Machine Learning Engineer Nanodegree

Capstone Proposal

Patrick Poon April 18, 2018

Proposal

Domain Background

> Most people...the interaction that they're going to have with a police officer is because [...] they're stopped for speeding. Or, forgetting to turn their blinker off.[1]
>
> -- Cheryl Phillips, Journalism Professor at Stanford University

On a typical day in the United States, police officers make more than 50,000 traffic stops.[2] In recent years, there have been numerous incidents that have made national headlines that involve an officer shooting and, in some cases, killing the driver or an occupant. Many cite racial biases against Blacks and Hispanics for the disproportionate amount of such incidents for these communities. Here are some relevant articles:

- Was the Sandra Bland traffic stop legal -- and fair? (https://www.cnn.com/2015/07/23/opinions/cevallos-sandra-bland-traffic-stop/index.html)
- Philando Castile shooting: Dashcam video shows rapid event (https://www.cnn.com/2017/06/20/us/philando-castile-shooting-dashcam/index.html)

This Capstone project will not attempt to prove or disprove this controversial topic, and will attempt to avoid making any controversial statements on either side of the conversation.

Problem Statement

Instead, this project aims to create a multiclass classifier that takes various discrete traffic stop situational values to predict the outcome of a traffic stop, specifically in the state of Connecticut. Given a driver's age, gender, race, violation, and the county where the traffic stop occurs, can we reliably predict whether the traffic stop will result in a verbal/written warning, a ticket, a summons to appear in court, or an arrest?

Datasets and Inputs

The dataset I will be using comes from the The Stanford Open Policing Project (SOPP), which gathers, analyzes, and releases records from millions of traffic stops by law enforcement agencies across the country. The organization aims to help researchers, journalists, and policymakers investigate and improve interactions between police and the public.

For this project, I will be using a small subset of SOPP's data collections[3], specifically for the state of Connecticut [4]. Most American states have their own policies and practices for collecting traffic stop data, so there is no standard policy common across all states. In fact, SOPP has collected data for only 31 out of 50 states, that consumes 21G of disk space at the time of this project. Analyzing each of the states' data, Connecticut had the cleanest and most consistent set of records compared to the others. It has a total of 318,669 records of traffic stops made between 2013 to 2015 with the following fields:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 318669 entries, 0 to 318668
Data columns (total 24 columns):
id                      318669 non-null object
state                   318669 non-null object
```

```
stop_date               318669 non-null object
stop_time               318447 non-null object
location_raw            318628 non-null object
county_name             318627 non-null object
county_fips             318627 non-null float64
fine_grained_location   317006 non-null object
police_department       318669 non-null object
driver_gender           318669 non-null object
driver_age_raw          318669 non-null int64
driver_age              318395 non-null float64
driver_race_raw         318669 non-null object
driver_race             318669 non-null object
violation_raw           318669 non-null object
violation               318669 non-null object
search_conducted        318669 non-null bool
search_type_raw         4846 non-null object
search_type             4846 non-null object
contraband_found        318669 non-null bool
stop_outcome            313313 non-null object
is_arrested             313313 non-null object
officer_id              318669 non-null object
stop_duration           318669 non-null object
dtypes: bool(2), float64(2), int64(1), object(19)
memory usage: 54.1+ MB
```

318,669 records should be sufficient for training and testing purposes. I will extract the `stop_outcome` column to use as output labels and ground truth.

Solution Statement

To create a multiclass classifier to predict the outcome of a traffic stop, I plan to evaluate and experiment with various supervised learning models that are available in the scikit-learn Python library. Under consideration are the following:

- Gaussian Naive Bayes (GaussianNB)
- Decision Trees
- Ensemble Methods (Bagging, AdaBoost, Random Forest, Gradient Boosting)
- K-Nearest Neighbors (KNeighbors)
- Stochastic Gradient Descent Classifier (SGDC)
- Support Vector Machines (SVM)
- Logistic Regression

Benchmark Model

As far as I know, there is no external benchmark to compare the results to, so I propose generating a naive predictor to set the benchmark. The most common `stop_outcome` value in this dataset is 'Ticket' which comprises 70% of all stop outcomes as described in the following table:

Outcome	Count	%
Arrest	7,312	2.33%
Summons	12,205	3.90%
Ticket	218,973	69.89%
Verbal Warning	47,753	15.24%
Written Warning	27,070	8.64%
	313,313	

As a base model without any intelligence, predicting every traffic stop will result in a 'Ticket' will generate an accuracy score of around 0.70 and serve as our benchmark model.

Evaluation Metrics

Initially, I had planned on proposing to use the F-beta score for my evaluation metric, but after some tests, I encountered the following error when attempting to do so:

