss function becomes $J = -(TP + TN)$

Why do we use the continuous instead of the binary in the loss function? Because we want the predictions to be robust, so that when we use the model on unseen data, we can be more certain

that it will correctly classify new instances. The binary, however, are much easier to explain to non-technical people, or even technical people in other fields.

Scenario

The medical ethicists and politicians decide on a number, p, such that we are willing to automatically dispatch p ambulances when they aren't needed in order to send one ambulance when it is needed. We want

$$\frac{\Delta FP}{\Delta TP} \leq p$$

Binary h_θ

Our loss function is

$$FP - p \cdot TP$$

Continuous h_θ

Use the α-weighted cross-entropy loss function, as in Yu's paper and widely available.

$$J = -\sum_{i=1}^{N} \alpha y_i \log(h_\theta(x_i)) + (1-\alpha)(1-y_i)\log(1-h_\theta(x_i)), \quad \alpha = \frac{p}{p+1}$$

Why are these equivalent?

Adding a constant to the loss function, or multiplying it by a positive constant, does not change its effect.

$FP - p \cdot TP$ is equivalent to $FP - p \cdot TP + (TN + FP)$, because $TN + FP$ is constant, so $FP - p \cdot TP$ is equivalent to $-(p \cdot TP + TP)$.

$$FP - p \cdot TP$$

$$-(p \cdot TP + TN)$$

Multiplying by $\frac{1}{p+1}$ gives an equivalent loss function, because $\frac{1}{p+1} > 0$.

$$-\frac{p \cdot TP + TN}{p+1}$$

$$-\left(\frac{p}{p+1}TP + \frac{1}{p+1}TN\right)$$

$$-\left(\frac{p}{p+1}TP + \left(1 - \frac{p}{p+1}\right)TN\right)$$

$$-(\alpha TP + (1-\alpha)TN)$$

The continuous version of TP is $\sum_{i=1}^{N} y_i \log(h_\theta(x_i))$

The continuous version of TN is $\sum_{i=1}^{N} (1-y_i) \log(1-h_\theta(x_i))$

$$J = -\sum_{i=1}^{N} \alpha y_i \log(h_\theta(x_i)) + (1-\alpha)(1-y_i)\log(1-h_\theta(x_i)), \quad \alpha = \frac{p}{p+1}$$

Paper Focus

Yu et al introduced to the crash-analysis field the alpha-weighted cross-entropy loss function to deal with imbalanced data. We propose another application of the alpha-weighted loss, to encode and implement tradeoffs that come from our ethical/political values decided by community leaders.

4.4 Data Level Methods

4.4.1 Imbalanced Cleaning: Tomek and Condensed Nearest Neighbor

Batista [**BATISTA˙2004**] uses two imbalanced cleaning method called *Tomek links* and *Condensed Nearest Neighbor*. If examples from the majority and minority class are close to each other, it deletes the majority samples. One could think of it as targeted undersampling of the majority set.

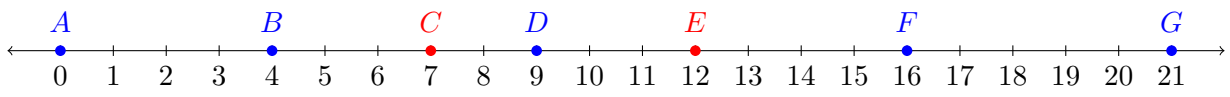
Imbalanced-Learn, an add-on to Scikit-Learn, has these algorithms read to use. Tomek and Wilson's papers introducing these algorithms are from the 1970's.

4.4.2 Tomek's Links

This method undersamples the majority-class samples, eliminating ones that are too close to minority-class samples, presuming them to be noise, and helping clarify clusters of minority samples.

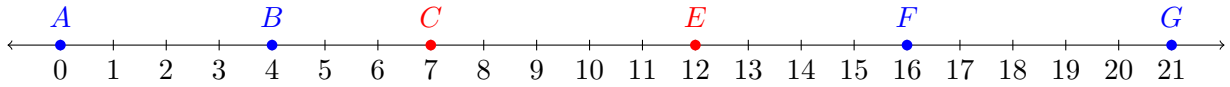
A pair of samples are a *Tomek link* if one is majority and one minority, and they are each other's nearest neighbors. To use Tomek's inks as an undersampling strategy for imbalanced data, delete the positive sample in each Tomek's link. Other cleaning strategies (for balanced sets) would eliminate both the positive and negative.

It is possible to iterate Tomek's several times. Here's an example of how it works in one round and in a second round. The blue samples are from the majority set and the red are from the minority. Assume that these seven points are a small part of a large dataset, but these are the only points in this region.

