

Highlights

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

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 data from the Crash Report Sampling System (CRSS).
- 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, then compared the results with other methods of imputation.
- Explicit Incorporation of Imbalanced Costs
- Explicit Incorporation of Political Dimensions
- Perennial Machine Learning Challenge: Imbalanced Datasets

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

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 Cost
Imbalanced Data
Imputation

ABSTRACT

New Google Pixel phones can automatically notify an emergency dispatcher if the phone detects the deceleration profile of a vehicular crash. Most crash notifications come from an eyewitness who can say whether an ambulance is needed, but the automated notification from the cell phone cannot provide that information directly. Should the dispatcher immediately send an ambulance before receiving an eyewitness report? There are three options: Always, Wait, and Sometimes. The “Always” option refers to sending an ambulance to every automatically reported crash, even though most of them will not be needed. In the “Wait” option, the dispatcher sends police, but always waits for a call from an eyewitness (perhaps the police) before sending an ambulance. In the “Sometimes” option, the dispatcher relies on a machine learning recommendation system to decide whether to immediately dispatch an ambulance, reserving the option to send one later based on an eyewitness report.

This paper explores one option for building a machine learning (ML) model for making a recommendation in the “Sometimes” option. Our goal is to build a model that returns, for each feature vector (crash report, sample), a value $p \in [0, 1]$ that increases with the probability that the person needs an ambulance. Then we choose a threshold θ such that we immediately send ambulances to those automated crash reports with $p > \theta$, and wait for eyewitness confirmation for those reports with $p < \theta$. In an actual implementation, the choice of θ is political, not technical, so we consider and interpret several options.

Once a threshold has been chosen, the costs of the false positives (FP) and false negatives (FN) in dispatching ambulances are very different. The cost of sending an ambulance when one is not needed (FP) is measured in dollars, but the cost of not promptly sending an ambulance when one is needed (FN) is measured in lives. Choosing such a tradeoff threshold is ethically problematic, but governments implicitly choose such a tradeoff when they set budgets for emergency services.

We consider and interpret several options for the decision threshold θ based on the political consideration, “How much will it cost?” How many automated ambulance dispatches are we willing to fund (FP + TP) for each one of them that’s actually needed (TP)? We will explore two versions of that question, the total and the marginal.

We show that the quality of the model depends highly on the input data available, and we considered three levels of data availability. The “Easy” level includes time of day and weather, data the emergency dispatcher has before the notification. The “Medium” level adds the age and sex of the cell phone user and information about the location. The “Hard” level adds information about the vehicle likely to be driven by the cell phone user and detailed and temporal information about the location, like lighting conditions and whether it is currently a work zone.


We used the data of the Crash Report Sampling System (CRSS) to validate our approach. 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. To Henry, 22 July 2023

Henry,

Greetings from Frankfurt! And later in the writing process, Barcelona!

I think I’ve found something that works for building and interpreting a model in a way that’s actually useful in the political settings where these decisions are made.

 FirstAuthor@gmail.com (F. Author)

ORCID(s):

The analysis all builds on moving the decision threshold from the default $p = 0.5$ to a value of p that gives you the tradeoff the political process chooses. I haven't seen this done much, and I don't know whether there are good reasons to not do it.

I haven't found such an approach in the crash analysis literature, particularly using the slope of the ROC curve (or something equivalent to it). There are three possibilities.

1. I haven't looked hard enough.
2. I've found something new, at least in the application.
3. The thing I'm doing doesn't work.

1.1. Big Question

I think that whether my analysis works hinges on this question.

Once you've built a binary classification model on the training set and evaluate it on the test set, the model returns for each sample in the test set a value p that I've been told gives the probability that the sample is in the positive class. Then you analyze the model by picking a decision threshold θ and seeing how many elements of the positive and negative classes have values of p on the correct side of θ .

The value of p isn't exactly the probability, but my analysis hinges on this conjecture.

1. If a model gives two samples the same value of p , then the model says that those two samples have the same probability of being in the positive class.
2. Probability is an increasing function of p : If a model gives samples a and b values p_a and p_b with $p_a < p_b$, then, according to the model, sample b has a higher probability of being in the positive class than sample a .

I would appreciate your thoughts and direction.

Thanks,

Brad

1.2. Scenario

The scenario is that the emergency dispatchers receive an automated crash notification from a cell phone and have yet not received a call from an eyewitness. The dispatchers do not know with strong certainty whether an ambulance is needed, but they have some indicators. The dispatchers have three options.

Always Always send an ambulance immediately to all such notifications, knowing that only about 15% will be needed.

Wait Dispatch police, but wait for a report from an eyewitness (perhaps the police) before sending an ambulance.

Sometimes Use a machine learning model with some decision threshold to decide which crashes to send an ambulance to immediately, reserving the option to send an ambulance later based on an eyewitness report.

1.3. Decision Threshold

Here's something new from my last email.

How to determine the decision threshold is a political decision, not a technical one. We will consider three ways politicians might answer that question and how to implement each in our models and decision thresholds.

1. Our local fleet of ambulances now goes to n crashes per year. In the short term, without buying more ambulances and hiring more teams, we can increase the number of ambulance runs to crashes by some percentage, or in the longer term we are willing to increase the number of ambulances going to crashes by a larger, but still fixed, percentage.

We will use 5% as our example of how to implement this policy. The increase does not include the true positives (TP), because those ambulances would go anyway; the increase is the allowable number of false positives (FP). Set the decision threshold where the number of false positives is 5% of the positive class ($P = FN + TP$).

$$\frac{FP}{P} = \frac{FP}{FN + TP} = 0.05$$

Lots of ratios of TN, FP, FN, and TP have names, but I haven't found a name for FP/P or where other people have used it.

2. We are willing to send ambulances based on automated crash reports, but only up to the point where a certain proportion of the ambulances we immediately dispatch (FP+TP) are actually needed (TP), which is equivalent to saying that we are willing to immediately send a certain number of unneeded ambulances for each one we automatically send that is needed.

This is what I was trying to get at with FP and TP, but I realized it's equivalent to Precision, which the readers will understand.

Choose the decision threshold where the precision is the specified level. We will use 1/3 for our example, being willing to send two FP for each TP.

3. By looking at the slope of the ROC curve we can (roughly) estimate the probability that a particular crash needs an ambulance. Some of them almost definitely need an ambulance, and we should dispatch those immediately, but we will choose a minimum probability to which we will immediately dispatch an ambulance.

I'm going to use 50% as an example of the minimum probability. The model assigns to each sample in the test set at value $p \in [0, 1]$. I had read that p was the probability that the sample was in the positive class, but I don't think that's exactly true. What is true is that p generally increases with probability.

In each sufficiently large* range of p (like $p \in [0.60, 0.61]$) there are some number of elements of the negative and positive class. For a given range of p , call the number of elements of the negative class "Neg" and the number of elements of the positive class "Pos." For the samples in that range of p , the probability that they are in the positive class is

$$\frac{\text{Pos}}{\text{Pos} + \text{Neg}}$$

This expression is proportional to the slope of the ROC curve at that value of p .

