

Highlights

Modeling the Need for an Ambulance based on Automated Crash Reports from iPhones

First Author, Second Author, Third Author, Fourth Author

- Supports transferability and benchmarking of different approaches on a public large-scale dataset. We have attached the code we used to perform the analysis on the Crash Report Sampling System.
- Novel Application motivated by Emerging Technology: Machine Learning Classification Models for Dispatching Ambulances based on Automated Crash Reports
- New Use of Dataset: Used Crash Report Sampling System (CRSS), which has imputed missing values for some features, but not all of the ones we wanted to use. For the first time we have seen, we used the software the CRSS authors use for multiple imputation (IVEware) to impute missing values in more features.
- Perennial Machine Learning Challenge: Imbalanced Datasets.

Modeling the Need for an Ambulance based on Automated Crash Reports from iPhones

First Author^{a,b}, Second Author^a, Third Author^{a,c} and Fourth Author^c

^aSchool, University,

^bOther School,

^cOther Department, University,

ARTICLE INFO

Keywords:

Automated crash notification
Ambulance dispatch
Emergency medical services
Machine learning
Imbalanced Data
Imputation

ABSTRACT

New Google Pixel phones can automatically notify police if the phone detects the deceleration profile of a crash. From the data available from such an automatic notification, can we build a machine-learning model that will recommend whether police should immediately, perhaps automatically, dispatch an ambulance? If the injuries are serious, time to medical care is critical, but few crashes result in serious injuries, and ambulances are in limited supply and expensive.

The costs of the false positives and false negatives are very different. The cost of sending an ambulance when one is not needed is measured in dollars, but the cost of not sending an ambulance when one is needed is measured in lives. Each society chooses a marginal ethical tradeoff rate $etr = \Delta FP / \Delta TP$ when it set budgets. For our work we arbitrarily chose a value for etr and incorporated it into the model in the class weight and decision threshold.

We will show that the quality of the model depends mostly on what information is available to inform the decision of whether to immediately dispatch an ambulance. We used the data of the Crash Report Sampling System (CRSS). This data is freely available online. We have applied new methods (for this dataset in the literature) to handle missing data, and we have investigated several methods for handling the data imbalance. To promote discussion and future research, we have included all of the code we used in our analysis.

1. Introduction
2. Introduction
3. Literature Review
4. Dataset
5. Methods


We have written this section as a guide for other to replicate and adapt our work, so some details may seem pedantic.

5.1. Preparing the Data

The CRSS data is available [online at this link](#). The three main files for each year are Accident, Vehicle, and Person, and one uses the CASENUM and VEH_NO fields to merge them into one dataset.

5.1.1. Order of Operations

To prepare the data we needed to do two things, to bin (discretize) some fields and to impute missing data. We did not know which to do first, so we tested both ways using IVEware (Raghunathan, Solenberger, Berglund and van Hoewyk) for the imputation. The imputation is a stochastic process, and the difference between discretizing first and imputing first was as small as the difference between running twice with different random seeds. Since IVEware can only handle up to about forty categories in each categorical field, we had had to discretize some fields first either way, so we decided on discretizing first.

 FirstAuthor@gmail.com (F. Author)
ORCID(s):

5.1.2. Binning

To bin a field's many categories into fewer categories, sometimes the meaning of the categories was a sufficient guide. In the HOSPITAL field, which we used as our target variable, we were only interested in two values, whether or not the person went to the hospital. The CRSS field has six values indicating how the person went to the hospital (ground ambulance, air ambulance, ...), and we merged those into one. For fields where the binning was not so obvious, we looked at how each value in the field correlates to hospitalization. We wanted to put AGE into bands, and looked to divide where the hospitalization rate changed. Interestingly, ages 16, 17, and 18 have lower hospitalization rates than ages below or above, so we put them into their own band. Around age 52 the hospitalization rate started to go up, so we split there. We binned other fields in a similar way.

The merging, dropping, and discretizing are all in the CRSS_04_Discretize code.

5.1.3. Imputing Missing Values

About 47% of the samples had unknown values in the thirty-eight fields we use for our analysis. The CRSS authors imputed unknown values in ten of those fields, another seventeen had no unknown values, but eleven fields we want to use had missing values that were not imputed by CRSS. The CRSS authors have a very helpful report on their imputation methods. (Herbert, 2019) The reasons why some fields get imputed include historical consistency going back to 1982.

(See CRSS_04_5_Count_Missing_Values)

When the CRSS authors imputed unknown values for a field, they published two fields, one with the imputed values and one with the values signifying "Unknown." We discarded the imputed fields and compared three methods for imputing missing values. Impute to Mode assigns to all missing values in a feature the most common value in that feature. IVEware: Imputation and Variance Estimation Software employs multivariate sequential regression, and is the method the CRSS authors used. Round Robin Random Forest, like in MissForest, was consistently the most accurate. We tested the methods by dropping all samples with missing values, randomly deleting (but keeping a copy of) fifteen percent of the known values, imputing, and comparing to the ground truth.

(See CRSS_05_Impute_Random_Forest for details.)

We did not address the question of incorrect data.

