Classification Competition

William G. Hatcher, Kevin McNamara
Department of Computer and Information Sciences
Towson University, Maryland, USA 21252, USA
Emails: whatch2@students.towson.edu, kmcnamara@towson.edu

***Abstract*—Smart policing is in increasing demand as unrest grows around the world, social media allows for rapid organization and demonstration, and governments seek to reduce harm and increase safety overall. In the era of big data, the need for both citizens and law enforcement officers to have access to real-time analytics can help to drive public policy, resource allocation, and accountability. In this work, we assess eleven different machine learning algorithms, in analyzing the fatalities resulting from police shootings in the line of duty. We first assess the data and determine binning schemes, as well as clean the erroneous instances. We then conduct a thorough analysis of our algorithms by classifying eight of the original data attributes and two derived attributes. Our results show that, in general the advanced algorithms perform better, though only marginally so in most cases. We also demonstrate that our derived classes can increase the prediction Accuracy, Precision, Recall, and F1-Scores.**

***Index Terms*—Data mining, Machine learning, Classification**

I. Introduction

Data analytics are increasingly being applied across all manner of fields, technologies, and applications. Moreover, intelligent systems are being used to analyze data for the betterment of humanity and the protection of the public in general. For instance, self-driving vehicles have the potential to greatly reduce traffic accidents and alleviate traffic congestion. Similarly, advances in image segmentation and classification are improving disease and pathology detection. Furthermore, smart policing systems have the potential to identify and track suspects in crimes and aid in their resolution, while at the same time giving the general populace accountability for the actions taken by law enforcement.

Traditionally, technologies that can applied by law enforcement officials to have a direct impact on reducing crime and determining suspect motive and guilt have been met with widespread appeal and swift implementation. Examples include fingerprint and DNA analysis, general purpose CCTV cameras, and more recently body cameras. In addition, in the current era of big data, the ability to assess and cross-correlate policing data has significant potential for identifying suspects as well as providing meaningful policies and practices to address or reduce certain types of crime.

In this work, assess a dearth of data surrounding fatal shootings of civilians by police in the line of duty to consider the implications of various police-civilian interactions and look for meaningful ways to make predictions and reduce incidences where possible. The investigated dataset covers 2015 through 2017, and includes information concerning the mental health of the suspect, whether they were armed and what with, and what type of threat the officer perceived.

The remainder of this paper is as follows. In Section II, we provide a basic description of the datasets introduced. In Section III we outline the preprocessing steps taken. In Section IV we conduct an exploratory data analysis and make some preliminary results. In Section V, we conduct a regression analysis and develop a predictive model. Finally, in Section VI, we provide concluding remarks.

II. Data Description

The dataset utilized was compiled by the Washington Post, tracking approximately 13 details or features of every police shooting of a civilian that was fatal. The data was compiled from local news reports, social media, law enforcement websites, and independent databases. This dataset does not include people in police custody, fatal shootings by off-duty officers or non-shooting deaths.

The data includes some 2,143 instances with attributes, which include: name, date, manner of death, armed, age, gender, race, city, state, signs of mental illness, threat level, flee, and body camera. In particular, the *armed* attribute indicates whether the suspect was armed and with what; *manner of death* indicates whether the suspect was shot or shot and tasered; the data lists whether or not *signs of mental illness* were perceived in the suspsect during the encounter; *threat level* indicates the percieved threat of violence by the acting officer; the attribute *flee* indicates whether the suspect fled on foot, by car, or did not flee; and *body camera* indicates whether the incident was captured on body camera footage.

The vast majority of the attributes comprise categorical sets of string values, including attributes *manner of death* (shot, shot and Tasered), *gender* (M, F), *race* (A - Asian, B - Black, H - Hispanic, N - Native American, O - Other, W - White), *city* (limited to all cities within the US), *state* (51 possible values including the District of Columbia), *threat level* (attack, other, undetermined), and *flee* (foot, car, not fleeing, other). Two attributes, signs of mental illness and body camera, are binary boolean values (True, False). *Name* is a unique-valued string, while *armed* could be considered either a categorical set or unique-valued strings (63 possible values that may overlap). Finally, *date* and *age* are fixed numerical values, date formatted and integer respectively.