I want to call this "marginal precision," but I haven't seen that term used that way in the ML literature. I think "marginal precision" means something else in statistics. See that

$$\text{Pos} = \frac{\Delta \text{TP}}{\Delta p}, \quad \text{Neg} = \frac{\Delta \text{FP}}{\Delta p}, \quad \text{and} \quad \text{Pos} + \text{Neg} = \frac{\Delta(\text{TP} + \text{FP})}{\Delta p}, \quad \text{so} \quad \frac{\text{Pos}}{\text{Pos} + \text{Neg}} = \frac{\Delta \text{TP}}{\Delta(\text{TP} + \text{FP})}$$

*We have two challenges with choosing Δp , the size of our range of p , which we would like to be really small. One is that some of our ML algorithms return (almost all) values of p rounded to two decimal places, so we can't get more precision than that. One of my algorithms (Balanced Bagging) gives p for each sample to only one decimal place; thus, "sufficiently large" depends on the algorithm.

The other problem is that if you zoom in too close, the number of samples in each Δp isn't large enough to compensate for the randomness, and you see that your "curve" is actually jagged. Because I wanted more samples in my test set, I went from using a 70/30 train/test split to using 5-fold validation, where all of the samples are in a test set.

1.4. Understanding the Data

1.4.1. Total Effects

The confusion matrix gives the totals based on the model and choice of decision threshold.

		Predicted	
		PP	PN
Actual	P	TP	FN
	N	FP	TN

Ambulance Dispatch

P	Number of crash persons who need an ambulance, also the number to whom we will <i>immediately or eventually</i> send an ambulance
N	Number of crash person who do not need an ambulance
PP	Number of crash persons to whom we immediately dispatch an ambulance
PN	Number of crash persons to whom we do not <i>immediately</i> dispatch an ambulance
TP	Number of crash persons to whom we send a needed ambulance
FN	Number of crash persons who need an ambulance, but to whom we do not <i>immediately</i> send one, but we will send one later
FP	Number of crash persons to whom we dispatch an ambulance that is not needed
TN	Number of crash persons who do not need an ambulance and to whom we do not immediately (or ever) send one

$\frac{TP}{FP + TP} = \frac{TP}{PP}$	Precision, the proportion of immediately dispatched ambulances that are needed
$\frac{TP}{FN + TP} = \frac{TP}{P}$	Accuracy, the proportion of crash persons needing an ambulance to whom we immediately dispatch one
$\frac{FP}{FN + TP} = \frac{FP}{P}$	Proportion of increase in number of ambulances sent because of immediate dispatch. The P number of ambulances would have been dispatched anyway, but now the FP number of ambulances are also being sent.

1.4.2. Marginal Effects of Moving the Decision Threshold

What if we move the decision threshold from $\theta = 0.85$ to $\theta = 0.86$? What would the marginal effect be?

p	Neg	Pos	mPrec	TN	FP	FN	TP	Prec	Rec	FP/P	\hat{p}
0.80	1,392	1,585	0.53	589,317	11,498	90,351	22,400	0.66	0.20	0.10	0.05
0.81	1,358	1,673	0.55	590,675	10,140	92,024	20,727	0.67	0.18	0.09	0.04
0.82	1,244	1,590	0.56	591,919	8,896	93,614	19,137	0.68	0.17	0.08	0.04
0.83	1,112	1,560	0.58	593,031	7,784	95,174	17,577	0.69	0.16	0.07	0.04
0.84	972	1,594	0.62	594,003	6,812	96,768	15,983	0.70	0.14	0.06	0.03
0.85	952	1,479	0.61	594,955	5,860	98,247	14,504	0.71	0.13	0.05	0.03
0.86	886	1,489	0.63	595,841	4,974	99,736	13,015	0.72	0.12	0.04	0.03
0.87	712	1,481	0.68	596,553	4,262	101,217	11,534	0.73	0.10	0.04	0.02
0.88	731	1,429	0.66	597,284	3,531	102,646	10,105	0.74	0.09	0.03	0.02
0.89	622	1,389	0.69	597,906	2,909	104,035	8,716	0.75	0.08	0.03	0.02
0.90	565	1,357	0.71	598,471	2,344	105,392	7,359	0.76	0.07	0.02	0.01

p		Value returned by the model for each sample (crash person), a proxy for the probability that the sample is in the positive class (crash person needs an ambulance). It is not exactly the probability, but it increases with the probability.
Δp	0.01	Width of each band of p. For the above model, we can't go smaller because almost all of the returned p values are rounded to two decimal places. In some models the values of p are more continuous, so we could go with smaller Δp , but if we go too small, the randomness inherent in the algorithm would make the results not smooth.
Neg	1,479	The number of additional unneeded ambulances we would send if we changed p from 0.86 to 0.85 The number of actual negative samples in this band of p

Ambulance Dispatch

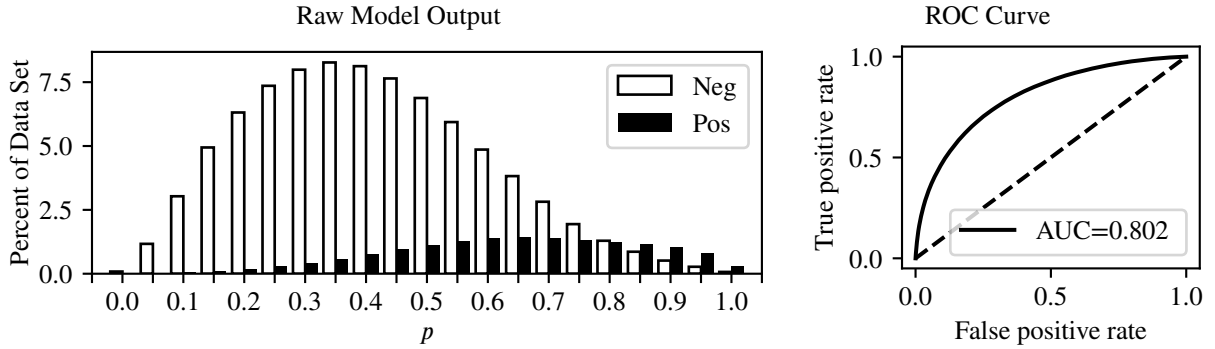
$\text{mPrec} = \frac{\text{Pos}}{\text{Neg} + \text{Pos}}$	Pos	952	The number of additional needed ambulances we would send if we changed p from 0.86 to 0.85 The number of actual positive samples in this band of p
		0.61	61% of the <i>additional</i> ambulances we are sending (because we moved p from 0.86 to 0.85) that are needed “Marginal Precision” All of the ambulances we are immediately dispatching have <i>at least</i> a 61% chance of being needed
	TN	594,955	If we choose $p = 0.85$, then we will correctly decide that these 594,955 people do not need an ambulance, immediately or at all. Running total of Neg from $p = 0$
	FP	5,860	If we choose $p = 0.85$, then we will immediately dispatch 5,860 unneeded ambulances Reverse running total of Neg from $p = 1.00$
	FN	98,247	If we choose $p = 0.85$, then we will not <i>immediately</i> send ambulances to these 98,247 people, but will dispatch them when we hear from an eyewitness that an ambulance is needed Running total of Pos from $p = 0$
	TP	14,504	If we choose $p = 0.85$, then we will immediately dispatch these 14,504 ambulances Running reverse total of Pos from $p = 1.00$
	FP/P	0.05	By choosing $p = 0.85$, we will send 5% more ambulances than if we did not dispatch any ambulances. This number shows the increased cost of ambulance service.
$\hat{p} = \frac{\text{PP}}{\text{PN} + \text{PP}}$		0.03	Proportion of crash persons to whom we are immediately dispatching an ambulance. I think Henry said that, usually, we want \hat{p} to be the same as the proportion of crash persons who need an ambulance, $P/(N + P)$, which I think is equivalent to saying that $\text{FP} = \text{FN}$. I’m pretty sure that’s not what we want in this application.