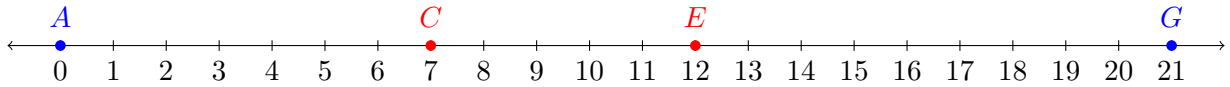


In the original dataset, C and D are each other's nearest neighbors, C from minority and D from majority, so they are a Tomek link. On the other hand, D is the nearest neighbor to E, but E is not D's nearest neighbor, so they are not a Tomek link.

Eliminate sample D.



Now the pairs (B, C) and (E, F) are Tomek links, so if we ran Tomek undersampling a second time, we would remove samples B and F.



Now C and E are each other's nearest neighbors and of the same (minority) class, so this part of the dataset would not change under another run of Tomek.

The idea of Tomek assumes that the minority samples should cluster, and any majority samples in or near those clusters must be noise, so we can eliminate them. We now have a clear cluster of two minority samples with no close majority samples.

I saw multiple runs of Tomek mentioned [somewhere] in my reading, so I tried it on the crash data, running it up to five times, and saw that it converged, with fewer positive samples eliminated in each round. I had conjectured that a negative sample in a Tomek link in a later round must have been a negative sample in a Tomek link in an earlier round, digging itself out of a field of positive-class dust, but I suspected that there might be (perhaps unusual) cases where one minority-class sample (C in the example above) created a Tomek link, and eliminating the majority-class sample in that link (D above) allowed a Tomek link for a different minority-class sample (E above). I then played with it until I found a counterexample to my conjecture, so the conjecture, that a minority-class sample in a Tomek link in a later round of Tomek undersampling must have been in a Tomek link in every previous round of the Tomek undersampling, is false.

If the conjecture had been true, then we could greatly speed up subsequent rounds of Tomek undersampling by only considering the minority samples in Tomek links in the previous round. That would not be thorough, but this approach would.

Algorithm for Repeated Application of Tomek's Links

For the first round of Tomek undersampling, one has to consider each element of the minority class. In the Tomek's links, call the minority-class elements $\{A_1, A_2, \ldots, A_{n1}\}$, and the majority-class elements $\{B_1, B_2, \ldots, B_{n1}\}$. Tomek undersampling for minority classes eliminates all of $\{B_1, B_2, \ldots, B_{n1}\}$.

In the second round of Tomek undersampling, we only need to consider as possible Tomek links the nearest neighbors of $\{A_1, A_2, \ldots, A_{n1}\}$ and any element of the minority class that had one of $\{B_1, B_2, \ldots, B_{n1}\}$ as its nearest neighbor.

In subsequent rounds, consider the minority-class samples from the Tomek's links from the previous round, and the elements of the minority class that had as their nearest neighbor an element of the majority class in the Tomek's links.

In theory there could be more Tomek's links in one round than in the previous round, but in practice they go to zero and the set converges to a set with no Tomek's links.

4.4.3 Cleaning Multiclass Data

Wei (2021) [**WEI˙2021**] uses something similar to Tomek's links for a multi-class problem with a majority class and multiple minority classes.

- Splits an imbalanced multi-class problem with $n+1$ classes (n of them being minority) into n imbalanced binary problems for data cleaning.
- Uses cleaning undersampling (similar to Tomek's Links) to remove noisy spots in the data.

4.4.4 Oversampling

Naive oversampling would be to create 99 copies of each of the positive samples, so that the two sets are balanced. That would have exactly the same effect on the loss function, because there would now be 100 times as many samples with $y_i = 1$.

4.4.5 Underssampling

Undersampling would erase 99% of the negative samples so that the classes would be balanced. That seems like a bad idea, because you would lose a lot of information about the majority class.

4.4.6 SMOTE: Synthetic Minority Oversampling TEchnique

Especially if we're doing fatalities, we have a terribly imbalanced data set. Ideally we'd like to have an equal number of fatal and nonfatal crashes to plug into our ML algorithm, but we have about 0.47% fatal and 99.53% nonfatal.

One solution is to randomly choose 681 nonfatal crashes to compare with our 681 fatal crashes, but that leaves behind a LOT of information.

Many of the papers I've read use SMOTE, which balances the data set by creating synthetic elements for the minority set (fatal crashes). It picks an element of the minority set, a, and picks one of its nearest neighbors, b, and creates a new synthetic element c. For each data category, D_i, in which they differ, SMOTE chooses $D_i(c)$ to be between $D_i(a)$ and $D_i(b)$. It randomly chooses a random number $r \in [0, 1]$, and makes $D_i(c) = D_i(a) + r(D_i(a) - D_i(b))$.

I get how that works for continuous variables. I get that it would work if $D_i(a)$ and $D_i(b)$ weren't very different.

How would that work for boolean variables? SMOTE would choose nearest neighbors a and b that agree on most variables, but for values of i where $D_i(a) = 0$ and $D_i(b) = 1$, it would randomly choose $D_i(c) \in \{0, 1\}$. There is no *between* for boolean variables. It doesn't seem to me that it would work as well.

