Imbalanced Data in Classification: Opioid Misuse Prediction

Peixuan Zhang
Department of Industrial and manufacturing engineering
The Pennsylvania State University

***Abstract*—In this paper, we explored the multiple machine learning methods available for the classification problems with imbalanced data sets and conducted a case study on Opioid Misuse Prediction. Findings suggest that machine learning techniques combined with appropriate sampling techniques can be quite promising for the prediction of opioid misuse.**

***Index Terms*—classification, imbalanced data, opioid misuse, machine learning, feature selection.**

I. Literature Review

A. Introduction

In the real world applications, the big data are generated from highly skewed distributions, which is referred as imbalanced data sets. The imbalanced classification problems are concerned with the performance of learning algorithms in the presence of minority class. Developing an effective method for imbalanced classification problems can be well applied to the many areas such as detecting fraud in banking operations, and medical diagnosis prediction of rare but important disease [1].

The main techniques to address the imbalanced classification problems can be categorized into two types [1]: data-preprocessing approach and the algorithmic approach.

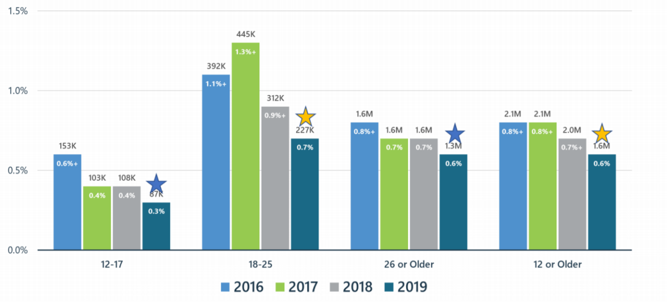
1) Data Preprocessing Techniques: The most popular data preprocessing techniques used in the imbalanced classification problems are oversampling and undersampling. In order to compensate for an imbalance in the skewed distribution of the target class, both oversampling and undersampling are designed to select more samples from one class than from another.

The first oversampling method is random oversampling, which refers to randomly choosing samples in the minority class with replacement. In [2], this method is proven to be robust. N.Chawla, K.Bowyer, L.Hall and W.Kegelmeyers in [3] introduced Synthetic Minority Over-sampling Technique (SMOTE) and proved it can achieve better classifier performance.

Instead of adding more samples to minority class, undersampling removes samples from the majority class, with or without replacement. However, it may increase the variance of the classifier and may not be able to provide enough training samples for classifier in some cases [4].

2) Algorithmic approach: In addition to those well-known classifiers (GLM, SVM [5], Random Forest [6], etc.), Chris Seiffert and Taghi M. Khoshgoftaar [7] proposed a hybrid sampling/boosting algorithm to alleviating class imbalance, that is, RUSboost. The RUSboost provides a simpler and faster alternative to SMOTEBoost, which is another algorithm that combines boosting and SMOTE techniques [3].

Fig. 1. Opioid Misuse in 2016-2019



G.King and L.Zeng talked about Rare Events Logistic Regression (relogit) for dichotomous dependent variables in [8] and [9]. The relogit model estimates the same model as the standard logistic regression. However, relogit corrects the bias that occurs when the observed events are rare (i.e., if the dependent variable has many more 0s than 1s or the reverse).

B. Motivation

Opioids have analgesic and sedative effects, and are commonly used for the management of pain. The common risks of using prescription opioids include dependence and addiction. Figure 1 provided by [10] shows opioid misuse, addiction, and overdoses remain serious public health problems in the United States in different age groups. In [11], Dae-Hee Han, Shieun Lee and Dong-Chul Seo found that machine learning (ML) can offer a promising and efficient technique in the prediction of adolescent opioid misuse.

The paper [11] and [12] inspire us to ask the question, ”Will we be able to construct a well performed model for the prediction of opioid misuse for all age groups? Is there a better model?”

Furthermore, if there exists a ”best” model for the opioid misuse, it could be applied to the similar problems, such as other drug use, cancer detection, or even more general imbalanced classification problems.

The organization of this paper is as follows. In the next section we introduce the problem of interest, opioid misuse classification problem. After discussing about the data set in section 3, we outline ML models used in the analysis in section 4. The analysis and the numerical results will be given in section 5 and 6. In the last two sections, we will talk about the future work and the difficulties encountered in our project.

II. Problem Definition

Opioid misuse can be regarded as imbalanced classification problems. People who misused opioid account for a small percentage of the whole population. In other words, the distribution of misusing opioid or not is highly skewed. As a result, the prediction for opioid misuse is an obvious imbalanced classification problem.

The goal of this study was (1) evaluate predictive performance of different ML techniques for opioid misuse (2) exploit classifiers with oversampling method.