1.5. Evaluating Models

The politicians’ choice of options above determine where we set the decision threshold for each model. Choosing between models is then easy: Choose the model that yields the largest number of true positives (maximizes TP), which is the one that maximizes recall, because $\text{Recall} = \text{TP}/P$, and the number of positive samples is the same in all of the models we’re building.

1.6. Example: Finding Decision Thresholds in One Model

Here’s a histogram of the p values of the elements of the negative and positive classes for one model, the Balanced Random Forest Classifier with no class weights, on the full-feature (“Hard”) dataset that includes relevant information that may or may not be available in real time. This model doesn’t do a great job of separating the positive and negative classes, but it’s better than random.



Here's a table of the same data with $\Delta p = 0.05$. With the output from this model we can go as small as $\Delta p = 0.01$, but that would take several pages to show. I will zoom in on parts of the table below.

p	Neg	Pos	mPrec	TN	FP	FN	TP	Prec	Rec	FP/P	\hat{p}
0.00	617	2	0.00	617	600,198	2	112,749	0.16	1.00	5.32	1.00
0.05	8,421	69	0.01	9,038	591,777	71	112,680	0.16	1.00	5.25	0.99
0.10	21,762	308	0.01	30,800	570,015	379	112,372	0.16	1.00	5.06	0.96
0.15	34,768	625	0.02	65,568	535,247	1,004	111,747	0.17	0.99	4.75	0.91
0.20	45,253	1,271	0.03	110,821	489,994	2,275	110,476	0.18	0.98	4.35	0.84
0.25	52,539	2,041	0.04	163,360	437,455	4,316	108,435	0.20	0.96	3.88	0.77
0.30	57,103	3,074	0.05	220,463	380,352	7,390	105,361	0.22	0.93	3.37	0.68
0.35	59,125	4,177	0.07	279,588	321,227	11,567	101,184	0.24	0.90	2.85	0.59
0.40	57,636	5,410	0.09	337,224	263,591	16,977	95,774	0.27	0.85	2.34	0.50
0.45	54,781	6,860	0.11	392,005	208,810	23,837	88,914	0.30	0.79	1.85	0.42
0.50	48,984	7,956	0.14	440,989	159,826	31,793	80,958	0.34	0.72	1.42	0.34
0.55	42,326	8,998	0.18	483,315	117,500	40,791	71,960	0.38	0.64	1.04	0.27
0.60	34,742	10,006	0.22	518,057	82,758	50,797	61,954	0.43	0.55	0.73	0.20
0.65	27,178	10,132	0.27	545,235	55,580	60,929	51,822	0.48	0.46	0.49	0.15
0.70	20,014	9,991	0.33	565,249	35,566	70,920	41,831	0.54	0.37	0.32	0.11
0.75	13,935	9,409	0.40	579,184	21,631	80,329	32,422	0.60	0.29	0.19	0.08
0.80	9,356	8,739	0.48	588,540	12,275	89,068	23,683	0.66	0.21	0.11	0.05
0.85	6,131	8,400	0.58	594,671	6,144	97,468	15,283	0.71	0.14	0.05	0.03
0.90	3,687	7,562	0.67	598,358	2,457	105,030	7,721	0.76	0.07	0.02	0.01
0.95	1,968	5,691	0.74	600,326	489	110,721	2,030	0.81	0.02	0.00	0.00
1.00	489	2,030	0.81	600,815	0	112,751	0	nan	0.00	0.00	0.00

1.7. Answers to Political Questions for This Model

1. Send up to 5% more ambulances.

p	Neg	Pos	mPrec	TN	FP	FN	TP	Prec	Rec	FP/P	\hat{p}
0.80	1,392	1,585	0.53	589,317	11,498	90,351	22,400	0.66	0.20	0.10	0.05
0.81	1,358	1,673	0.55	590,675	10,140	92,024	20,727	0.67	0.18	0.09	0.04
0.82	1,244	1,590	0.56	591,919	8,896	93,614	19,137	0.68	0.17	0.08	0.04
0.83	1,112	1,560	0.58	593,031	7,784	95,174	17,577	0.69	0.16	0.07	0.04
0.84	972	1,594	0.62	594,003	6,812	96,768	15,983	0.70	0.14	0.06	0.03
0.85	952	1,479	0.61	594,955	5,860	98,247	14,504	0.71	0.13	0.05	0.03
0.86	886	1,489	0.63	595,841	4,974	99,736	13,015	0.72	0.12	0.04	0.03
0.87	712	1,481	0.68	596,553	4,262	101,217	11,534	0.73	0.10	0.04	0.02
0.88	731	1,429	0.66	597,284	3,531	102,646	10,105	0.74	0.09	0.03	0.02
0.89	622	1,389	0.69	597,906	2,909	104,035	8,716	0.75	0.08	0.03	0.02
0.90	565	1,357	0.71	598,471	2,344	105,392	7,359	0.76	0.07	0.02	0.01

At $p = 0.85$ we get $FP/P = 0.05$, so if we immediately dispatched ambulances to crashes the model predicts with $p > 0.85$, we would increase the load on our ambulance fleet by the allowed 5%. At this decision threshold,

- Recall = 0.13, so we would be immediately dispatching ambulances to 13% of the people who need one.
- Precision = 0.71, so 71% of the ambulances we immediately dispatched would be needed, and
- mPrec = 0.61, so each ambulance we immediately dispatched would have at least a 61% chance of being needed.

2. **Immediately dispatch a total of two unneeded ambulances for each needed ambulance, i.e. Precision = $1/(2+1) = 1/3$.**

p	Neg	Pos	mPrec	TN	FP	FN	TP	Prec	Rec	FP/P	\hat{p}
0.41	10,889	1,438	0.12	370,854	229,961	21,063	91,688	0.29	0.81	2.04	0.45
0.42	10,778	1,476	0.12	381,632	219,183	22,539	90,212	0.29	0.80	1.94	0.43
0.43	10,541	1,504	0.12	392,173	208,642	24,043	88,708	0.30	0.79	1.85	0.42
0.44	10,283	1,455	0.12	402,456	198,359	25,498	87,253	0.31	0.77	1.76	0.40
0.45	10,095	1,622	0.14	412,551	188,264	27,120	85,631	0.31	0.76	1.67	0.38
0.46	9,770	1,592	0.14	422,321	178,494	28,712	84,039	0.32	0.75	1.58	0.37
0.47	9,459	1,650	0.15	431,780	169,035	30,362	82,389	0.33	0.73	1.50	0.35
0.48	9,164	1,770	0.16	440,944	159,871	32,132	80,619	0.34	0.72	1.42	0.34
0.49	8,761	1,716	0.16	449,705	151,110	33,848	78,903	0.34	0.70	1.34	0.32
0.50	8,578	1,787	0.17	458,283	142,532	35,635	77,116	0.35	0.68	1.26	0.31
0.51	8,255	1,787	0.18	466,538	134,277	37,422	75,329	0.36	0.67	1.19	0.29
0.52	7,940	1,871	0.19	474,478	126,337	39,293	73,458	0.37	0.65	1.12	0.28

At $p = 0.47$ we get Precision = 0.33, which fits our political constraints. Also at this decision threshold,

- Recall = 0.73, so we would be immediately dispatching ambulances to 73% of the people who need one.
- mPrec = 0.61, so each ambulance we immediately dispatched would have at least a 61% chance of being needed.
- $FP/P = 1.50$, so we would increase the number of ambulances being sent by 150%, which may not be possible in the short run and too expensive in the long run.