Original SMOTE only works with continuous variables. There is something called SMOTE-NC that handles continuous and categorical, but it has to have some continuous variables to work on.

Unlike SMOTE, SMOTE-NC for dataset containing numerical and categorical features.
However, it is not designed to work with only categorical features.

`https://imbalanced-learn.org/dev/references/generated/imblearn.over_sampling.SMOTENC.html`

Since we have ≈ 200 times as many nonfatal crashes as fatal crashes, to balance the data set with SMOTE, we would have to make two hundred synthetic elements for each fatal crash. It seems to me that we would be making a mess of our data set.

4.4.7 Flavors of SMOTE

SMOTE, or Synthetic Minority Oversampling TEchnique, [**CHAWLA˙2002**] creates extra samples of the minority class, but rather than making exact copies, it finds two similar samples and creates more samples "between" them, with feature values between the values of the two samples. SMOTE only works for continuous features, not for categorical features. Almost all of my features are categorical.

I got this list of flavors of SMOTE from a 2021 review by Mahmudah. [**MAHMUDAH˙2021**] I've investigated some of them and given some flesh to some parts of this skeleton.

- SMOTE: Synthetic Minority Oversampling TEchnique [**CHAWLA˙2002**]
  Uses k-nearest neighbors to find two close positive (minority) samples, and creates a synthetic sample between them. Works on continuous data, not on categorical or binary data.
- ADASYN: ADAptive SYNthetic sampling approach for imbalanced learning. [**MAHMUDAH˙2021**]
  Creates synthetic samples based on the level of difficulty in learning the samples of the minority class. A positive samples is "difficult" if it has more negative samples as its nearest neighbors. The more difficult a sample is, the more synthetic copies of that sample ADASYN creates.
- Borderline SMOTE [**MAHMUDAH˙2021**]
  Generates synthetic positive samples along the border between the positive and negative classes. Brad's Question: This assumes you know where the border is. I suppose you could do it iteratively.
- Safe-level SMOTE [**MAHMUDAH˙2021**]
  When SMOTE finds the nearest positive-class neighbors of a positive sample, it ignores the negative (majority-class) neighbors. [I think this is what it means:] Creating synthetic positive-class samples in a neighborhood with lots of negative samples just makes more of a mess, so this is not considered a "safe" place to make synthetic samples. Safe-level SMOTE creates synthetic positive samples only in majority-positive neighborhoods.
- Relocating-safe-level SMOTE (RSLS) [**MAHMUDAH˙2021**]
  Avoids creating synthetic positive samples near negative samples.
- Density-based SMOTE (DBSMOTE) [**MAHMUDAH˙2021**]

Integration of DBSCAN and SMOTE. DBSCAN, Density-Based Spatial Clustering of Application with Noise, discovers clusters with an arbitrary shape (?) DGSMOTE creates synthetic samples at the pseudo-centroids of the clusters of positive samples.

- Adaptive Neighbor SMOTE (ANS) [**MAHMUDAH˙2021**]
  Focuses not on -where- to generate synthetic samples, but on -how many- samples to generate in a particular neighborhood.
- D2GAN
  This 2020 article by Zhai [**ZHAI˙2020˙D2GAN**] builds on the Dual Discriminator Generative Adversarial Nets (D2GAN) paper from 2017 by Nguyen [**NGUYEN˙2017**]. They want to do better oversampling, comparing D2GAN with SMOTE. I don't understand what this is, but they say SMOTE has three drawbacks:

  1. Ignores the probability distribution of minority class samples.
  2. Synthetic examples lack diversity.
  3. Interating SMOTE many times will give synthetic samples with significant overlap.

  This 2022 article by Zhai [**ZHAI˙2022**] slightly modifies Zhai's claims against SMOTE.

  1. Does not extend the training field of positive samples.
  2. Synthetic examples lack diversity.
  3. Does not accurately approximate the probability distribution of minority class samples.

  The authors propose two new methods of diversity oversampling by generative models, one based on "extreme machine learning autoencoder," and the other based on generative adversarial networks (GAN).

4.4.8 Train/Test Split

The application in Sharififar's 2019 article [**SHARIFIFAR˙2019**] is digital mapping of farmland, categorizing areas by soil type. Some soil types are rare but significant. This is the first article I've seen that, at the beginning, says that making sure each minority class appears in appropriate distribution in the validation and test sets is an important challenge. They explicitly say that they split 30% for the validation set by taking 30% of each class.

4.4.9 Feature Selection

This 2012 article by Tan [**TAN˙2012**] introduces a feature selection model specifically for imbalanced data sets. I haven't dug in yet.

4.5 Bagging and Boosting

- Boosting and Bagging [**BATISTA˙2004**] [**CHABBOUH˙2019**] [**DABLAIN˙2021**] [**MAHMUDAH˙202**
  [**SHARIFIFAR˙2019**]

4.6 Lit Review: Medium.com *Towards Data Science* Articles

These aren't exactly peer reviewed, but they're current.