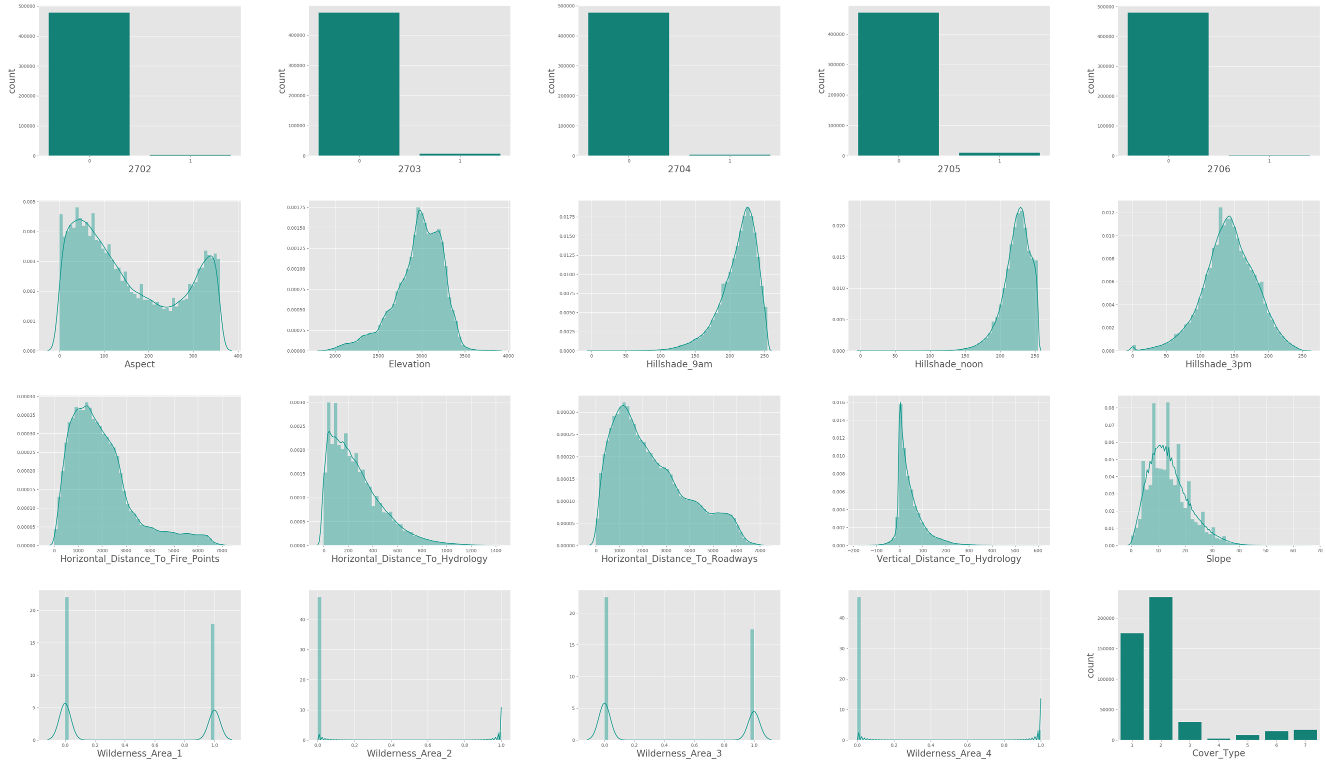

Fig. 1. Bar charts, histograms, and KDE plots (Top to Bottom, Left to Right) of manner of death, gender, race, flee, body camera, signs of mental illness, state, age, armed categories 1, and armed categories 2 attributes.

III. Data Preprocessing

The state of the unprocessed dataset is poor. The dataset includes many empty values, especially in the *armed*, *gender*, *race*, *age*, and *flee* attributes. Thus, data preprocessing is needed to clean the data and decide whether to replace values, and if so, what to replace them with. In addition, given the interrelation of some data items, such as city and state, it may be necessary to split or combine data items for a clarity. This can help to control for outliers and anomalies.

A. Data Cleaning

As noted above, most of the data items are missing values, and no erroneous data, such as misspellings or unique cases exist. To clean the dataset, we first assess the degree of missing data for each attribute. In this case, *armed* is missing 6 datapoints, *gender* is missing 1 datapoint, *race* is missing 103 datapoints, *age* is missing 43 datapoints, and *flee* is missing 36 datapoints. We began by removing the six missing from *armed*, as there exist unarmed and undetermined categories, along with the one missing gender. Note that no non-binary gender has been expressed. We also removed all instances with missing age and missing flee attributes as well, bringing the total number of attributes to 2,063. Finally, removing the remaining instances missing race attributes, our final dataset includes a total of 1,987 instances. In this case, we resolved to remove the erroneous data points based on the lack of domain knowledge and clear ability to make reasonable assumptions.