III. Data

A. Raw data set

The data used in this project is from National Survey on Drug Use and Health (NSDUH-2019) which is a public source of statistical information on the use of illicit drugs, alcohol, and tobacco and on mental health issues among people aged 12 or older in the United States. The original data set is consisting of 56,136 records with 1,742 features (including the missing values).

B. Meaures

The target variable of the current study was a binary measure of opioid misuse (OPINMYR). The independent variables were preliminarily selected based on the results from the prior literature ([12]; [11]; [13]). At last, we chose 40 independent variables including sociodemographic variables, such as sex, income, race, and health-related variables, such as Tobacco/Alcohol/Marijuana use experience, health condition, etc.

C. Missing Values

Almost half of the independent variables have more than 60% missing values as it is shown in the Figure 2. However, it is reasonable to transform those missing values to ”0” for most variables in Figure 2 according to the corresponding responses of missing values. Then the rest missing values account for less than 3% of the whole data set. It can be eliminated without affecting the results. After cleaning the raw data set, there were 42,739 instances left.

D. Data dictionary

The data is consisting of 40 independent variables and one binary dependent variable (opioid misuse). Only one independent variable, K6SCMON, is quantitative variable. The rest independent variables are either binary factors or multi-level factors. More information about the data dictionary is included in the Table I.

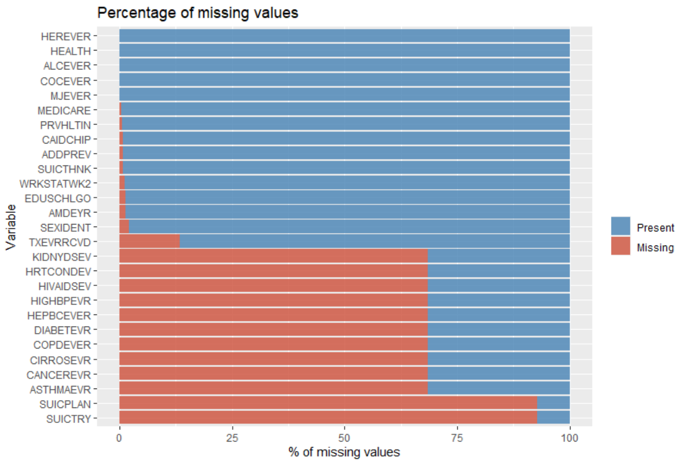


Fig. 2. Missing Values in the Opioid Misuse Data Set

IV. Model Discussion

Prediction models for opioid misuse were developed via three different ML algorithms: Logistic Regression, Decision Tree, Random Forest, artificial neural networks, RUSBoost and Relogit.

- Logistic regression (LR) [8]: the logistic model (or logit model) is used to model the probability of opioid misuse. With the ”L1” (or ”L2”) penalization, the LR model becomes penalized LR model.
- Decision Tree (DT) [14]: a tree-like structure model which include the internal nodes as the test on the attribute. Each ending node a class label.
- Random Forests (RF) [15]: consist of multiple decision trees and merges them together to get a more accurate and stable prediction.
- Multilayer Perceptron (MLP) [16]: a class of feedforward artificial neural network (ANN) which contains one or more hidden layers (apart from one input and one output layer).
- RUSBoost [17]: combine data sampling strategy and boosting, providing a simple and efficient method for imbalanced classification problems.
- Zelig [8] [9]: Rare Events Logistic Regression for Dichotomous Dependent Variables with bias corrected for the estimates.

V. Analysis

A. Data Preparation and Exploration

Data cleaning procedure involved dealing with missing values and data normalization. Next, Pearson’s chi-squared statistic was applied to testing the between-group differences between opioid misuse and non-misuse.

To prepare the data, we considered about three kinds of data set:

- Data with full features: 40 independent variables.