Soleymani (4/1/22) says that class weights are more effective than SMOTE, and gives an example of why SMOTE doesn't do what you think it should. [**Soleymani˙TDS˙04˙01˙2022**]

Raj (9/5/19) is a brief article that introduces what an imbalanced data set is, and resampling, including naïve oversampling, undersampling, and SMOTE. [**Raj˙TDS˙09˙05˙19**]

Soni (10/9/20) introduces Balanced Random Forest, with code, in addition to undersampling and oversampling. Balanced Random Sampling is, I think, a form of bagging. You take a bootstrap sample of the minority class and the same number of elements from the majority class, and run random forest; then aggregate the results. [**Soni˙TDS˙10˙09˙20**]

Brownlee isn't in TDS, but gives an easy introduction to ROC curves. [**Brownlee˙TDS˙11˙26˙14**] Also gives good references in [**Brownlee˙TDS˙08˙19˙15**].

Stewart also mentions Tomek Links. [**Stewart˙TDS˙07˙01˙22**]

Bordia reviews variants of SMOTE, including SMOTE_NC, which works with datasets with some (but not all) categorical data and some continuous data. NC is for Nominal and Continuous. [**Bordia˙TDS˙02˙25˙22**]

Boyle recommends Random Forests for imbalanced data. [**Boyle˙TDS˙02˙03˙19**]

Keras can do random forest classifiers, although you may need to make it yourself. `https://keras.io/examples/structured_data/deep_neural_decision_forests/`

How to do an ROC curve and find AUC for Keras and sklearn: `https://medium.com/hackernoon/simple-guide-on-how-to-generate-roc-plot-for-keras-classifier-2ecc6c73115a`

Badr includes bagging. [**Badr˙TDS˙02˙22˙19**]

Rocca gives many different ideas. Read this one carefully. [**Rocca˙01˙27˙19**]

Lador gives good examples of when different metrics are useful. [**Lador˙TDS˙09˙05˙17**]

Jaitley also recommends Random Forest, Gradient Boosting, and AdaBoost. [**Jaitley˙02˙01˙19**]

Ahamed had entirely different recommendations, Ensemble Cross-Validation (CV), Class Weights, and Over-Predicting the class of the minority class, *i.e.* setting a lower probability threshold for the minority class. [**Ahamed˙04˙15˙18**]

Chapter 5

Lit Review: Datasets

5.1 Crash Datasets

- SHRP2, Strategic Highway Research Program 2, Naturalistic Driving Study
  Federal Department of Transportation
  Most cited dataset.
- Second Highway Research Program (Data Set)
  I have an account.
- Virginia 100-car Database
- Next Generation Simulation, NGSIM Trajectory Data https://iswitrs.chp.ca.gov/Reports/jsp/index.jsp
- NASS-CDS: National Automotive Sampling System – Crashworthiness Data System
- Canada's National Collision Database
- Michigan Safety Pilot
- Roadway Information Database (RID)
- Shanghai Naturalistic Driving Study
- California Statewide Integrated Traffic Records System (SWITRS)
  Apparently anyone can get an account?
  `https://iswitrs.chp.ca.gov/Reports/jsp/index.jsp`
- Highway Safety Information System
  Not updated since 2018?
  `http://www.hsisinfo.org`

From `24_May_2021_Report`:

5.1.1 Jargon to Understand

- Naturalistic Driving Data - Data collected from sensors installed in the driver's own car, trying to get as close as possible to the driver's "natural" behavior.
- Heterogeneity. I understand vaguely what "data heterogeneity" means, but I'm going to watch for the term to see how it's used in the context of these papers.

5.1.2 IRB, SHRP Database

Eleven of the papers in *Accident Analysis and Prevention* used the Strategic Highway Research Program 2 (SHRP2) Naturalistic Driving Study (NDS), which put sensors in 3400 cars and recorded five million trips, including crashes. To get "Qualified Researcher Status" with "full access to data that has been made available through the SHRP 2 NDS Data Access Website," I had to submit a certificate of training on research with human subjects. I did the training through the UL Institutional Review Board (IRB). I now have access.

5.1.3 NGSIM Database

Three papers use the Next Generation Simulation dataset from the US Dept of Transportation, and it's available for download with no restrictions.

5.2 Datasets with Imbalanced Data

5.2.1 Datasets, Annotated

5.2.2 Articles using These Datasets

Zheng 2021 [**ZHENG˙2021**]

- Oversampling, undersampling, and hybrid methods use random sampling ratios. [What? How? I thought the user set the sampling ratios.]
- This paper proposes three algorithms to automatically set the sampling ratios using genetic algorithms.
- Used fourteen datasets, some of which may be useful benchmark datasets.

Wang 2021 [**WANG˙2021**]

- Uses seven benchmark imbalanced datasets from the UCI machine learning repository
- Implicit regularization for dynamic ensemble selection of classifiers.