5.2. Selecting Features

We selected three groups of features to see whether more information would improve the model.

The first group of features held information that the police would already know before receiving a crash notification, like time of day, day of week, and urban/rural. A crash on a Saturday night in a rural area is far more likely to need an ambulance than one in a city at rush hour, so if no information specific to the crash is available, how well can we predict whether an ambulance is needed? We thought of this set of features as "easy" or "baseline."

The second group of features also included specific location and the age and sex of the primary user of the phone. Is the vehicle in an intersection or in a parking lot? Did the car end up off the roadway? What is the speed limit on that road? Getting that information from the latitude and longitude in the automated report would require instantaneous correlation with detailed maps. Whether such information significantly improves the model will inform whether policymakers should invest the time and effort to have that information available. We thought of this information as "medium" in cost.

The "hard" or "expensive" features would require regularly updated maps (work zones, lighting conditions), correlating records to guess which car the cell phone user is driving, and correlating multiple cell phone reports to count how many people are involved.

We dropped all crashes with a pedestrian, because unlike a tree or other vehicle, hitting a pedestrian may not cause the sudden deceleration that a cell phone could distinguish from sudden braking, so the cell phone likely would not register it as a crash.

(See CRSS_06_Build_Model for details.)

5.3. Handling Imbalanced Data

In our dataset only about fifteen percent of the people needed an ambulance. A recommendation system never send an ambulance, the model would have 85% accuracy, but be useless. Most algorithms for training models are designed for balanced data, with half of the samples in each of the negative and positive classes. With an imbalanced data set we

can address the imbalance in four levels: Resampling the dataset, modifying the loss function, choosing metrics other than accuracy, and using learning methods that account for the imbalance.

5.3.1. Resampling the Dataset

We can balance the dataset by undersampling the majority class (negative, “No ambulance”) or oversampling the minority class (positive, “Send Ambulance”). To balance by undersampling would mean throwing out eighty percent of the majority class, losing valuable information. A very popular method for oversampling is SMOTE (Synthetic Minority Oversampling TEchnique), which creates new minority samples between existing minority samples, but the “between” requires continuous data, and all of our data is discrete or categorical. What is between a Buick and a Volvo?

Tomek Links is one of the few resampling methods that works for categorical data. It is a selective undersampling method that removes majority samples that seem out of place. A Tomek Link is a majority/minority pair that are each others’ nearest neighbors, which was the case with about four percent of the majority samples. We used the Tomek algorithm to remove the majority sample of each Tomek link, undersampling the majority class, and then running it again to remove more that had not been Tomek links in the first round. We were disappointed to not see a significant improvement in the model metrics from the undersampling. (**Put in Label Reference**).

5.3.2. Modifying the Loss Function

A popular and well established way to modify the loss function for imbalanced data is with class weights, which can have the same effect as naïve oversampling. We are going to use the class weights to impose the $\Delta FP/\Delta TP < 2.0$ goal, as described in (**Put in Label Reference**).

A newer method is with focal loss, which increases the penalty for badly misclassified samples. (Lin, Goyal, Girshick, He and Dollár, 2017) We did not see significant improvement using focal loss. (**Put in Label Reference**).

5.3.3. Metrics

Precision tells us, of the ambulances we sent, how many were needed. **Recall** tells us, of the ambulances that were needed, how many we sent. Recall only looks at elements of the minority class, so is independent of the class imbalance. Precision is affected by class imbalance, but is still relevant to our decisions in its imbalanced form.

The **F1 score** is the harmonic mean of precision and recall. Why the harmonic mean instead of the arithmetic or geometric? For two positive numbers a and b with $0 < a < b$,

$$a < Harm(a, b) < Geo(a, b) < Arith(a, b) < b$$

so the F1 score emphasizes what the model does poorly. We will use F1 as our primary indicator, while looking at precision and recall.

The area under the curve (**AUC**) of the receiver operating characteristic (ROC) is a measure of how well a model separates the samples of the positive and negative classes. We will use it to show that the additional features in the “hard/expensive” and “medium” datasets are important for discriminating between the two classes.

The $\Delta FP/\Delta TP$ curve is related to the ROC; $\Delta FP/\Delta TP$ is the reciprocal of the product of the slope of the ROC curve and a factor that corrects for class imbalance.

$$\frac{\Delta FP}{\Delta TP} = \frac{N}{P} \cdot \frac{\frac{\Delta FP}{N}}{\frac{\Delta TP}{P}} = \frac{N}{P} \cdot \frac{\Delta FPR}{\Delta TPR} = \frac{1}{\frac{P}{N} \cdot \frac{\Delta TPR}{\Delta FPR}} = \frac{1}{\frac{P}{N} \cdot mROC}$$

We will use this curve to find the value of the discrimination threshold where $\Delta FP/\Delta TP = 2.0$

