Machine Learning Approach towards the Police Shooting Dataset

Prepared By

Fanyu Zuo
Pinky Sindhu
COMP 379
Machine Learning
Dr. Dmitriy Dligach
12/15/2017

Table of Contents

Introduction **3**

Dataset Description **3**

Baseline Approach Description **6**

Method description **6**

Naive Bayes 7

Logistic Regression 8

Random Forest 8

Decision Tree 9

SVM 9

Evaluation **9**

Discussion **10**

Conclusion **11**

Appendix **11**

Introduction

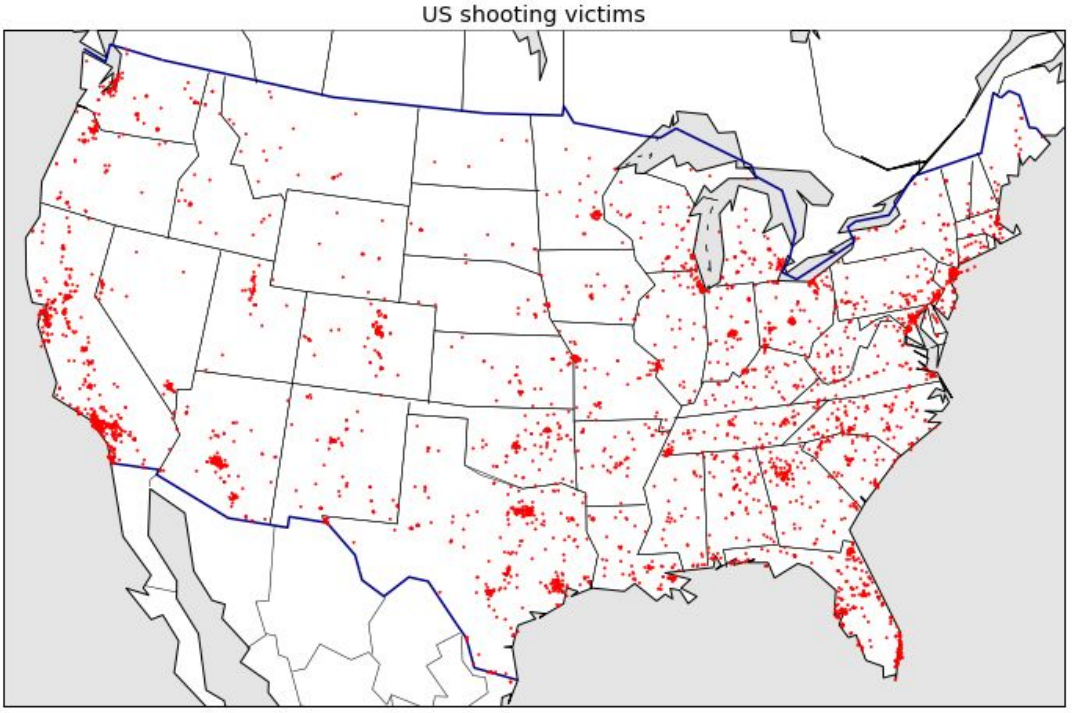
The project is intended to run binary classification algorithms on the police shooting datasets. The dataset contains the data of the victims of police shootings. It described 3918 tuples of victims involved in police shooting incidents, and there are 47 attributes describing the victims including name, age, race, mental health, education, armed and the incidents description (text) from police as well as other related attributes. We set our main direction of machine learning analysis on discovering the facts of victims' mental health conditions on the police shooting incidents with consideration of the other relative attributes.