B. Data Transformation

Due to the categorical nature of the majority of attributes, we have binned the various attributes into finite integer sets. These include *manner of death*, *armed*, *gender*, *race*, *state*, *threat level*, and *flee*. In addition, because the *armed* attribute includes some 63 unique instances, we have made two additional binning groups *armed categories 1* and *armed categories 2*. In particular, *armed categories 1* condenses the set of categories down to 10, these being: Small - 0, Medium - 1, Large - 2, Projectile - 3, Two Weapons - 4, One Weapon One Projectile - 5, Vehicle - 6, Unarmed - 7, Unknown weapon - 8, Undetermined - 9, and Toy Weapon - 10. Note that the sizes Small, Medium, and Large indicate the relative size of a hand weapon that is not a projectile weapon, and that projectile weapons include any gun, nail gun, crossbow, etc. that fires some object. In *armed categories 2*, we further reduce this set to Weapon - 0, Gun - 1, Vehicle - 2, Unarmed - 3, Unknown weapon - 4, Undetermined - 5, and Toy Weapon - 6. In this case, Weapon indicates any non-projectile weapon or combination of weapons, while Gun indicates any projectile weapon or combination of projectile and non-projectile weapons.

TABLE I. Basic Learning - Accuracy

ACCURACY	**Full Set 1**	**Full Set 2**	**Binary Only 1**	**Numeric Only 1**	**Numeric Only 2**
KNN-3	0.596	0.596	0.415	0.596	0.596
DT-gini	0.927	0.927	0.540	0.897	0.897
DT-entropy	**0.930**	**0.930**	0.540	0.910	0.909
RF-10	0.925	0.927	**0.546**	**0.918**	**0.917**
NB-gaussian	0.634	0.634	0.488	0.634	0.634

TABLE II. Basic Learning - F1-Score

ACCURACY	**Full Set 1**	**Full Set 2**	**Binary Only 1**	**Numeric Only 1**	**Numeric Only 2**
KNN-3	0.590	0.590	0.400	0.590	0.590
DT-gini	**0.930**	**0.930**	0.540	0.900	0.900
DT-entropy	**0.930**	**0.930**	0.540	0.910	0.910
RF-10	0.920	**0.930**	**0.550**	**0.920**	**0.920**
NB-gaussian	0.640	0.640	0.320	0.640	0.640

TABLE III. Improved Basic Learning - Accuracy

ACCURACY	**Full Set 1**	**Full Set 2**	**Full Set 1 min-max**	**Full Set 2 min-max**
KNN-7	0.596	0.596	0.875	0.875
DT-entropy, min-leaf-5	0.921	0.921	0.921	0.921
ET-entropy	0.825	0.816	0.841	0.822
RF-100	**0.942**	**0.942**	**0.943**	**0.943**
NB-multinomial	-	-	0.642	0.642

TABLE IV. Improved Basic Learning - F1-Score

ACCURACY	**Full Set 1**	**Full Set 2**	**Full Set 1 min-max**	**Full Set 2 min-max**
KNN-7	0.590	0.590	0.870	0.870
DT-entropy, min-leaf-5	0.920	0.920	0.920	0.920
ET-entropy	0.820	0.820	0.840	0.820
RF-100	**0.940**	**0.940**	**0.94**0	**0.940**
NB-multinomial	-	-	0.620	0.620

IV. Exploratory Data Analysis

With the exception of the *armed* attribute and the derived *armed categories 1* attributes, we can see the majority of the attributes in Fig. 1. Note that the x-axis labels were removed for the state attribute for clarity. Also, as the number categories of the *armed* attribute natively are 63, we show only the reduced *armed categories 2* derived attribute instead. Also notice that the only attribute to resemble a normal distribution is *age*. We also note that there are no direct positive or negative relationships observed from which to recommend a particular fitting method of any given pair of attributes. Indeed, sorting the data by each attribute shows not direct correlations. We can see that many attributes are highly biased, which may adversely affect the classification results. In addition, in the classification evaluation to follow, we will normalize our binned attributes to determine if they have any impact on the learning mechanisms or the classification results.