```
Error: Sample-based precision, recall, fscore is not meaningful outside multilabel classification. See the accuracy_scc
```

Instead, I shall use **accuracy classification score** as my evaluation metric. According to the sklearn page for the `accuracy_score` function[5], in the context of multiclass classification, the function is equivalent to the `jaccard_similarity_score` which calculates the Jaccard index, also known as "Intersection over Union," as illustrated in the following formula:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

Project Design

To start, I will **explore the data**, and see if I can extract any insights or build some intuition about it. I will graph different columns to determine which may be unbalanced.

Next, I will **prepare the data** by performing a number of operations, such as the following:

1. **Remove rows that have no `stop_outcome` value**: Since my main objective is to predict the outcome of a traffic stop, null values for this field render the associated record unusable. It would not make sense to attempt to fill the empty values with a median or average value.
2. **Remove empty columns**: There are a few columns that have no values at all, so it makes sense to drop these entirely.
3. **Handle columns with some missing values**: Some may need to be dropped, while others will be filled in with median or average values as appropriate.
4. **Recategorize very granular values to broader categorical values**: For instance, day values are very specific, and may not lend themselves to provide much insight. However, they may be able to provide some insight if they were categorized by annual seasons, like Spring, Summer, Fall, and Winter. Similarly, the `stop_time` values are very specific, and would benefit from being converted to time of day, like morning, afternoon, evening, and small hours.
5. **Convert binary values to boolean values**: The `driver_gender` column has values of **M** and **F**. It would be better to convert these to boolean and rename the column to `is_male`.
6. **Clean up messy column data**: The `violation_raw` and `violation` columns seemed to have repetitive and inconsistent data entry issues. For example, two different violations can be phrased differently yet mean the same thing. I will need to settle on a standard value for these duplicate values. I will also need to manually perform one-hot encoding for each class value in the hopes that violations can provide predictive power to my classifier.
7. **Normalize numerical data**: For this project, the only column that might be considered numerical is `driver_age`, even though it is more characteristic of a discrete value.
8. **Extract `stop_outcome` column for classification labels and ground truth**: These are the output values for the predictions.

After these operations are performed, I will **preprocess the data**, by one-hot encoding columns that have stable categorical values. Once completed, I will move on to **shuffling and splitting the data** into training and testing sets, which I will use as inputs to **evaluate 3-5 supervised learning models** as I specified in the **Datasets and Inputs** section above.

Finally, I will choose the **best performing model** and **tune its hyperparameters** by doing a grid search, then determine whether certain features can be dropped by performing feature selection, which involves determining which features have the

highest prediction power.

References

[1] Cheryl Phillips, Journalism Professor at Stanford University, interview. Stanford Open Policing Project (July 17, 2007). Retrieved from https://youtu.be/iwOWcuFjNfw?t=4s.

[2] Stanford Open Policing Project (https://openpolicing.stanford.edu/)

[3] Stanford Open Policing Project: About the Data (https://openpolicing.stanford.edu/data/)

[4] Stanford Open Policing Project: Data Download for Connecticut (https://stacks.stanford.edu/file/druid:py883nd2578/CT-clean.csv.gz)

[5] sklearn.metrics.accuracy_score (http://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html)