TABLE I
ALL BASELINE VARIABLES STRATIFIED BY OPINMYR

	0 (n = 39279)	1 (n = 1886)	p test	standardized mean differences	Descriptions (Variable Type)
AGE2 (%)			< 0.01	0.294	Age groups (categorical)
> 65	3706 (9.4)	60 (3.2)			
18∼25	12969 (33.0)	729 (38.7)			
25∼35	7800 (19.9)	460 (24.4)			
35∼65	14804 (37.7)	637 (33.8)			
SEXIDENT (%)			< 0.001	0.274	Sexual identity (categorical)
1 = Heterosexual/straight	36046 (91.8)	1563 (82.9)			
2 = Lesbian or Gay	890 (2.3)	69 (3.7)			
3 = Bisexual	2343 (6.0)	254 (13.5)			
IRMARIT (%)			< 0.001	0.329	Marital status (categorical)
1 = Married	15897 (40.5)	501 (26.6)			
2 = Widowed	1246 (3.2)	32 (1.7)			
3 = Divorced or Separated	4137 (10.5)	225 (11.9)			
4 = Never Been Married	17999 (45.8)	1128 (59.8)			
WRKSTATWK2 (%)			< 0.001	0.298	Work status (categorical)
1 = Worked at full-time job	19354 (49.3)	910 (48.3)			
2 = Worked at part time job	5277 (13.4)	225 (11.9)			
3 = Has job or volunteer worker	2251 (5.7)	126 (6.7)			
4 = Unemployed/on layoff	1767 (4.5)	145 (7.7)			
5 = Disabled	1378 (3.5)	100 (5.3)			
6 = Keeping house full-time	1734 (4.4)	66 (3.5)			
7 = In school/training	1426 (3.6)	57 (3.0)			
8 = Retired	2986 (7.6)	51 (2.7)			
9 = Does not have a job	3106 (7.9)	206 (10.9)			
INCOME (%)			< 0.001	0.212	Income (categorical)
1 = Less than $20,000	6950 (17.7)	451 (23.9)			
2 = $20,000 - $49,999	11791 (30.0)	633 (33.6)			
3 = $50,000 - $74,999	6351 (16.2)	270 (14.3)			
4 = $75,000 or More	14187 (36.1)	532 (28.2)			
COUTYP4 (%)			0.081	0.054	County metro/nonmetro status (categorical)
1 = Large Metro	17551 (44.7)	854 (45.3)			
2 = Small Metro	13968 (35.6)	698 (37.0)			
3 = Nonmetro	7760 (19.8)	334 (17.7)			
EDUSCHLGO = 1 (Yes) (%)	32409 (82.5)	1597 (84.7)	0.017	0.059	Go to school (binary)
NEWRACE2 (%)			< 0.001	0.195	Race (categorical)
1 = NonHisp White	23469 (59.7)	1162 (61.6)			
2 = NonHisp Black/Afr Am	5011 (12.8)	215 (11.4)			
3 = NonHisp Native Am/AK Native	473 (1.2)	47 (2.5)			
4 = NonHisp Native HI/Other Pac Isl	200 (0.5)	10 (0.5)			
5 = NonHisp Asian	1917 (4.9)	38 (2.0)			
6 = NonHisp more than one race	1323 (3.4)	83 (4.4)			
7 = Hispanic	6886 (17.5)	331 (17.6)			
BOOKED = 1 (Yes) (%)	6057 (15.4)	692 (36.7)	< 0.001	0.499	Ever arrested and booked for breaking the law (binary)
HEALTH (%)			< 0.001	0.327	Health condition (categorical)
1 = Excellent	8947 (22.8)	254 (13.5)			
2 = Very good	14744 (37.5)	636 (33.7)			
3 = Good	11221 (28.6)	636 (33.7)			
4 = Fair	3740 (9.5)	293 (15.5)			
5 = Poor	627 (1.6)	67 (3.6)			
HRTCONDEV = 1 (Yes) (%)	2611 (6.6)	120 (6.4)	0.662	0.012	Ever told had heart condition (binary)
DIABETEVR = 1 (Yes) (%)	2740 (7.0)	113 (6.0)	0.110	0.040	Ever told Diabetes (binary)
COPDEVER = 1 (Yes) (%)	1070 (2.7)	78 (4.1)	< 0.001	0.078	Ever told had COPD (binary)
CIRROSEVR = 1 (Yes) (%)	76 (0.2)	8 (0.4)	0.056	0.042	Ever told had cirrhosis of the liver (binary)
HEPBCEVER = 1 (Yes) (%)	291 (0.7)	93 (4.9)	< 0.001	0.254	Ever told had Hepatitis B or C (binary)
KIDNYDSEV = 1 (Yes) (%)	495 (1.3)	30 (1.6)	0.253	0.028	Ever told had kidney disease (binary)
ASTHMAEVR = 1 (Yes) (%)	4308 (11.0)	254 (13.5)	0.001	0.076	Ever told had asthma (binary)
HIVAIDSEV = 1 (Yes) (%)	56 (0.1)	8 (0.4)	0.006	0.053	Ever told had HIV or AIDS (binary)
CANCEREVR = 1 (Yes) (%)	1445 (3.7)	51 (2.7)	0.032	0.055	Ever told had cancer (binary)
HIGHBPEVR = 1 (Yes) (%)	4778 (12.2)	215 (11.4)	0.338	0.024	Ever told had high blood pressure (binary)
K6SCMON (mean (SD))	4.88 (5.06)	8.32 (6.38)	< 0.001	0.598	K6 total score (numeric)
ADDPREV = 1 (Yes) (%)	26145 (66.6)	869 (46.1)	< 0.001	0.422	Felt sad most time (binary)
SUICPLAN = 1 (Yes) (%)	801 (2.0)	140 (7.4)	< 0.001	0.256	Had suicidal plan (binary)
SUICTHNK = 1 (Yes) (%)	2646 (6.7)	353 (18.7)	< 0.001	0.365	Had suicidal thoughts (binary)
SUICTRY = 1 (Yes) (%)	340 (0.9)	80 (4.2)	< 0.001	0.215	Had suicidal attempts (binary)
TXEVRRCVD = 1 (Yes) (%)	2208 (5.6)	466 (24.7)	< 0.001	0.552	Received drug/alcohol treatment (binary)
AMDEYR = 1 (Yes) (%)	3921 (10.0)	464 (24.6)	< 0.001	0.394	Felt depressive (binary)
CIGEVER = 1 (Yes) (%)	22115 (56.3)	1519 (80.5)	< 0.001	0.540	Ever smoked a cigarrete (binary)
ALCEVER = 1 (Yes) (%)	33548 (85.4)	1787 (94.8)	< 0.001	0.316	Ever drank alcohol (binary)
MJEVER = 1 (Yes) (%)	20383 (51.9)	1561 (82.8)	< 0.001	0.697	Ever used marijuana binary
COCEVER = 1 (Yes) (%)	5459 (13.9)	914 (48.5)	< 0.001	0.804	Ever used cocaine (binary)
HEREVER = 1 (Yes) (%)	609 (1.6)	325 (17.2)	< 0.001	0.558	Ever used heroin (binary)
MEDICARE = 1 (Yes) (%)	4512 (11.5)	121 (6.4)	< 0.001	0.178	Insurance covered by Medicare (binary)
CAIDCHIP = 1 (Yes) (%)	6950 (17.7)	528 (28.0)	< 0.001	0.247	Insurance covered by Medicaid (binary)
PRVHLTIN = 1 (Yes) (%)	25052 (63.8)	968 (51.3)	< 0.001	0.254	Covered by private insurance (binary)
UDPYOPI = 1 (Yes) (%)	2 (0.0)	311 (16.5)	< 0.001	0.628	Recoded Opioid dependence (binary)
ABODALC = 1 (Yes) (%)	2624 (6.7)	408 (21.6)	< 0.001	0.439	Alcohol dependence (binary)
ABODMRJ = 1 (Yes) (%)	929 (2.4)	210 (11.1)	< 0.001	0.355	Marijuana dependence (binary)
ABODCOC = 1 (Yes) (%)	104 (0.3)	102 (5.4)	< 0.001	0.314	Cocaine dependence (binary)
ABODHER = 1 (Yes) (%)	2 (0.0)	102 (5.4)	< 0.001	0.338	Heroin dependence (binary)