V. Classification Analysis

We now carry out an evaluation of multiple classification algorithms as applied to our fatal police shooting data. We first introduce the primary sampling methodology used, develop the metrics for evaluation, describe the algorithms evaluated, and then provide the evaluation results.

A. Sampling Method

In sampling the fatal police shooting dataset, we apply the holdout method, separating our data into approximately 70% training and 30% testing sets. Moreover, we select the data at random. Ideally, this should be carried out multiple times and cross-validated. Otherwise, we may find that the particular split affects our learning mechanisms adversely, providing skewed results not representative of the data overall.

B. Metrics

The primary metrics we will use to assess the results of our classification are Accuracy, Precision, Recall, and F1-Score. In classifying the test data once trained, the resulting classifications fall into one of four categories: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). In more detail, a prediction is labeled as TP if the correct positive class was assigned correctly as positive. Likewise a prediction is labeled as FP if the predicted class was the positive class, but the ground truth was the negative class. So if the positive class is 1, then predicting 1 when the answer was 0 is an FP. Note that this becomes more difficult in multivariate systems. In this case, we look at each category in a class alone as the positive class. So for values 0, 1, and 2, we extract TP, FP, TN, FN, Precision, Recall and F1-Score values three times, setting 0, 1, and 2 each as the positive class and the other two as the negative. Here, if 0 is positive, then 1 and 2 are negative, and FP includes both 1 and 2 ground truth values predicted as 0. Thus, we derive 3 Precision, Recall, and F1-Score values and take the average of each. Also note that the overall Accuracy remains the same.

Extending from our definitions of TP, FP, TN and FN, we define Accuracy, Precision, Recall, and F1-Score as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}. \tag{1}$$

$$\text{Precision} = \frac{TP}{FP + TP}. \tag{2}$$

$$\text{Recall} = \frac{TP}{FN + TP}. \tag{3}$$

$$F1 = \frac{2 * (\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}}, \tag{4}$$

Notice that Fi-Score is defined as the harmonic average of precision and recall.

TABLE V. Advanced Learning - Accuracy

ACCURACY	Full Set 1	Full Set 2	Full Set 1 min-max	Full Set 2 min-max
LogReg	0.717	0.647	**0.660**	0.665
NN-30x3-sgd	**0.851**	**0.652**	0.487	**0.804**
NN-30x3-lbfgs	0.810	0.651	0.481	0.744
SVM-rbf	0.790	0.651	-	0.770
Bagging	0.113	0.126	-	0.655
Boosting	0.418	0.613	-	0.613

TABLE VI. Advanced Learning - F1-Score

ACCURACY	Full Set	Binary Only	Numeric Only	Numeric Only min-max
LogReg	0.700	0.630	**0.630**	0.640
NN-30x3-sgd	**0.850**	**0.640**	0.240	**0.800**
NN-30x3-lbfgs	0.810	0.640	0.370	0.740
SVM-rbf	0.810	0.640	-	0.740
Bagging	0.660	0.740	-	0.640
Boosting	0.580	0.620	-	0.610

TABLE VII. Top Basic Learning - Random Forests

FULL DATA min-max	Accuracy	Precision	Recall	F1-Score
RF-250	0.943	0.94	0.94	0.94
RF-500	0.943	0.94	0.94	0.94
RF-1000	0.943	0.94	0.94	0.94
RF-100-entropy	0.945	**0.95**	**0.95**	0.94
RF-250-entropy	**0.951**	**0.95**	**0.95**	**0.95**

TABLE VIII. Top Advanced Learning - Dense Neural Networks with TensorFlow

FULL DATA min-max	Accuracy*	Precision	Recall	F1-Score
250-100-10-7_1-64	0.937*	0.78	0.77	0.76
250-100-10-7_1-64-bin	0.899*	0.62	0.63	0.61
250-100-10-7_1-64-mult	0.917*	0.69	0.7	0.69
250-100-10-7_32-16	**0.968***	**0.90**	**0.88**	**0.89**
500-250-100-10-7_32-16	0.958*	0.84	0.85	0.84

C. Algorithms

In this evaluation, we compare five Basic and six Advanced learning algorithms. What's more, we compare these across ten binned attributes (*age*, *armed*, *body camera*, *flee*, *gender*, *manner of death*, *signs of mental illness*, *race*, *state*, and *threat level*) as well as two of our derived attributes (*armed categories 1* and *armed categories 2*). The Basic algorithms are K-Nearest Neighbors (KNN-3), two Decision Trees (DT-CART and DT-entropy), Random Forest (RF-10), and Naive Bayes (NB-gaussian). The Advanced learning algorithms are Logistic Regression (LogReg), two Multilayer Perceptrons (NN-30x3-sgd and NN-30x3-lbfgs), SVM (SVM-rbf), Bagging and Boosting.