5.2.3 Database Repositories

UCI Machine Learning Repository

`https://archive.ics.uci.edu/ml/about.html`

Chapter 6

Lit Review: Seminal and Interesting Papers

6.1 Seminal Papers

- Lin [**LIN˙2020**] introduced Focal Loss in 2017. The 2017 versions of this article are only available through Inter Library Loan, because the UL Library apparently doesn't subscribe to IEEE, and the version I found was from 2020.

6.2 Review Papers

6.2.1 Chawla

Chawla [**CHAWLA˙2004**] gives an overview of the state of the field in 2004.

- Data Methods
  - Random Oversampling with Replacement
  - Random Oversampling
  - Directed Oversampling
    No new examples are created, but the choice of which ones to replace is informed rather than random.
  - Directed Undersampling
  - Oversampling with informed generation of new samples
  - Combinations of the above
- Algorithmic Methods
  - Adjusting class costs
  - Adjusting the probabilistic estimate at the tree leaf (for tree methods)
  - Recognition-based methods (learning from one class) rather than discrimination-based.
- Issues at 2000 Conference

–

- Issues at 2003 ICML Conference
  - Probabilistic estimates
  - Pruning
  - Threshold adjusting
  - Cost-matrix adjusting.
- Interesting Topics at 2003 ICML Conference
  - Selective sampling based on query learning (Abe)
- Overlapping Problems
  - Class Imbalance
  - Small Disjunct Problem (?)
  - Rare Cases
  - Data Duplication
  - Overlapping Classes

By 2003, the field started to mature.

6.2.2 Chabbouh 2019

This article [**CHABBOUH˙2019**] has a nice table classifying existing work in imbalanced classification; however, I think much of the information was old in 2019, particularly C4.5, an early decision tree base classifier that may not be used much anymore.

6.2.3 Mahmudah 2021

This article [**MAHMUDAH˙2021**] is really a review of current methods. They have some datasets, most public benchmark sets, and throw every combination of tools at them. The "methods" section is really an overview of current methods.

Has a section on techniques for feature extraction (feature engineering?) by dimensionality reduction, not particularly related to imbalanced data.

6.3 Examples of Good Writing, Models to Follow

- Elassad 2020 [**ELAMRANIABOUELASSAD2020102708**] is a good model.
- Paez 2021 [**PAEZ2020105666**] is not ML, but a solid paper. The conclusion suggests looking into imbalanced learning.
- Soleimani (LSU) 2019 [**SOLEIMANI201965**] gives a thorough analysis.

6.3.1 Elassad 2020

Good model to follow.

- In the title and first sentence of the abstract talks about an application, Collision Avoidance Systems, that the paper does not work with directly, which is like what I'm doing with mobile phones.
- Has several glaring mistakes, like crash avoidance systems on the vehicle having access to data from loop detectors, which are embedded under the road.
- Projects into the future, assuming that vehicles will detect the physiological state of the driver. I do this when I assume that police departments will have access to up-to-date and well-calibrated maps, to personal data from phone companies, and to be able to corollate several pieces of data (from multiple phones) in real time.
- Critique: Doesn't define terms well. What is an "ensemble fusion framework"? How are "ensemble" and "fusion" different? In layman's language, they sound the same. Uses "fusion" to mean both classifier ensembles and data fusion.
- Good overview at the end of the Introduction.
- The ML guts of this paper are trying different combinations of classifiers for an ensemble method. The guts of my paper will be different combinations of imbalanced data techniques.
- Only uses two imbalanced data techniques: Class weights and SMOTE.
- Algorithms
- Table of features
- Six points for future research

Chapter 7

Dataset

7.1 Overview

7.1.1 Misspellings

In the 'CITY' feature in the data, the name of the city of Shreveport is spelled nineteen different ways. It's not a problem, though, because it's spelled correctly about 47,000 times and incorrectly only 35 times.

7.2 Properties

7.2.1 Boolean Nature of our Data

Most of our data is boolean. Was alcohol involved? Did the car leave its lane? Was there a pedestrian? We have categorical variables, like type of vehicle which we represent as dummy (boolean) variables. We have some categories we could represent as numbers (like day of the week), and we could impose an order, (Monday comes before Tuesday), but the order isn't relevant in predicting injuries or fatalities, (Neither increases or decreases as the days "progress."), so we should represent them as categories, in dummy variables.

7.2.2 Top Twenty Features that Correlate with Fatality

Last column is the *balanced f1* score.