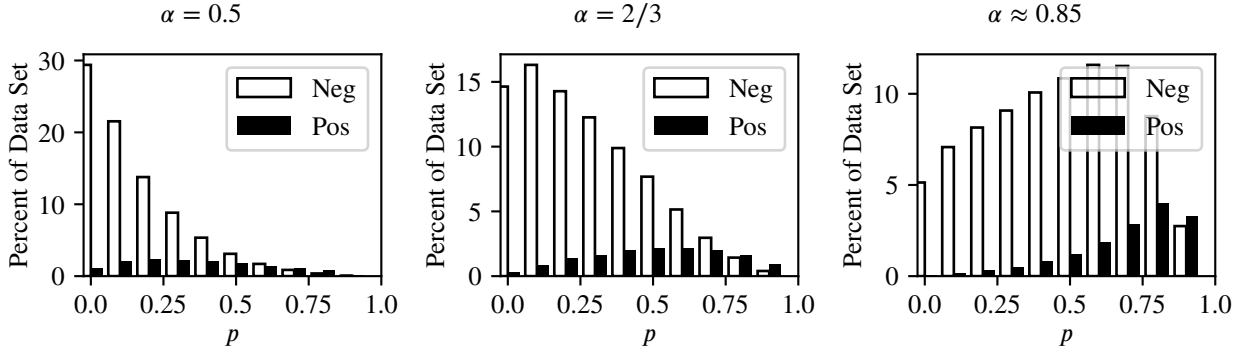
6. Results

6.1. Binary Focal Crossentropy

The Binary Focal Crossentropy loss function for Keras’s neural network algorithm takes both a class weight hyperparameter α and a dampening factor hyperparameter γ that gives more weight to samples badly misclassified. We tried combinations, including the values of γ tested in the paper Lin et al. (2017) The balanced class weight will vary with undersampling of the majority class.

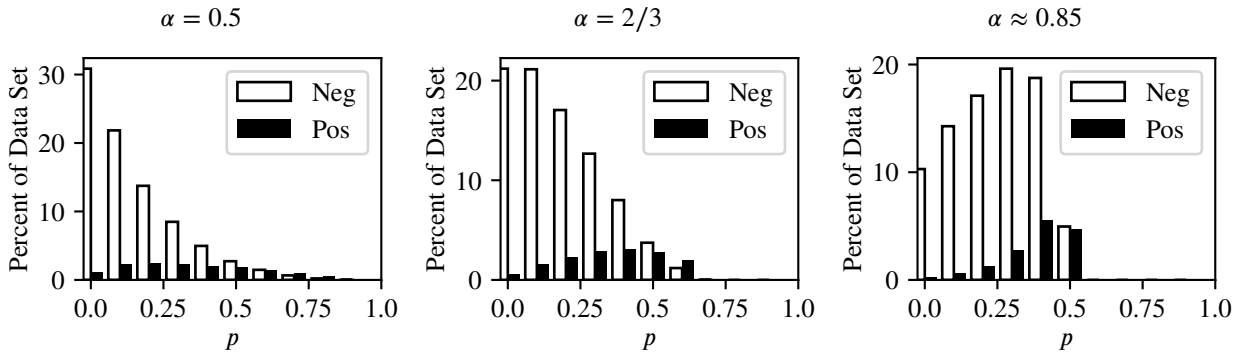
| α | Meaning | γ | Notes |
|----------------|------------------------|----------|-----------------------------|
| 0.5 | No class weight | 0.0 | Same as binary crossentropy |
| 2/3 | $r_{target} = 2$ | 0.5 | Very light damping |
| ≈ 0.85 | Balanced class weights | 1.0 | Light damping |
| | | 2.0 | Recommended by Lin |
| | | 5.0 | Heavy damping |

Varying class weights with no Tomek undersampling and $\gamma = 0.0$, raw probabilities.



With $\alpha = 0.5$, the model classifies the negative class well, but the positive class almost randomly. With $\alpha = 0.85$, balanced class weights, the model classifies the positive class well, but the positive class so poorly that the model does not separate the two classes well.

Varying class weights with no Tomek undersampling and $\gamma = 0.0$, probabilities linearly transformed to center where $\Delta FP / \Delta TP = 2.0$



| | | Prediction | |
|--------|---|------------|--------|
| | | N | P |
| Actual | N | 117,929 | 32,842 |
| | P | 5,928 | 20,693 |

0.4 Precision
0.6 Recall
0.5 F1

7. Conclusions

8. Discussion

9. Future Work

Funding Statement

Conflict of Interest

The authors have no relevant financial or non-financial interests to disclose.

Acknowledgements

[STUDENT] contributed to this work in the [FUNDED PROGRAM]

Data Availability

The CRSS data is publicly available at

<https://www.nhtsa.gov/crash-data-systems/crash-report-sampling-system>

10.

CRedit authorship contribution statement

First Author: Conceptualization, Investigation, Writing - original draft, Visualization. **Second Author:** Supervision, Methodology, Writing - review and editing. **Third Author:** Investigation, Methodology. **Fourth Author:** Data curation, Writing - review and editing.

References

- Herbert, G., 2019. Crash Report Sampling System: Imputation. Technical Report DOT HS 812 795. National Highway Traffic Safety Administration.
- Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, pp. 2980–2988.
- Raghunathan, T., Solenberger, P., Berglund, P., van Hoewyk, J., . Ivedere: Imputation and variation estimation software. URL: <https://www.src.isr.umich.edu/software/iveware/>.