Regarding the Basic algorithms, we apply the K-Nearest Neighbors algorithm for classification, setting the initial number of clusters to 3 and using the euclidean distance to segregate clusters. Also, we utilize two examples of Decision Trees that use the CART algorithm to build the binary tree. The first uses the gini index to split the data and second uses entropy instead. For the Random Forest classifier, this operates by implementing multiple decision trees and taking the arithmetic Mode of all of them. In this case, we set the initial number of trees to 10. Finally, we additionally use the probabilistic Naive Bayes classifier.

Concerning the Advanced algorithms, we use the Logistic Regression with a inverse regularization strength parameter set to 100,000. We also consider two neural networks of the same shape (i.e., three layers of 30 neurons each). The two are separated by the solver parameter, otherwise known as the optimization algorithms, in this case stochastic gradient descent and limited memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS). In this case, both optimizers attempt to minimize the loss function by following the steepest descent of the gradient of the target function. Additionally, we apply the Support Vector Machine (SVM) with the Radial basis function (rbf) kernel, which is used to map the inputs into higher dimensions to determine a gap as large as possible between class values. Finally, we use two Ensemble learning methods: Bagging and Boosting. In Bagging, we subdivide the original dataset into multiple subsets, and each is supplied to a learning (in this case the same Naive Bayes algorithm above). The multiple parallel learning algorithms provide predictions and the class with the highest mean probability is selected. Here we use the default value of 10 estimators. In Boosting, the learning estimator is applied to the original data, and weight are then adjusted based on the incorrectly classified instances repetitively. Here we apply 100 estimators of the AdaBoost-SAMME algorithm.

D. Evaluation

We now consider the results of applying the five Basic and six Advanced learning algorithms to classify our data. Note that we have classified eight of the original classes and two derived classes, as denoted above. The results of our evaluation can be seen in Tables I-IV. First, we note that all of the algorithms struggled to classify the *age* attribute, which is to be expected, as this attribute is actually a continuous integer variable better suited to regression analysis. Indeed the accuracy scores for *age* were less than 5%. Second, focusing on top performers, we see that the Accuracy scores, Precision, Recall, and F1-Scores were all quite significant in the binary classes of *body camera*, *gender*, and *manner of death*. We should note that these classes are very highly skewed, so this result is also not a surprise. Indeed, it may be possible to attain such accuracy by simply predicting only the dominant class.

Third, looking simply at Accuracy, we see that the advanced algorithms generally performed better, though in some cases only marginally so. Indeed, in many instances, the differences between the top performing advanced algorithms and their basic counterparts are only a couple of percent. The best performer across all attributes appears to be the SVM algorithm. Fourth, considering Precision and Recall, we see that Naive Bayes had the best Precision scores of the basic algorithms, while Boosting won for Precision in the advanced algorithms. In contrast, neither was nearly so successful in terms of Recall. In fact, Naive Bayes was often the worst out of all algorithms in terms of Recall. In the case of the advanced algorithms, Logistic Regression and the first Neural Network were the top performers. Considering that Precision tells us what proportion of positive identifications were actually correct, while Recall telling us what proportion of actual positives were identified correctly, we can conjecture about the nature of Naive Bayes. Indeed, it appears to incur quite a high FN rate, while maintaining a significantly low FP rate.

Finally, considering the F1-Scores, which take the harmonic mean of Precision and Recall, we get an estimation of how well the algorithms perform overall. Those with the highest F1 generally have well balanced Precision and Recall. From our results in Table IV, we can see that Boosting performed the best in this category.

VI. Conclusion

Properly assessing law enforcement interactions with the public can provide significant public policy decisions and can lead to more safe and congenial environment. In this work, we have assessed eleven machine learning algorithms for classification in a dataset of police-involved shooting fatalities. The results demonstrate, in some cases, a high level of accuracy and strong predictive qualities, especially on binary data, such as gender, body cameras, and manner of death. Moreover, we have considered in detail the abilities of each algorithm applied, overall observing that the advanced algorithms indeed perform better. However, this performance is not typically significantly better, only marginally so. In future work, it would be necessary to expand the assessment to improve the performance of each algorithm through parameter tuning.