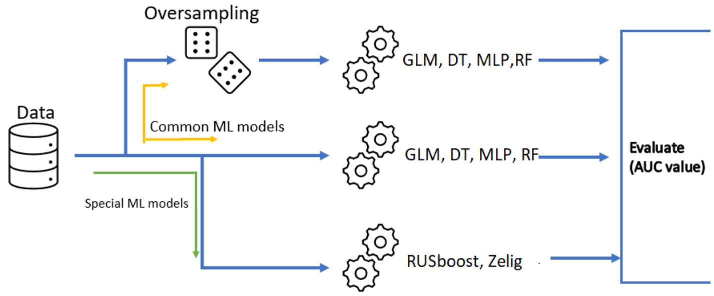


Fig. 3. Flowchart

- Data after feature selection: we employed χ^2 feature selection method to select a subset of 40 independent variables.
- Data after dimension reduction: Principal Component Analysis (PCA) was used in exploratory data analysis and for making predictive models.

In the following section 6, we showed the models' performance after feature selection or dimension reduction as a comparison to the performance of models with all 40 features.

B. Modeling Procedure

After examining descriptive statistics and preparing data, we fitted six ML prediction models of opioid misuse using three strategies.

1) Use standard ML algorithms include Logistic regression, penalized Logistic regression, Decision Tree, Random Forests, and Multilayer Perceptron.
2) The same standard ML methods are adopted, followed by the oversampling procedure.
3) Employ special ML models, RUSBoost and Relogit for this imbalanced classification problem.

Figure 3 is the flowchart of our modeling procedure, which is a graphic illustration of the three strategies.

C. Model Evaluation

1) Cross Validation: In order to precisely evaluate the models' performance, we used 10-fold stratified cross validation, instead of the standard cross validation. The reason lies in the fact that each fold contains roughly the same proportions of the two types of class labels, misused opioid and not misused opioid.

2) Evaluation Metric: Because of the highly skewed distribution of opioid misuse, AUC was chosen to evaluate the model's prediction ability. AUC, Area Under the ROC Curve, provides an aggregate measure of performance across all possible classification thresholds.

VI. Results and Conclusion

A. Descriptive Statistics

As shown in Table I, among all 41165 people in the data sample, 4.58% (n = 1521) misused opioids in the past year. Compared to respondents who did not misuse opioids, those who misused showed significant differences in terms of most sociodemographics, and health-related factors. This finding is consistent with the conclusion in [12].