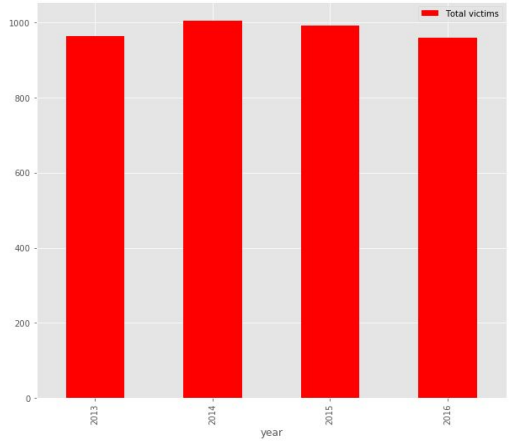
It is found that the majority of the victims were armed and thus results in police shooting. The initial analysis of the data suggests that the some of the victims where suffering from mental illness and hence showed the violent tendencies. Text Based and Numerical based binary classifications are designed, modelled and tuned for the dataset to predict the mental illness of the victims. Classification accuracy and F1 scores are applied as performance evaluation metric, and they are calculated after fine tuning the model. We also compare a handful of different classifiers to identify the appropriate classification model for the dataset.

Dataset Description

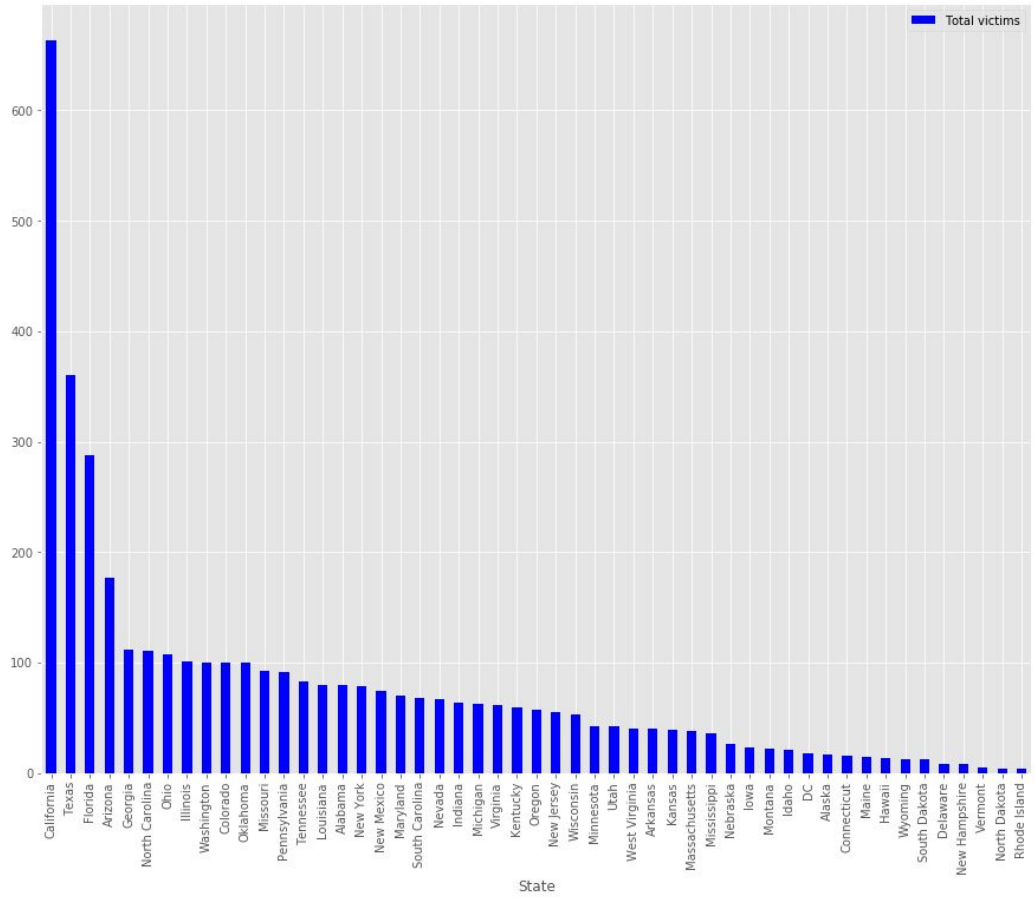
The dataset used for classification contains the data of victims in Policy Shooting from 2013 - 2016 across the country. Dataset contains only cleaned data and missing values are mostly excluded for the classification purpose.The shooting data is from the **Washington Post** and the **Mapping Police Violence** database. It is combined with data from various **US Census** and the **FBI Uniform Crime** Reporting database.