The political decision makers may choose to stay with this Precision metric but change it to something less expensive like Precision = 0.5, and with the data we have we could tell them the implications of that decision.

3. **Immediately dispatch ambulances to crashes with at least a 50% probability of needing one.**

p	Neg	Pos	mPrec	TN	FP	FN	TP	Prec	Rec	FP/P	\hat{p}
0.73	2,295	1,749	0.43	576,458	24,357	78,787	33,964	0.58	0.30	0.22	0.08
0.74	2,311	1,696	0.42	578,769	22,046	80,483	32,268	0.59	0.29	0.20	0.08
0.75	2,125	1,672	0.44	580,894	19,921	82,155	30,596	0.61	0.27	0.18	0.07
0.76	1,993	1,653	0.45	582,887	17,928	83,808	28,943	0.62	0.26	0.16	0.07
0.77	1,859	1,640	0.47	584,746	16,069	85,448	27,303	0.63	0.24	0.14	0.06
0.78	1,656	1,717	0.51	586,402	14,413	87,165	25,586	0.64	0.23	0.13	0.06
0.79	1,523	1,601	0.51	587,925	12,890	88,766	23,985	0.65	0.21	0.11	0.05
0.80	1,392	1,585	0.53	589,317	11,498	90,351	22,400	0.66	0.20	0.10	0.05
0.81	1,358	1,673	0.55	590,675	10,140	92,024	20,727	0.67	0.18	0.09	0.04
0.82	1,244	1,590	0.56	591,919	8,896	93,614	19,137	0.68	0.17	0.08	0.04

A crash report with $p = 0.78$ has a mPrec = 51% chance of needing an ambulance, and crashes with $p > 0.78$ have a higher chance, so immediately dispatch ambulances to crash reports with $p > 0.78$. Also at this decision threshold,

- Recall = 0.23, so we would be immediately dispatching ambulances to 23% of the people who need one.
- Prec = 0.64, so the ambulances we immediately dispatched would have at least a 64% chance of being needed.
- FP/P = 0.13, so we would increase the number of ambulances being sent by 13%.

In this case we would like to dig further into the data to see more precisely where in $p \in (0.77, 0.78)$ the value of mPrec is closest to 0.50, but except for some noise, this model only returns values of p to two decimal places. In this model, only 8% of the non-unique values of p have more than two places of precision. If we tried to dig deeper we would be looking at really small counts and see more randomness than actual insight.

1.8. Comparing Models

In all of these comparisons the Balanced Random Forest Classifier (BRFC) is clearly best, but I did not spend much time optimizing the models or test other algorithms. Mainly what I'm doing here is showing how I would compare the algorithms.

1. Send up to 5% more ambulances, i.e. FP/P = 0.05

Model	p	mPrec	Prec	Rec	FP/P
BRFC	0.88	0.62	0.71	0.13	0.05
KBFC	0.76	0.57	0.68	0.11	0.05
OBFC	0.57	0.57	0.67	0.11	0.05
AdaBoost	0.71	0.52	0.57	0.07	0.05
RUSBoost	0.72	0.49	0.58	0.07	0.05
LogReg	0.6	0.51	0.59	0.07	0.05
BalBag	0.9	0.56	0.68	0.07	0.03
EEC	0.84	0.49	0.52	0.06	0.05

2. Immediately dispatch a total of two unneeded ambulances for each needed ambulance, i.e. Precision = $1/(2+1) = 1/3$.

Model	p	mPrec	Prec	Rec	FP/P
BRFC	0.5	0.16	0.34	0.72	1.42
BalBag	0.48	0.17	0.32	0.66	1.41
KBFC	0.21	0.18	0.33	0.65	1.33
OBFC	0.26	0.18	0.33	0.65	1.32
LogReg	0.56	0.21	0.33	0.57	1.16
AdaBoost	0.51	0.2	0.33	0.57	1.14
RUSBoost	0.51	0.21	0.33	0.56	1.12
EEC	0.51	0.19	0.33	0.49	0.97

3. **Immediately dispatch ambulances to crashes with at least a 50% probability of needing one, i.e. mPrec = 0.50**

Model	p	mPrec	Prec	Rec	FP/P
BRFC	0.78	0.5	0.64	0.23	0.13
KBFC	0.88	0.48	0.6	0.18	0.12
BalBag	0.88	0.49	0.6	0.16	0.11
OBFC	0.5	0.5	0.62	0.16	0.1
AdaBoost	0.8	0.5	0.56	0.08	0.07
LogReg	0.54	0.5	0.58	0.08	0.06
RUSBoost	0.87	0.5	0.57	0.08	0.06
EEC	0.92	0.5	0.52	0.06	0.05

1.9. Comparing Models over Three Sets of Features

I ran the models on three sets of features. You can think of them as “Easy, Medium, and Hard” or “Cheap, Moderate, and Expensive.”

Easy The Easy features are the information the dispatchers already have (including time of day, day of week, weather) plus a bit of information from the location (like whether it’s on an interstate highway).

Medium The Medium features add more detailed information about the location (like whether it’s at an intersection or a parking lot) and a bit about the user of the phone (like age and sex).

Hard The Hard features add really detailed information about the location (like whether it’s in a work zone), correlates phone user information from the cell service provider with government and insurance records on vehicle ownership to guess at the kind of vehicle involved, and correlates multiple simultaneous notifications from the same location to guess at the number of people involved and whether it’s a school bus. Getting the “hard” features may also pose privacy issues.

Political decision makers can use the differences in the model results to decide whether to invest in the infrastructure to get the more expensive levels of data in real time.

Future work could give better detail by going through each feature, ranking how much it would cost to get that data and how much that feature would contribute to the quality of the model.

1. **Send up to 5% more ambulances, i.e. FP/P = 0.05**

Features	Model	p	mPrec	Prec	Rec	FP/P
Easy	BRFC	0.96	0.36	0.43	0.05	0.06
Medium	BRFC	0.91	0.5	0.57	0.07	0.05
Hard	BRFC	0.88	0.62	0.71	0.13	0.05

Recall goes from 5% to 13% of needed ambulances being dispatched immediately.