DR_COND_CD2	I	DRUG USE - IMPAIRED	0.33
SEC_CONTRIB_FAC_CD	L	CONDITION OF PEDESTRIAN	0.32
PRI_CONTRIB_FAC_CD	L	CONDITION OF PEDESTRIAN	0.25
PRI_CONTRIB_FAC_CD	M	PEDESTRIAN ACTIONS	0.20
VEH_TYPE_CD1	G	OFF-ROAD VEHICLE	0.18
M_HARM_EV_CD1	B	FIRE/EXPLOSION	0.17
DR_COND_CD2	F	APPARENTLY ASLEEP/BLACKOUT	0.17
CRASH_TYPE	C	[Unknown]	0.17
SEC_CONTRIB_FAC_CD	M	PEDESTRIAN ACTIONS	0.16
M_HARM_EV_CD1	O	PEDESTRIAN	0.15
VEH_COND_CD	E	ALL LIGHTS OUT	0.15
F_HARM_EV_CD1	O	PEDESTRIAN	0.15
M_HARM_EV_CD1	F	FELL/JUMPED FROM MOTOR VEHICLE	0.15
F_HARM_EV_CD1	F	FELL/JUMPED FROM MOTOR VEHICLE	0.14
PEDESTRIAN			0.13
VEH_TYPE_CD1	E	MOTORCYCLE	0.13
DR_COND_CD2	G	DRINKING ALCOHOL - IMPAIRED	0.13
CRASH_TYPE	A	[Unknown]	0.13
MOVEMENT_REASON_2	G	VEHICLE OUT OF CONTROL, PASSING	0.12

7.3 Thoughts on our Data Set: Trees

I suspect that a decision tree is the only realistic way to make a predict model for any aspect of crash data. If a pedestrian is involved, or it's a rural area, or alcohol is involved, the dynamics of the problem change. That there could be some linear (or nonlinear) function of all of the variables to fatality or injury is not reasonable to hope. If we think of it not as one big problem but as lots of little problems, like "What factors predict a fatality/injury in a crash involving a pedestrian in a rural area at night?" and, "What factors predict a fatality/injury in a crash where alcohol is involved at rush hour in an urban area?", we'll have much more likelihood of success.

7.4 Times

From the `Brads_Report_11_01_21`

7.4.1 New Features

Interesting features I didn't have before:

- `AMBULANCE` $\in \{0, 1\}$
- `CRASH_TIME`
- `TIME_POLICE_NOTE`

- TIME_POLICE_ARR
- TIME_AMB_CALLED
- TIME_AMB_ARR

In the 828,248 records, 167,662 (20.2%) have AMBULANCE==1.

7.4.2 Dirty Data

In many of the records, one of the times could be 0, which could indicate midnight, but more likely indicates missing data. Lots of the records mix up AM and PM. Some of them have the police or ambulance called before the crash time. In some of them, the ambulance isn't called until more than half an hour after the crash time, which could be real, but more likely a data entry error. Adding to the messiness is that some of the crashes roll over midnight.

There may be ways to fix some of those records, but for now I'll thrown them out. I threw out 47,640 records (28%), leaving 120,002 records.

7.4.3 Strange Data

The CRASH_TIME feature is in the format "1/1/01 HH:MM:SS," but the second are either "00" or "39." I don't know why. I'm going to ask Malek whether it appears that way in the original Access file.

7.4.4 Ambulance Call within/after 5 min after Crash

- In 64% of the cases, the ambulance was called within 5 min of the crash.
- In 15% of the cases, the ambulance was called more than 5 min after the crash and after the police arrived. Those 18,037 are the interesting cases.

7.4.5 Hospitalized

Of the 120,022 clean records where an ambulance was called, 43,902 (37%) had no one hospitalized, so while the ambulance crew may have applied minor first aid, it wasn't an emergency.

Chapter 8

Methods and Experimental Results To Date

8.1 scikit-learn

I ran just about every scikit-learn classifier, with results in my `12_July_2021_Report`.

Most Keras examples I see use tools from scikit-learn as well.

There's an add-on to scikit-learn called imbalanced-learn which has SMOTE, Tomek, and other tools.

8.2 Focal Loss and Tomek

Working with our crash database, with the cleaning and organizing in which I had it in February 2022, I tried different values for γ_1 and γ_2 with and without Tomek Links cleaning.

Tomek Links is a method for cleaning a noisy dataset for binary classification. A *Tomek Link* is a pair of samples, one from the positive and one from the negative class, that are each others' closest neighbors. The idea is that one of them is noise, or that having these two interferes in making a good classification, that you want the classes to cluster. In a balanced dataset you eliminate both of them from the training set. In an imbalanced dataset, you eliminate the element from the majority class.

From my weekly report 2/21/22:

- Unfortunately, p means two different things below.
  - p_i is the probability returned by the model that each sample belongs to the positive set.
  - p is a hyperparameter, ideally the proportion of the negative to positives samples, to use $\alpha = p/(p+1)$ in the Focal Loss function, to create the class weights that have the same effect as random oversampling. In our dataset, $p = 88.8$. (No, I'm not kidding.)
- All runs without Tomek used the same training and test sets
- All runs with Tomek used the same training and test sets

- The two test/train splits used the same random sampling seed, so they should be the same sets.

$$\begin{aligned}\text{Focal Loss} &= \sum_{i=1}^{n} \alpha(1-p_i)^{\gamma_1} y_i \log(p_i) + (1-\alpha)p_i^{\gamma_2}(1-y_i)\log(1-p_i) \\ &= \sum_{y_i=1} \alpha(1-p_i)^{\gamma_1}\log(p_i) + \sum_{y_i=0}(1-\alpha)p_i^{\gamma_2}\log(1-p_i)\end{aligned}$$