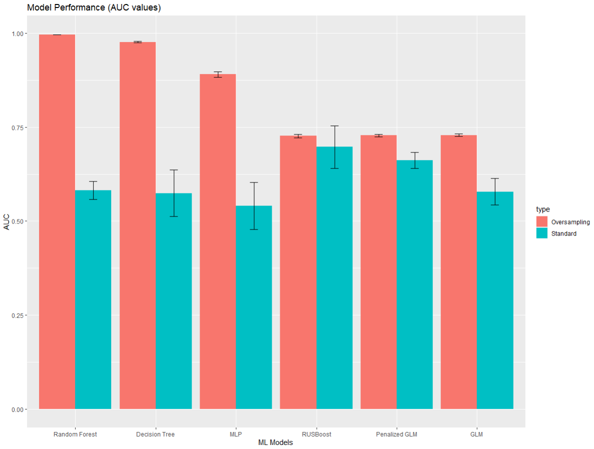


Fig. 4. Model Performance

B. Model performance with all features

TABLE II
Model Performance based on AUC values

	Model	AUC (sd)
Standard	GLM	0.578 (0.0356)
	penalized GLM	0.662 (0.0214)
	Decision Tree	0.575 (0.0620)
	Random Forest	0.582 (0.0238)
	MLP	0.541 (0.0622)
	RUSBoost	**0.697** (0.0567)
	Relogit	**0.724** (0.0119)
Oversampling	GLM	0.729 (0.00323)
	penalized GLM	0.728 (0.00328)
	Decision Tree	**0.977** (0.00180)
	Random Forest	**0.997** (0.00064)
	MLP	0.891 (0.00722)
	RUSBoost	0.727 (0.00447)

From Table II and Figure 4 both show the best-performed ML models are Random Forest (0.997) and Decision Tree (0.977) after oversampling.

When implementing the standard ML algorithms without oversampling, the penalized logistic regression performed slightly better. Besides, RUSBoost and Relogit, special ML models for the imbalanced classification, improve the predictive models' performance. This proved that RUSBoost and Relogit are more effective for the imbalanced data set.

However, RUSBoost and Relogit only slightly improve the models' performance to around 0.7 with respect to the AUC values. After increasing the ratio of the minority group (misued opioid in the past year) in the training data set, all the standard ML models significantly enhanced predictive ability, especially Random Forest (0.337) and Decision Tree (0.977).

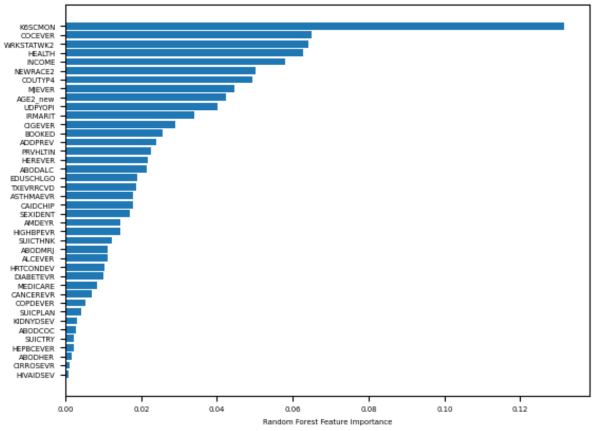


Fig. 5. Importance of variables in the Random Forest

TABLE III
VARIABLE IMPORTANCE IN THE RANDOM FOREST

	Variable	Importance
1	K6 total score	0.1317
2	Ever used cocaine	0.0650
3	Work status	0.0641
4	Health status	0.0628
5	Income	0.0578
6	Race2	0.0501
7	County metro/nonmetro status	0.0495
8	Ever used marijuana	0.0446
9	Age group	0.0424
10	Recoded Opioid dependence	0.0401

To explain the model, the importance of the all variables in the best-performance model, Random Forest, is given in the Figure 5.

The top 10 important variables are listed in the Table III. K6 score is ranked to be the most influential explanatory variable in explaining people past-year opioid misuse followed by the exposure to cocaine, work status and self-evaluated health status. It is interesting to notice that there is no significant difference between respondents who misused opioid and who did not misuse opioid in terms of County metro/nonmetro status (COUTYP4) in Table I. However, it ranked the 7th most important variable in Random Forest model.

C. Model performance with feature selection or dimension reduction

Implementing the same ML algoritihms on data set after feature selection or dimension reduction, the corresponding results are shown in Table IV.

Compared to the results obtained from the model with all 40 features, feature selection and dimension reduction did not significantly improve the models' predictive ability.

D. Time Complexity

When running the different ML models, we noticed that it took much longer to train the MLP models regardless of the number of features. However, longer training time does not guarantee better predictive ability. Random Forest outperformed MLP in most cases.