2. **Immediately dispatch a total of two unneeded ambulances for each needed ambulance, *i.e.* Precision = $1/(2+1) = 1/3$.**

Features	Model	p	mPrec	Prec	Rec	FP/P
Easy	BRFC	0.79	0.28	0.33	0.2	0.41
Medium	BRFC	0.59	0.21	0.33	0.51	1.02
Hard	BRFC	0.5	0.16	0.34	0.72	1.42

Recall goes from 20% to 72% of needed ambulances being dispatched immediately, but with the total number of ambulances being sent to crashes increasing 40% to 142%, which may not be possible in the budgeting process.

3. **Immediately dispatch ambulances to crashes with at least a 50% probability of needing one, *i.e.* mPrec = 0.50**

Features	Model	p	mPrec	Prec	Rec	FP/P
Easy	EEC	0.83	0.44	0.32	0.07	0.15
Medium	KBFC	0.73	0.5	0.54	0.08	0.07
Hard	BRFC	0.78	0.5	0.64	0.23	0.13

Recall goes from 7% to 23% of needed ambulances being dispatched immediately. Going from Easy to Medium does not significantly increase the recall, but it does decrease the ambulance cost, with the percentage of additional ambulances being sent to crashes going from 15% to 7%.

1.10. Marginal

Each of our models gives, for each sample x_i , a prediction $\text{predict_proba}(x_i) \in [0, 1]$ that the sample belongs to the positive class, *i.e.* that that crash person needs an ambulance. For some modeling algorithms, the prediction $\text{predict_proba}(x_i)$ is reasonably close to the posterior probability, but for other (otherwise useful) algorithms, we can only be confident that $\text{predict_proba}(x_i)$ increases with the probability $P(y_i = 1|x_i)$. ?

Because we are doing supervised learning (training and testing our model on data where we know the answer $y_i \in \{0, 1\}$ to whether or not the crash person needed an ambulance), we can find the probabilities directly. If we choose ranges of $\text{predict_proba}(x_i)$ large enough to smooth out the randomness, we can find the probability in that range from the ratio of the number of samples for which $y = 1$.

Sophisticated algorithms exist for recalibrating the $\text{predict_proba}(x_i)$ to give the posterior probability. One such method, called Platt Scaling, learns a sigmoid transformation, and another process uses isotonic regression to learn a monotonically increasing mapping from $\text{predict_proba}(x_i)$ to the posterior probability. The scikit-learn method `calibration_curve` implements both methods.

2. Introduction

2.1. Outline

- Dataset
 - CRSS
 - * 2016-2020, 2021
 - * Over-represents more serious crashes
 - Feature Selection and Engineering
 - Discretization
 - Imputing Missing Values
- Imbalanced Data
 - Can't use SMOTE

- Class Weights
- Focal Loss
- Bagging and Boosting Methods
- Moving the Discrimination Threshold
- Threshold Options
 - Choose the precision that is politically acceptable
 - Total Precision, including Prior Probability equals Posterior Probability
 - Marginal Precision
- Results and Conclusions

3. Total and Marginal Precision

Given a model and a choice of decision threshold θ , the total number of needed ambulances we send (TP) divided by the total number we send (FP + TP) is called the *precision*. Note that TP is all of the elements of the positive class with $p > \theta$, FP is all of the elements of the negative class with $p > \theta$, and FP+TP is all of the elements of either class with $p > \theta$.

The *marginal precision* at θ is the ratio of the number of positive samples to the total number of samples in the neighborhood of p around θ . The marginal precision the minimum probability that an ambulance sent is needed. In the language of economics, it is the probability that last ambulance sent is needed.

For example, if the decision makers are willing to send two unneeded ambulances (FP = $2k$ for some k) for every one that is needed (TP = $1k$), we look for the value of p where $\text{Prec} = \frac{1}{2+1} = 1/3$. If we want each ambulance sent to have at least a $1/3$ probability of being needed, then we look for the neighborhood of p where $m\text{Prec} = 1/3$.

The marginal precision is equivalent to the slope of the ROC curve, as there is an invertible mapping between them.

$$\begin{aligned}
 m\text{ROC} &= \frac{\Delta\text{TPR}}{\Delta\text{FPR}} = \frac{\Delta(\text{TP}/P)}{\Delta(\text{FP}/N)} \\
 &= \frac{(\Delta\text{TP})/P}{(\Delta\text{FP})/N} \quad (\text{because in a given model on a given data set, } P \text{ and } N \text{ are constant}) \\
 &= \frac{N}{P} \cdot \frac{\Delta\text{TP}}{\Delta\text{FP}} = \frac{N}{P} \cdot \frac{\Delta\text{TP}/\Delta p}{\Delta\text{FP}/\Delta p} = \frac{N}{P} \cdot \frac{\text{Pos}}{\text{Neg}} \\
 &= \frac{N}{P} \cdot \frac{1}{\frac{\text{Neg}}{\text{Pos}}} = \frac{N}{P} \cdot \frac{1}{\frac{\text{Neg}}{\text{Pos}} + 1 - 1} = \frac{N}{P} \cdot \frac{1}{\frac{\text{Neg}+\text{Pos}}{\text{Pos}} - 1} = \frac{N}{P} \cdot \frac{1}{\frac{1}{m\text{Prec}} - 1} = \frac{N}{P} \cdot \frac{m\text{Prec}}{1 - m\text{Prec}} \\
 m\text{Prec} &= \frac{P \cdot m\text{ROC}}{N + P \cdot m\text{ROC}}
 \end{aligned}$$

A challenge with calculating the marginal precision is choosing the margin ϵ for the neighborhood about p . If we make ϵ just large enough, the marginal precision will be a decreasing function of p and we will glean one value of θ where $m\text{Prec}$ is closest to the goal. Because our data set is discrete, however, too small values of ϵ will yield some neighborhoods with few or no values of the positive or negative class. Because two of our model algorithms give most values of p rounded to two decimal places, we have chosen to use one hundred non-overlapping intervals of p ($\epsilon = 0.005$) for our analysis.

The table below gives the values for each of a hundred p neighborhoods for one of our models. Looking at $p = 0.45$ and 0.46 , for instance, for $p \in [0, 0.45]$ the model has correctly classified 117,225 of the 180,245 elements of the negative class and 26,573 of the 33,825 elements of the positive class. Moving from $p = 0.45$ to $p = 0.46$, the model correctly classifies 3,079 more elements of the negative class and 436 fewer elements of the positive class.

Claim: The precision is an increasing function is equivalent to

$$\frac{\text{Pos}}{\text{Neg}} < \frac{\text{TP}}{\text{FP}}$$

which is not necessarily true if we zoom in to a sufficiently small interval of p , because of the stochastic and discrete nature of our data set. Over sufficiently large intervals of p , however, it is generally true that precision is an increasing function of the decision boundary θ , being equivalent to the ROC curve curving down.

If the politicians have decided that they will trade off two immediately dispatched ambulances for each needed ambulance, then we choose the decision threshold θ at the value of p where $\text{Prec} = 0.33$, which happens around $p = 0.49$. Note that in this case we would be sending some ambulances to some crashes where there is only a 15% chance that the ambulance is needed. Similarly for other political decisions about total tradeoffs; we will investigate $\text{Prec} = 1/2$ and $\text{Prec} = 2/3$.