8.2.1 Different Values of p with $\gamma_1 = 0$, $\gamma_2 = 0$

Tomek?	p	γ_1	γ_2	TN/FN	FP/TP	Comments
No	1	0	0	573308	0	
				6466	0	
No	20	0	0	562182	11126	
				5850	616	
No	88.8	0	0	428929	144379	This is the natural p
				3105	3361	for our dataset.
No	100	0	0	411813	161495	
				2737	3729	
No	200	0	0	287151	286157	
				1464	5002	

8.2.2 Fixed $p = 88.8$, Different values of γ_1 and γ_2

Tomek?	p	γ_1	γ_2	TN/FN	FP/TP	Comments
No	88.8	0.0	0.0	428929	144379	This is the natural p
				3105	3361	for our dataset.
No	88.8	0.5	0.5	399870	173438	
				2685	3781	
No	88.8	1.0	1.0	420343	152965	
				3092	3374	
No	88.8	2.0	2.0	433805	139503	
				3213	3253	
No	88.8	5.0	5.0	445445	127863	
				3519	2947	
No	88.8	0.0	2.0	337148	236160	
				2092	4374	
No	88.8	0.5	0	460148	113160	
				3520	2946	
No	88.8	1.0	0.0	391820	181488	
				2596	3870	
No	88.8	2.0	0.0	527871	45437	
				4877	1589	

8.2.3 Tomek

Tomek took out 760 negative samples, bringing p down to 88.66.

Tomek?	p	γ_1	γ_2	TN/FN	FP/TP	Comments
No	88.8	0.0	0.0	428929	144379	
				3105	3361	
Yes	88.8	0.0	0.0	387313	185995	FN goes up 29%
				2504	3962	TP goes up 18%
Yes	88.66	0.0	0.0	387313	185995	
				2504	3962	

8.2.4 Discussion

- The different values of p, γ_1, and γ_2, and Tomek, give us different tradeoffs between false positives and false negatives, but no combination gives us fewer of both.
- It would be challenging to argue that one set of hyperparameters is "better" than another.
- I suspect that there just isn't enough of a pattern in this crash data to give us much confidence.
- I need to also work on other datasets that either might give clearer results, or will show me that all results are this fuzzy and I need to learn how to deal with it.

8.3 Feature Engineering

8.3.1 Time of Day

Time of day is a continuous variable, but the correlation between time of day and [anything] is nonlinear. We could do some kind of data transformation, perhaps taking the ratio of the number of accidents to the typical traffic density at that time of day, but the typical car trip at 3 am on a Wednesday may be different in character than a car trip at 7 am on a Saturday, even if the traffic volumes are similar. Perhaps we should have boolean variables:

- Morning rush hour
- Mid-day
- Afternoon rush hour
- Evening
- Late night

and another variable, `Weekend`.

8.3.2 Number of Fatalities/Injuries

The number of fatalities or injuries is a function of how many people were in each vehicle, which (a) we don't know and (b) probably isn't correlated to any other data we have. Fatality and injury should be boolean variables, that there was a fatality or there was an injury, rather than a count of the number of fatalities or injuries.

8.3.3 Day of Week

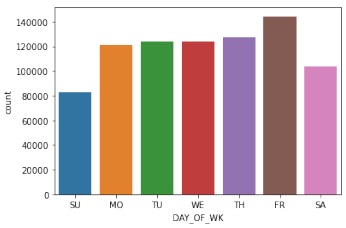


Figure 8.1: Number of Crashes, by Day of Week

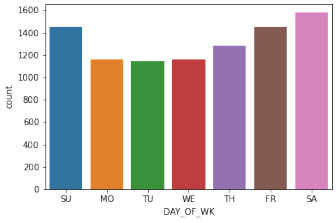


Figure 8.2: Number of Severe Injury Crashes, by Day of Week

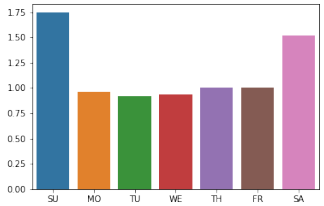


Figure 8.3: Percentage of Crashes with Severe Injury, by Day of Week

My understanding is that, for feature engineering, we don't care that there are more crashes on Friday than other weekdays, since the proportion of crashes that require an ambulance are the same. Saturday and Sunday, though, are different.