TABLE IV
MODEL PERFORMANCE WITH FEATURE SELECTION OR DIMENSION REDUCTION

	Model	AUC (χ^2)	AUC (PCA)
Standard	GLM	0.578	0.660
	penalized GLM	0.661	0.661
	Decision Tree	0.611	0.614
	Random Forest	0.583	0.583
	MLP	0.603	0.598
	RUSBoost	**0.720**	**0.720**
Oversampling	GLM	0.727	0.729
	penalized GLM	0.726	0.729
	Decision Tree	**0.964**	**0.977**
	Random Forest	**0.981**	**0.997**
	MLP	0.829	828
	RUSBoost	0.727	0.727

The rest ML algorithms spent almost the same time on training.

In conclusion, after applying oversampling to data, Random Forest or Decision Tree is the best-performed model for the opioid misuse predication. Feature selection and PCA did not show great improvements on the models' prediction ability.

VII. FUTURE WORK

In this study, we showed the great potential of ML algorithms combing with oversampling in the imbalanced classification problem.

A few methods can be explored to build more valuable models for opioid misuse. For example,

- Use a more general data set with more features and employ feature selection.
- Employ the idea of anomaly or outlier detection such as One-Class Classification for Imbalanced Data [18].
- Develop a new algorithm with the help of a new statistical concept: data depth [19].

In addition, it is very challenging but also promising to build a classification model for more general imbalanced data.

VIII. DIFFICULTIES

In this project, the main difficulties come from the data preprocessing and model exploration.

The original data set contains more than 1700 features. We obtained our data set with a subset of all the features with the help of the similar work done by other researchers. However, there were not many references for opioid misuse prediction. Therefore, it is possible that a more effective feature selection method for data prepare. In addition, dealing with the missing values is a big concern. Features such as suicidal attempts have more than 50% missing (legitimately skipped). However, some other data points are Missing Completely at Random. Lots of exploration work has to be done before deciding how to handle the missing values properly.

The research of finding the appropriate models was tremendous. To explore the latest and most relevant modeling techniques, I reviewed a great many of journal articles. Instead of simply implementing those traditional and popular classification ML models, I was aiming to find new and promising models to predict opioid misuse. Furthermore, these new models can be well applied to more general imbalanced classification problems.

REFERENCES

[1] R. Longadge and S. Dongre, “Class imbalance problem in data mining review,” *arXiv preprint arXiv:1305.1707*, 2013.

[2] C. X. Ling and C. Li, “Data mining for direct marketing: Problems and solutions.” in *Kdd*, vol. 98, 1998, pp. 73–79.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[4] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, “Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary,” *Journal of artificial intelligence research*, vol. 61, pp. 863–905, 2018.

[5] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser, “Svms modeling for highly imbalanced classification,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 1, pp. 281–288, 2008.

[6] T. M. Khoshgoftaar, M. Golawala, and J. Van Hulse, “An empirical study of learning from imbalanced data using random forest,” in *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, vol. 2. IEEE, 2007, pp. 310–317.

[7] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, “Rusboost: A hybrid approach to alleviating class imbalance,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 1, pp. 185–197, 2009.

[8] G. King and L. Zeng, “Logistic regression in rare events data,” *Political analysis*, vol. 9, no. 2, pp. 137–163, 2001.

[9] ——, “Explaining rare events in international relations,” *International Organization*, vol. 55, no. 3, pp. 693–715, 2001.

[10] E. F. McCance-Katz, “The national survey on drug use and health: 2019,” *U.S. Department of Health and Human Services*.

[11] D.-H. Han, S. Lee, and D.-C. Seo, “Using machine learning to predict opioid misuse among us adolescents,” *Preventive medicine*, vol. 130, p. 105886, 2020.

[12] R. Mojtabai, M. Amin-Esmaeili, E. Nejat, and M. Olfson, “Misuse of prescribed opioids in the u nited s tates,” *Pharmacoepidemiology and drug safety*, vol. 28, no. 3, pp. 345–353, 2019.

[13] W.-H. Lo-Ciganic, J. L. Huang, H. H. Zhang, J. C. Weiss, Y. Wu, C. K. Kwoh, J. M. Donohue, G. Cochran, A. J. Gordon, D. C. Malone *et al.*, “Evaluation of machine-learning algorithms for predicting opioid overdose risk among medicare beneficiaries with opioid prescriptions,” *JAMA network open*, vol. 2, no. 3, pp. e190 968–e190 968, 2019.

[14] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, “An introduction to decision tree modeling,” *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 18, no. 6, pp. 275–285, 2004.

[15] M. Pal, “Random forest classifier for remote sensing classification,” *International journal of remote sensing*, vol. 26, no. 1, pp. 217–222, 2005.

[16] M. W. Gardner and S. Dorling, “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences,” *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[17] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, “Rusboost: Improving classification performance when training data is skewed,” in *2008 19th International Conference on Pattern Recognition*. IEEE, 2008, pp. 1–4.