If the politicians decide that they want to automatically dispatch ambulances only to notifications where the likelihood that the victim requires an ambulance is greater than $1/3$, then choose the decision threshold θ at the value of p where $m\text{Prec} = 0.33$, which happens around $p = 0.68$. The marginal precision is much more volatile than the total precision, but we can narrow it down to somewhere in that region. At this value of θ over half, $13,617/(12,766 + 13,617) \approx 0.52$, of the ambulances that we automatically dispatch turn out to be needed. Similarly for other politically-chosen minimum percentages; we will also investigate $1/2$ and $2/3$.

Balanced Random Forest Classifier, Hard features, No Tomek undersampling, No class weights, Test set, Version 1

p	Neg	Pos	mPrec	TN	FP	FN	TP	Prec	Rec	\hat{p}
0.00	107	0	0.00	107	180,138	0	33,825	0.16	1.00	1.00
0.01	238	3	0.01	345	179,900	3	33,822	0.16	1.00	1.00
0.02	331	2	0.01	676	179,569	5	33,820	0.16	1.00	1.00
0.03	441	3	0.01	1,117	179,128	8	33,817	0.16	1.00	0.99
0.04	526	6	0.01	1,643	178,602	14	33,811	0.16	1.00	0.99
0.05	751	6	0.01	2,394	177,851	20	33,805	0.16	1.00	0.99
0.06	854	8	0.01	3,248	176,997	28	33,797	0.16	1.00	0.98
0.07	1,060	9	0.01	4,308	175,937	37	33,788	0.16	1.00	0.98
0.08	1,235	13	0.01	5,543	174,702	50	33,775	0.16	1.00	0.97
0.09	1,375	16	0.01	6,918	173,327	66	33,759	0.16	1.00	0.97
0.10	1,653	16	0.01	8,571	171,674	82	33,743	0.16	1.00	0.96
⋮										
0.45	3,206	424	0.12	117,225	63,020	7,252	26,573	0.30	0.79	0.42
0.46	3,079	436	0.12	120,304	59,941	7,688	26,137	0.30	0.77	0.40
0.47	3,021	547	0.15	123,325	56,920	8,235	25,590	0.31	0.76	0.39
0.48	2,990	453	0.13	126,315	53,930	8,688	25,137	0.32	0.74	0.37
0.49	3,020	533	0.15	129,335	50,910	9,221	24,604	0.33	0.73	0.35
0.50	2,874	501	0.15	132,209	48,036	9,722	24,103	0.33	0.71	0.34
0.51	2,804	533	0.16	135,013	45,232	10,255	23,570	0.34	0.70	0.32
0.52	2,675	542	0.17	137,688	42,557	10,797	23,028	0.35	0.68	0.31
0.53	2,543	526	0.17	140,231	40,014	11,323	22,502	0.36	0.67	0.29
0.54	2,438	545	0.18	142,669	37,576	11,868	21,957	0.37	0.65	0.28
0.55	2,350	579	0.20	145,019	35,226	12,447	21,378	0.38	0.63	0.26
⋮										
0.60	1,877	587	0.24	155,512	24,733	15,419	18,406	0.43	0.54	0.20
0.61	1,756	597	0.25	157,268	22,977	16,016	17,809	0.44	0.53	0.19
0.62	1,674	632	0.27	158,942	21,303	16,648	17,177	0.45	0.51	0.18
0.63	1,611	604	0.27	160,553	19,692	17,252	16,573	0.46	0.49	0.17
0.64	1,582	586	0.27	162,135	18,110	17,838	15,987	0.47	0.47	0.16
0.65	1,439	618	0.30	163,574	16,671	18,456	15,369	0.48	0.45	0.15
0.66	1,376	561	0.29	164,950	15,295	19,017	14,808	0.49	0.44	0.14
0.67	1,288	637	0.33	166,238	14,007	19,654	14,171	0.50	0.42	0.13
0.68	1,241	554	0.31	167,479	12,766	20,208	13,617	0.52	0.40	0.12
0.69	1,082	631	0.37	168,561	11,684	20,839	12,986	0.53	0.38	0.12
0.70	1,053	570	0.35	169,614	10,631	21,409	12,416	0.54	0.37	0.11
0.71	922	587	0.39	170,536	9,709	21,996	11,829	0.55	0.35	0.10
0.72	897	559	0.38	171,433	8,812	22,555	11,270	0.56	0.33	0.09
0.73	783	587	0.43	172,216	8,029	23,142	10,683	0.57	0.32	0.09
0.74	831	558	0.40	173,047	7,198	23,700	10,125	0.58	0.30	0.08
0.75	711	602	0.46	173,758	6,487	24,302	9,523	0.59	0.28	0.07
⋮										
0.95	74	251	0.77	180,091	154	33,222	603	0.80	0.02	0.00
0.96	61	211	0.78	180,152	93	33,433	392	0.81	0.01	0.00
0.97	55	168	0.75	180,207	38	33,601	224	0.85	0.01	0.00
0.98	22	125	0.85	180,229	16	33,726	99	0.86	0.00	0.00
0.99	12	66	0.85	180,241	4	33,792	33	0.89	0.00	0.00
1.00	4	33	0.89	180,245	0	33,825	0	nan	0.00	0.00

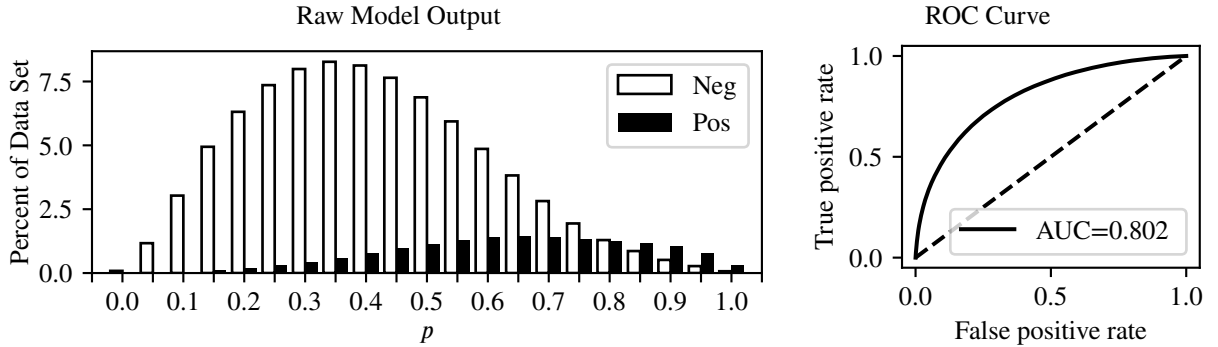
Visually on the histogram below, when we say that the precision of $1/3$ happens at $p \approx 0.50$, we mean that on the interval $p \in [0.50, 1.0]$, the area under the Neg curve is twice the area under the Pos curve. When we say that the marginal precision of $1/3$ happens at $p \approx 0.7$, we mean that the Neg bar at $p = 0.7$ is twice as tall as the Pos bar.