The below visualizations are generated from running few statistics of the dataset in Python, which explains the nature and content of the dataset. As you can see from the map visualization and histogram, the fatal force shooting incidents happen in California, Texas, Florida, and Arizona more frequently since you can see the red dots are denser along those regions.





The dataset contains shooting data from the years 2013 - 2016 and the image shows the range of data over the years.



The values are from across the country, though majority of the tuples are from the top 6 states including California, Texas, Florida, and Arizona, which as can be seen in the above histogram.

A correlation is run against the gender attribute and race attribute in the datasets and males are mostly the victims of the shooting compared to females. Also we can tell from the statistics that Black(or African Amercian) and White people are the largest population of the victims compared to other races, which is understandable due to they are the major population of US.

Race/ Gender	Asian	Black or African American	Hispanic and Latino Americans	Native American	Other Race	White
Female	2	37	21	22	4	106
Male	58	969	636	257	44	1677

Baseline Approach Description

The dataset contains numerical and textual attributes which can attribute to the police shooting. One of the reasons behind the violence can be considered as the mental illness or depression of the victims. The textual attribute of the dataset gives a brief description of the incident and the other numerical attributes give away the details of the victims, incident surroundings and environment in which the victim lives such as education rate in local areas. The project is intended to identify an accurate classification method for these type of datasets. To classify the victims suffering from mental illness or not, two approaches are used. Classification build on text attributes and classification build on other numerical attributes of the victim. The model with more accuracy is finally opted for the dataset.

Method description

Preparing the Data : The label attribute chose for the classification is mental illness, which is a boolean attribute. Therefore a binary classifiers are used to build the model. The dataset is cleaned by dropping empty values. Attributes irrelevant to shooting or victims, such as FIPS, name, state, geocode_exact, longitude, latitude, year, and date of the accidents, were also dropped before building the model. LabelEncoder is applied to convert categorical/boolean data into numerical integers, such as race, gender, and armed.

Selecting the Classifiers : Naive bayes and Logistic Regression models are selected for building model with textual attributes. Random Forest, Decision Tree and SVM are selected as classifiers for building the models with attributes which are numerical.

Training and testing the Models : We split the dataset into 70% training, 15% developing and 15% testing data. First training the model on 70% training dataset and then tuning the model on 15% developing dataset, once we are satisfied with the performance on developing dataset, we test the model on test dataset and make the report based on test data performance.

Tuning the Model : Each classifier models are tuned uniquely. The *Naive Bayes* classifiers are tuned with cleaning data and removing the stop words. The words with low frequency count are removed from the count. The *Logistic Regression* classifier with choosing an appropriate regularization between l1 and l2. Also when the words are split into token for generating the vecore model the stop words are removed too. For *SVM* classification the gamma and cost values are adjusted for fine accuracy. By varying the min_samples_split *Decision Tree* classifier is fitted accurately. *Random Forest* makes use of the n_estimators which allows us to fine tune the model.

Calculating the F1 Score : The reason we select F1 score as one of the performance metric is that the dataset on attribute Mental illness is an imbalanced dataset, the ratio being about 1:3. Since the accuracy might show bias on imbalanced data while the F1 score conveys the balance between the precision and the recall, we set F1 as the other metric as the complementary evaluation method.

Brief Explanation of the Classifiers

Naive Bayes

We implement our own version of Naive Bayes algorithm to classify if a victim is identified as mentally ill or not when the shooting incident happened. First we define a preprocessing method in the classifier to strip all strings from both negative and positive text files, then set random seed and shuffle the reviews with same random seed separately. After splitting for train_dev_test, we get 6 subsets. Secondly, creating a dictionary from two positive and negative input datasets and count each word frequency, using the word as the keys and the number of frequency as values. Nltk lib is used for stopword filtering. Also word from dictionary with frequency less than 2 will also be filtered out by a filter rule. Thirdly, calculate the prior by (number_of_class_documents/total number of documents). Conditional probabilities are calculated by equation (count_of_word_in_a_class/(volcabulary_count+total_word_count_of_a_class)) , after smoothing by 1, calculate the product as the probability of each word, and make the predictions of each document. After implementing the model to

developing dataset, final step is to evaluate the model by calculating a confusion matrix and get the accuracy and F1 score from the matrix.
With filtering out stop words:
Model Accuracy : *88.62%*
F1 Score : *0.94*