[18] L. Zhuang and H. Dai, “Parameter optimization of kernel-based one-class classifier on imbalance learning,” *Journal of Computers*, vol. 1, no. 7, pp. 32–40, 2006.

[19] R. Jörnsten, “Clustering and classification based on the l1 data depth,” *Journal of Multivariate Analysis*, vol. 90, no. 1, pp. 67–89, 2004.

APPENDIX

The complete codes can be found **here** in my github.

A. Data Preprocess

```
# packages
library(tidyverse)
library(gridExtra)
library(knitr)
library(dplyr)
library(ggplot2)

NSDUH2019 <- read.csv(”NSDUH_2019.CSV”,header = T)
names(NSDUH2019) <- toupper(names(NSDUH2019))

# independent variables
fea <- c(”AGE2”, ”SEXIDENT”, ”IRMARIT”, ”WRKSTATWK2”, ”INCOME”, ”COUTYP4
            ”HEALTH”,”HRTCONDEV”,”DIABETEVR”, ”COPDEVER”,”CIRROSEVR”, ”HEPB
            ”HIVAIDSEV”,”CANCEREVR”,”HIGHBPEVR”,
            ”K6SCMON”,”ADDPREV”,”SUICPLAN”, ”SUICTHNK”,”SUICTRY”, ”TXEVRRCV
            ”ALCEVER”,”MJEVER”,”COCEVER”,”HEREVER”,”MEDICARE”,”CAIDCHIP”,”P
            ”ABODALC”,”ABODMRJ”,”ABODCOC”,”ABODHER”)

dat <- NSDUH2019[,fea] %>% filter(AGE2>6)
target <- NSDUH2019%>% filter(AGE2>6) %>% select(OPINMYR)

# missing values

for (i in c(2:4,7,10:20,22:40)){
  dat[,i] = ifelse(dat[,i]<10, dat[,i],NA)
}

missing.values <- dat %>%
  gather(key = ”key”, value = ”val”) %>%
  mutate(isna = is.na(val)) %>%
  group_by(key) %>%
  mutate(total = n()) %>%
  group_by(key, total, isna) %>%
  summarise(num.isna = n()) %>%
  mutate(pct = num.isna / total * 100)


levels <-
  (missing.values  %>% filter(isna == T) %>% arrange(desc(pct)))$key

percentage.plot <- missing.values %>%
  ggplot() +
  geom_bar(aes(x = reorder(key, desc(pct)),
               y = pct, fill=isna),
           stat = 'identity', alpha=0.8) +
  scale_x_discrete(limits = levels) +
  scale_fill_manual(name = ””,
                    values = c('steelblue', 'tomato3'), labels = c(”Pres
  coord_flip() +
  labs(title = ”Percentage␣of␣missing␣values”, x =
         'Variable', y = ”%␣of␣missing␣values”)

percentage.plot

# transformation: AGE2, other variables(too many missing values)

dat$AGE2_new = rep(NA, nrow(dat))
for (j in 1:nrow(dat)){
  if (dat$AGE2[j]<=12&dat$AGE2[j]>6){
    dat$AGE2_new[j] = ”18-25”
  }
  else if (dat$AGE2[j]<=14&dat$AGE2[j]>12){
    dat$AGE2_new[j] = ”25-35”
  }
  else if (dat$AGE2[j]<=16&dat$AGE2[j]>14){
    dat$AGE2_new[j] = ”35-65”
  }
  else{dat$AGE2_new[j] = ”>65”}
}

# suicidal related
fea2 <-  c( ”HRTCONDEV”,”DIABETEVR”, ”COPDEVER”,”CIRROSEVR”, ”HEPBCEVER”
            ”HIVAIDSEV”,”CANCEREVR”,”HIGHBPEVR”,”SUICPLAN”, ”SUICTHNK”,”SUI
            ”ALCEVER”,”MJEVER”,”COCEVER”,”HEREVER”,”MEDICARE”,”CAIDCHIP”,”P
            ”ABODALC”,”ABODMRJ”,”ABODCOC”,”ABODHER”, ”ADDPREV”, ”EDUSCHLGO”

for (k in fea2){
  dat[,k] <- ifelse(is.na(dat[,k]), 2, dat[,k])
  dat[,k] <- ifelse(dat[,k]==1,1,0)
}

dat_miss = na.omit(cbind(dat[,2:41], target))

for (i in c(1:19, 21:41)){
  dat_miss[,i] = as.factor(dat_miss[,i])
}
```