On the ROC curve below, $m\text{Prec} = 1/3$ happens at $p \approx 0.7$, on the curve at $(\text{FPR}, \text{TPR}) = (0.06, 0.37)$ because

$$\text{FPR} = \frac{\text{FP}}{\text{N}} = \frac{10,631}{180,091} = 0.06 \quad \text{and} \quad \text{TPR} = \frac{\text{TP}}{\text{P}} = \frac{12,416}{33,825} = 0.37$$

and at that point

$$m\text{ROC} = \frac{\text{N}}{\text{P}} \cdot \frac{m\text{Prec}}{1 - m\text{Prec}} = \frac{180,254}{33,825} \cdot \frac{1/3}{1 - 1/3} \approx 3$$



4. Literature Review

5. Dataset

6. Methods

6.1. Analysis of Results

Our ML algorithms assign to each sample (feature vector, crash person) a probability $p \in [0, 1]$ that the person needs an ambulance. The histogram below left shows the percentage of the dataset in each range of p , showing the percentages for the negative class (“Does not need an ambulance”) and the positive class (“Needs an ambulance”). On the right, the Receiver Operating Characteristic (ROC) curve, and particularly the area under the curve (AUC), is a metric for how well the model separates the two classes, with $AUC = 1.0$ being perfect and $AUC = 0.5$ (the dashed line) being just random assignment with no insight.

We would love to have results like in the graphs below, where the machine learning (ML) algorithm nearly perfectly separates the two classes. There is some overlap between $p = 0.6$ and $p = 0.8$ with some samples the algorithm misclassifies, but the model clearly separates most samples. Having an AUC of 0.996 would be amazing.

[Put in `BRFC_Hard_alpha_0_5_Train_Pred_Wide.pgf` once we have it.)

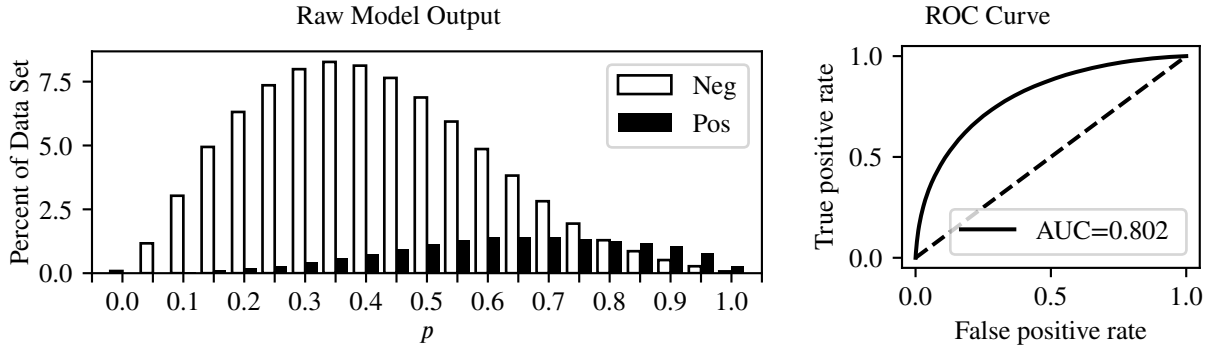
Unfortunately, our test results do not look quite that nice. They do not separate the two classes as well. Some distributions are clustered to one side or in the middle. Some models give the results in $p \in [0, 1]$ rounded to two decimal places so that we cannot hope for a level of detail beyond that, and one algorithm, Bagging, gives p rounded to only one decimal place.

Let us look at some examples. In all of them, AUC is in the range $[0.7, 0.8]$, so the various models separate the positive and negative classes about equally well overall, with none being dramatically better or worse. We will later show how we investigated which models do a better job in the ranges of interest.

BRFC_5_Fold_alpha_0_5_Hard_Test

This model does not separate the negative and positive classes as well as the ideal, giving a much lower AUC (area under the ROC curve). These results are actually from the same model as the ideal above, but the ideal are the results on the training set and below on the test set, showing overfitting.

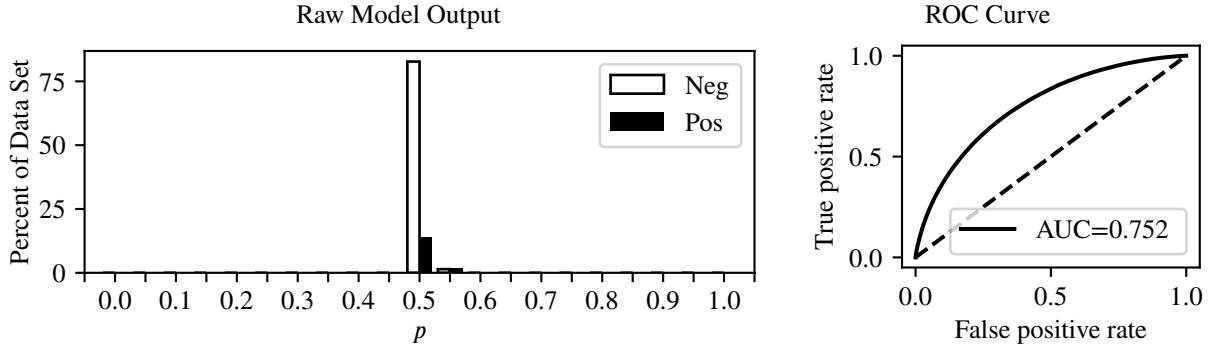
In these results, the 100 most frequent values comprised 93% of the results, meaning that, while there is some noise making the distribution look continuous, it is mostly discrete to two decimal places, so we cannot hope for fine detail in tuning the decision threshold.



AdaBoost_5_Fold_Hard_Test

In this model the values are clustered very tightly, but in that small range the 214,070 samples return 210,442 different values of p , so there is much diversity that we can't see in this representation.

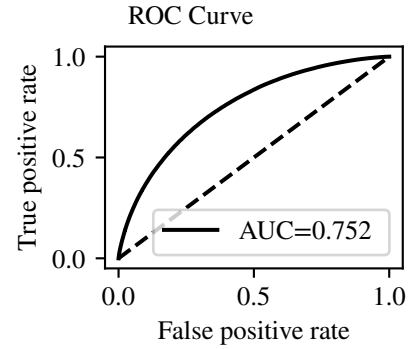
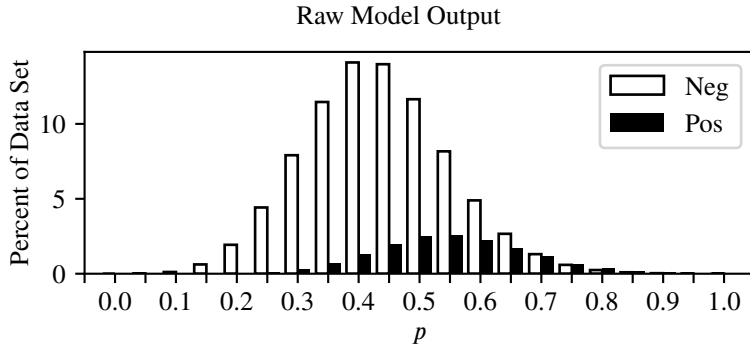
AdaBoost_5_Fold_Hard_Test



In this work we used two methods to give the results of different models similar distributions. This case illustrates directly transforming the y_{proba} values.