I made a feature, `Weekday_SA_SU`:

0	MO, TU, WE, TR, FR
1	SA
2	SU

8.3.4 Time of Day

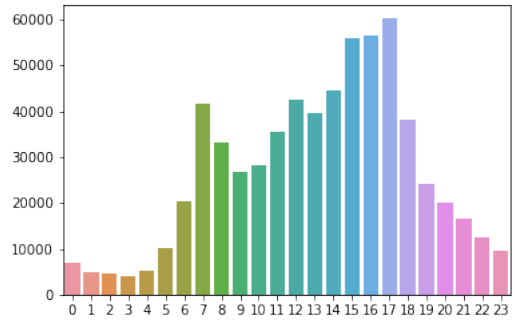


Figure 8.4: Number of Weekday Crashes, by Hour

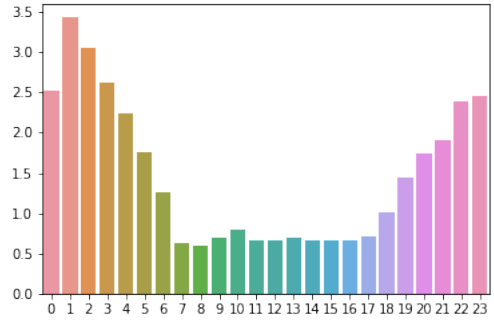


Figure 8.5: Percentage of Weekday Crashes with Severe Injury, by Hour

I note with interest that, at 7am, the number of crashes spikes, but the percentage of severe injury crashes does not change significantly. I created a `Rush_Hour` feature, but I don't know if it will be of any use.

The spike of percentage of crashes at 1am is just noise, because of the small number of crashes at that time.

The types of roads on which crashes occurs varies widely by time of day. I don't know what to do with that.

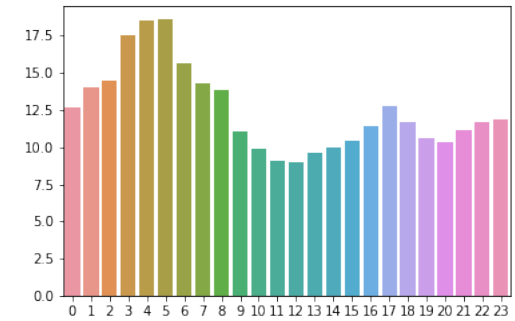


Figure 8.6: Percentage of Crashes on Interstates, by Hour

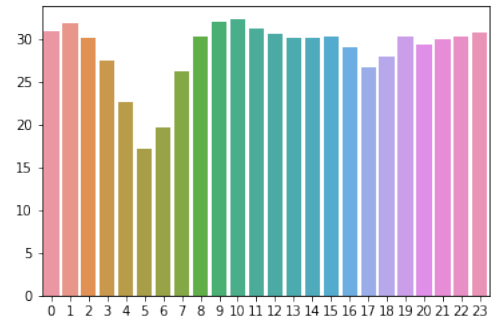


Figure 8.7: Percentage of Crashes on City Streets, by Hour

I made a feature, `Time_of_Day`, grouping together times with similar percentages of crashes having severe injuries:

0	Midnight - 3:00 am
1	3:00 am - 5:00 am
2	5:00 am - 7:00 am
3	7:00 am - 5:00 pm
4	5:00 pm - Midnight

8.3.5 Location

Location seems like it would be very important. One proxy we have is the parish and the road name. I've made a new feature, concatenating the parish and the road name. Each unique value in that feature will become a category, yielding a new feature in the dummy (one-hot encoding) dataframe that we will use for training.

There are 6.150 unique values, but most of them have few records. How many records do you need to make a useful correlation, and how many categories will overload the training?

Having a minimum of 1000 records per category gives me 142 categories plus 492,367 records in "Other"; a minimum of 100 records gives me 1103 categories plus 221,644 records in "Other." A minimum of 10 records gives me 2,534 categories and 171,802 in "Other." Note that 161,454 are in "Other" because of missing data.

8.3.6 Parish/Road Names

- We have 161,454 records with "0" for the `PRI_ROAD_NAME`. There's nothing we can do to recover those.
- We have 26,289 different values for `PRI_ROAD_NAME`.
- We have even more if we combine those with the, sometimes multiple, `PRI_ROAD_TYPE`, like St, Ave, and Blvd.
- In a few instances, roads with the same `PRI_ROAD_NAME` and different `PRI_ROAD_TYPE` are different roads, but usually within the same parish they're the same. A noteable exception is North St and North Blvd in Baton Rouge.
- For long roads, like interstates and some state highways, crash outcomes may differ based on which section of road you're on.
- To Do:
  - Combine `PARISH_CD` and `PRI_ROAD_NAME` into a new feature, `PARISH_CD_and_PRI_ROAD_NAME`.
  - Ignore the `PRI_ROAD_TYPE`
  - Keep the instances of `PARISH_CD_and_PRI_ROAD_NAME` that have more than 1000 crashes.
  - Change all of the others to "Other".
- Results:
  - This leaves us with 142 different names with 335,880 crashes, plus "Other" with 492,367 crashes.

- `PRI_ROAD_NAME` = "AIRLINE" appears (with at least 1000 crashes) in 11 parishes, with 51,399 crashes.

Chapter 9

Research Plan

9.1 Goals

9.2 Progress

9.3 Steps