B. Oversampling

```
data = data.drop(data.columns[[0]], axis=1)  #
data[['K6SCMON']] = data[['K6SCMON']].astype('int64')
data[['K6SCMON']] = (data[['K6SCMON']]-data[['K6SCMON']].min())/
         (data[['K6SCMON']].max()- data[['K6SCMON']].min())
data['AGE2_new']]=pd.factorize(data[['AGE2_new']].values.
                 ravel())[0]

# Up-sampling (Oversampling) is the process of
#randomly duplicating observations from the minority class
from sklearn.utils import resample
# Separate majority and minority classes
data_major = data[data.OPINMYR==0]
data_minor = data[data.OPINMYR==1]
# Upsample minority class
data_minor_upsampled = resample(data_minor,
                                replace=True,
# sample with replacement
                                n_samples=39276,
# to match majority class
                                random_state=123)
df_upsampled = pd.concat([data_major, data_minor_upsampled])
y = df_upsampled.OPINMYR
X_balanced = df_upsampled.drop('OPINMYR', axis=1)
X = X_balanced
Y = y
```

C. Standard ML algorithms

```
## PYTHON
X = data.drop(['OPINMYR'],axis = 1)
Y = data[['OPINMYR']]

skf = StratifiedKFold(n_splits=10)
print("GLM")
# GLM
for train_index, test_index in skf.split(X, Y):
    X_train, X_test = X.loc[train_index], X.loc[test_index]
    Y_train, Y_test = Y.loc[train_index], Y.loc[test_index]
    model = LogisticRegression(penalty='none')
    model.fit(X_train, Y_train)
    y_prob = model.predict_proba(X_test)
    y_pred = (y_prob[:,1]>=0.135)
    fpr, tpr, thresholds = metrics.roc_curve(Y_test, y_pred, pos_label=1)
    print(metrics.auc(fpr, tpr))

print('penalized␣GLM')
for train_index, test_index in skf.split(X, Y):
    X_train, X_test = X.loc[train_index], X.loc[test_index]
    Y_train, Y_test = Y.loc[train_index], Y.loc[test_index]
    model = LogisticRegression(penalty = 'l1', solver = 'liblinear')
    model.fit(X_train, Y_train)
    y_prob = model.predict_proba(X_test)
    y_pred = (y_prob[:,1]>=0.135)
    fpr, tpr, thresholds = metrics.roc_curve(Y_test, y_pred, pos_label=1)
    print(metrics.auc(fpr, tpr))

print("Decision␣Trees")
# Decision Trees
for train_index, test_index in skf.split(X, Y):
    X_train, X_test = X.loc[train_index], X.loc[test_index]
    Y_train, Y_test = Y.loc[train_index], Y.loc[test_index]
    model = DecisionTreeClassifier()
    model.fit(X_train, Y_train)
    y_pred = model.predict(X_test)
    fpr, tpr, thresholds = metrics.roc_curve(Y_test, y_pred, pos_label=1)
    print(metrics.auc(fpr, tpr))


print("MLP")
# MLP
for train_index, test_index in skf.split(X, Y):
    X_train, X_test = X.loc[train_index], X.loc[test_index]
    Y_train, Y_test = Y.loc[train_index], Y.loc[test_index]
    model = MLPClassifier(random_state=1)
    model.fit(X_train, Y_train)
    y_pred = model.predict(X_test)
    fpr, tpr, thresholds = metrics.roc_curve(Y_test, y_pred, pos_label=1
    print(metrics.auc(fpr, tpr))


print("RUSboost")
# RUSboost
for train_index, test_index in skf.split(X, Y):
    X_train, X_test = X.loc[train_index], X.loc[test_index]
    Y_train, Y_test = Y.loc[train_index], Y.loc[test_index]
    model = RUSBoostClassifier(random_state=0)
    model.fit(X_train, Y_train)
    y_pred = model.predict(X_test)
    fpr, tpr, thresholds = metrics.roc_curve(Y_test, y_pred, pos_label=1
    print(metrics.auc(fpr, tpr))

print("Random␣Forests")
for train_index, test_index in skf.split(X, Y):
    X_train, X_test = X.loc[train_index], X.loc[test_index]
    Y_train, Y_test = Y.loc[train_index], Y.loc[test_index]
    model = RandomForestClassifier(random_state=0)
    model.fit(X_train, Y_train)
    y_pred = model.predict(X_test)
    fpr, tpr, thresholds = metrics.roc_curve(Y_test, y_pred, pos_label=1
    print(metrics.auc(fpr, tpr))


## R

# zelig
library(Zelig)
library(caret)
# Folds are created on the basis of target variable
folds <- createFolds(factor(dat_miss$OPINMYR), k = 10, list = FALSE)

# cross validation
library(pROC)
for (i in 1:10){
  test = dat_miss[folds==i,]
  train = dat_miss[folds!=i,]
  ml <- zelig(OPINMYR ~., model = "logit", data = train, cite = FALSE)
  pred <- predict(ml, test, type = "response")
  pred_test = ifelse(pred[[1]]>0.05,1,0)

  roc_obj <- roc(test$OPINMYR, pred_test)
  print(auc(roc_obj))
}
```