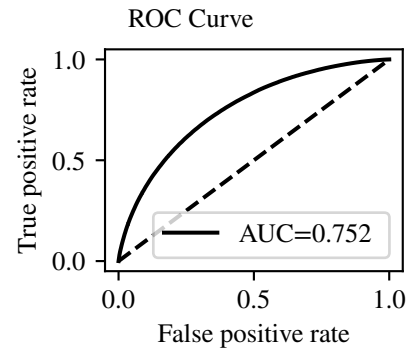
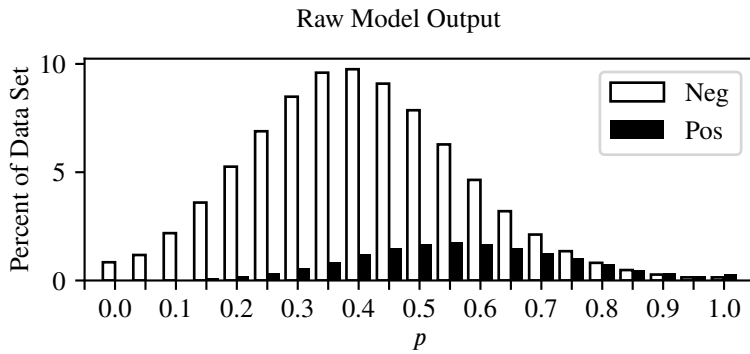
To make a useful visualization of the results where we can see the interplay between the negative and positive classes, we can transform the data. A transformation that preserves rank will have no effect on the ROC curve. [Cite] For the graph below, we mapped the smallest value in the set to 0 and the largest to 1.

AdaBoost_5_Fold_Hard_Test_Transformed_100



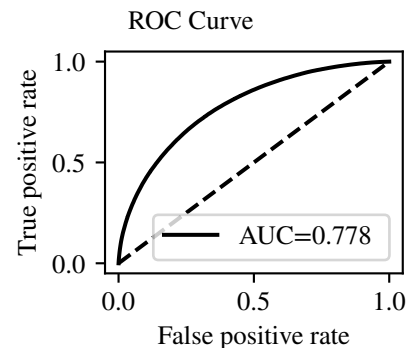
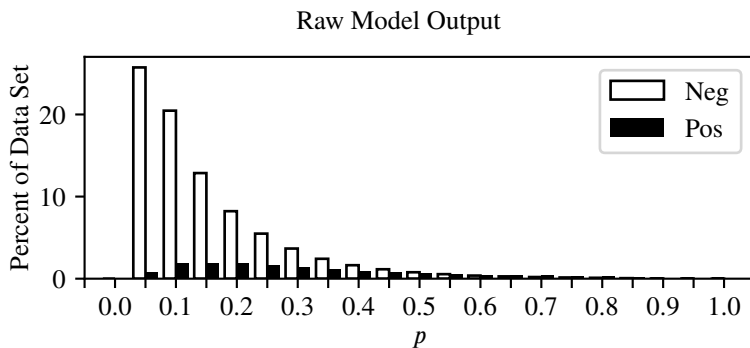
The distribution has long tails, so we can make a more useful visualization by truncating the ends. For this graph we mapped the 0.01 quantile to 0 and the 0.99 quantile to 1 leaving the center 98% of the distribution and truncated the ends. Our goal in clipping the tails is to make all of the models' results have approximately the same granularity when we choose the decision thresholds that give us the (politically) desired results.

AdaBoost_5_Fold_Hard_Test_Transformed_98



The model below is as effective at separating the two classes (ROC = 0.778), but the distribution is skewed to the left. Its results were nearly continuous, with the 214,070 samples returning 210,157 unique values of p , so we can fine tune the decision threshold.

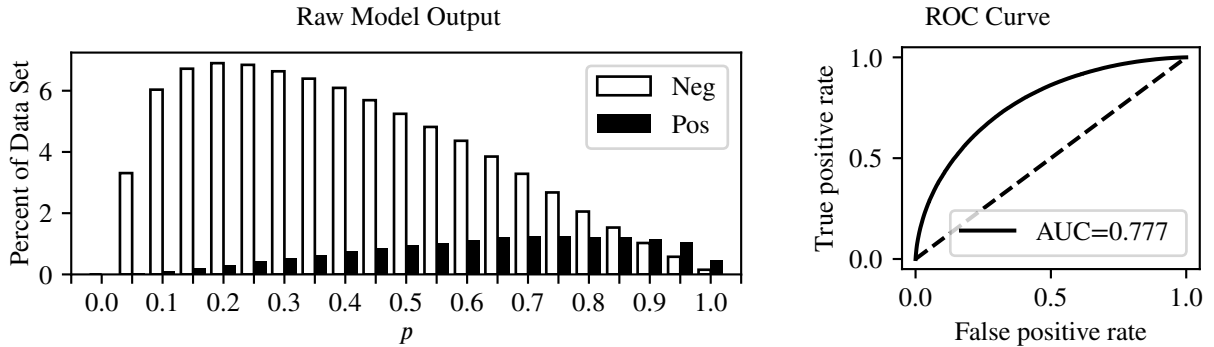
KBFC_5_Fold_alpha_0_5_gamma_0_0_Hard_Test



The second method we will use to modify the model outputs' distribution is to employ class weights in the model building process. Here we employed class weights proportional to the class imbalance. The motivation behind class weights is to better separate the positive and negative classes, but note that the area under the ROC curve does not

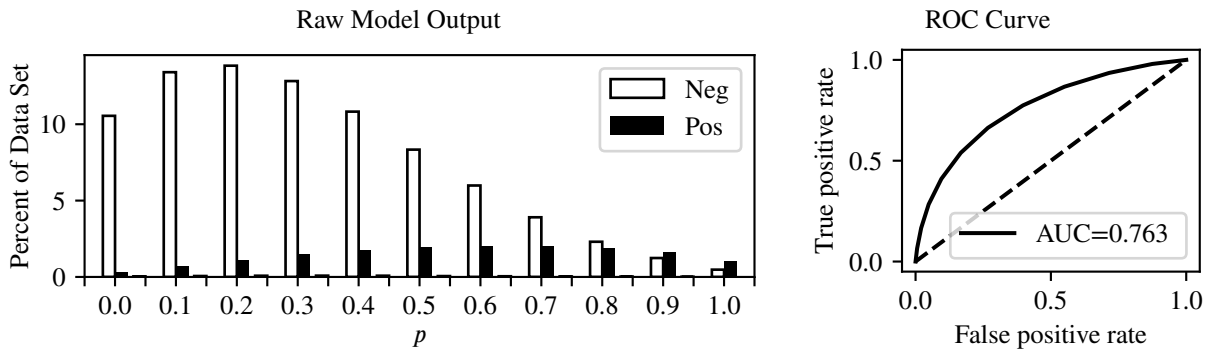
change. We have not investigated whether the model using class weights does a better job at separating the classes in some intervals, but overall the effect is negligible. One effect using class weights did have here is shifting the distribution.

KBFC_5_Fold_alpha_balanced_gamma_0_0_Hard



Bagging_Hard_Tomek_0_v1_Test

This model returned 217 different values, but most of them were rare. Taking out the 5% of the data set with the least frequent values, 95% of the samples had only 10 values of p . It may be a useful model, but we will not be able to fine tune the decision threshold.



Other stuff

7. Results

8. Conclusions

9. Discussion

10. Future Work

11. To Do, Notes to Self

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>

Declaration of Generative AI and AI-assisted technologies in the writing process

12.

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