Without filtering out stop words:
Model Accuracy : *86.90%*
F1 Score : *0.93*

Logistic Regression

The logistic regression model is is to classify the description of the incident which can be attributes to the mental state of the victim. Once the data is classified into training and testing datasets, we used a GridSearchCV object to find the optimal set of parameters for the regression model using 5-fold stratified cross validation. This textual classification is done by the technique names as term frequency-inverse document frequency. Scikit-learn in Python implements yet another transformer, the TfidfTransformer that takes the raw term frequencies from CountVectorizer as input and transforms them into tf-idf(term frequency-inverse document frequency) vectors. Documents are tokenized to preprocess the data to clean it and stop words are also removed. After tuning the model accuracy is calculated and the F1 score of the classifier is evaluated.
Model Accuracy : *89.70%*
Model Accuracy Before Tuning : *88.40%*
F1 Score : *0.84*

Random Forest

Random forest is an ensemble method to combine weaker learner decision trees to a strong and robust learner. Implemented from *sklearn.ensamble.RandomForestClassifier* it draw a random bootstrap(with replacement) sample to grow a decision tree, and repeat the procedure to grow 1000 trees, finally aggregate the predictions from the 1000 trees and assign the class label by majority votes.

Model Accuracy : *69.6%*
F1 Score : *0.54*

Decision Tree

Decision Trees are a type of Supervised Machine Learning where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split. Binary classification tree algorithm is used for classifying the dataset. The trained model is run with the test dataset after tuning which generates an accuracy around 65% and F1 score is also evaluated to validate the model.

Model Accuracy : *65.20%*
F1 Score : *0.50*

SVM

SVM classifies samples by maximize the margin. The margin is defined as the distance between the separating hyperplane and the support vectors. The trained model is using the default rbf kernel method, and setting kernel coefficient for gamma=0.001, the regularization Penalty parameter C=100 of the error term. Larger c values corresponds to larger error penalty while smaller c corresponds to smaller penalty.

Model Accuracy : *67.40*
F1 Score : *0.52*

Evaluation

The hold out method is applied for model performance evaluation. For each model, the dataset is split into three datasets: Train, Dev and Test. The Supervised Classifier models are trained on Train Dataset and the hyperparameters of models are tuned with the Dev Dataset. Finally the models are evaluated and accuracy is calculated on Test Dataset. The split of the dataset is done in the ratio 70:15:15.

The data split is stratified to which returns stratified randomized folds. The folds are made by preserving the percentage of samples for each class. Due to the imbalance in the data 5-fold stratified cross-validation is done with one of the classifier models(Logistic regression).

The performance of the model is evaluated more using F1 score than the classification accuracy. The precision and recall values are evaluated for each model and F1 score is calculated from the same.

Discussion

Model	Accuracy	F1 Score
Random Forest	69.6%	0.54
Decision Tree	65.2%	0.50
SVM	67.4%	0.52
Naive Bayes	88.62%	0.94
Logistic Regression	89.70%	0.84

The above table lists the accuracy and Performance measure F1 value for the different type of classifiers used. The dataset is highly imbalanced towards the label attribute mental illness. With almost 70% of the dataset are truly classified as mental illness = false. Hence from the above classifiers we can say that Naive Bayes and Logistic regression performs quite well. Hence the classification accuracy above 70% can be considered positive and used as a good classification model. Moreover the F1 score is pretty good for these two classifiers. However SVM and Decision Tree performed badly against the label attribute and it will be inappropriate for this type of dataset.

Conclusion

The textual Classification by Naive bayes yields good accuracy and high performance score of 0.94 as F1 value. It can be concluded that Naive Bayes is selected as the desired classifying model.

Appendix

Fanyu Zuo	Classification, Fine Tuning, Documentation
Pinky Sindhu	Classification, Fine Tuning, Documentation