police-eis Documentation

Release 0.0.1

Jennifer Helsby

February 14, 2017

police-eis is a data-driven Early Intervention System (EIS) for police departments. The system uses a police department's data to predict which officers are likely to have an adverse interaction with the public. An adverse incident can be defined on a department by department basis, but typically includes unjustified uses of force, officer injuries, preventable accidents and sustained complaints. This is done such that additional training, counseling and other resources can be provided to the officer *before* any adverse interactions occur.

This is a project of the University of Chicago's Center for Data Science and Public Policy.

CHAPTER 1

Contents

1.1 Introduction

This document describes the main functions of the software *without* going into detail into the technical details of how to set up the software. Please refer to Quickstart for that information.

1.1.1 Predicting Adverse Incidents

The goal of the system is to provide a data-driven way for police departments to monitor their officers.

The nomenclature that we will use is that of *adverse incidents*: these are the incidents that both the police department and the population that is policed would like to minimize. Typically this would include events like sustained complaints and unjustified uses of force, though the specific definition that is most useful depends on the police department. For example, predicting sustained complaints is predicted on a robust internal affairs process to determine whether of not complaints are sustained as well as a robust complaint process.

1.1.2 Individual Level Prediction

The individual level prediction predicts on an officer by officer basis the risk that the officer will have an adverse incident in the next time period. Individual feature weights are produced for each officer, allowing an analyst to understand why an officer has been assigned a given risk score. In order to dynamically determine from a police department which behaviors are predictive of future adverse incidents, we build many *features*. These features are behaviors that we believe may be predictive of going on to have an adverse incident in the future.

Individual feature importances for officers with high risk scores (for example data):

1.1.3 Group Level Aggregation

Group-level aggregation enables an analyst to examine the average risk of individual divisions and units. The evaluation webapp enables an analyst to see whether there are officers within a unit or division with unusually low or high risk scores.

Group prediction table (for example data):

1.2 Quickstart

This page describes how to get up and running using the police early intervention system from the installation of the software, setup of the datasets (once cleaning and importing the data into a centralized database has been completed), and model generation and selection.

1.2.1 Installation

*[Deprecated? Not sure if setup.py works with current repo] Git clone the repository and install with *setup.py*:

```
python setup.py install
```

1.2.2 Setup

To set up a new police department with the early intervention system, you will need to write some configuration files that define the data sources available, how the code should connect to the database, and what features you want to create.

1.2.3 Database Connection and Data Definition

Initial setup is performed via two configuration files, one that contains database credentials, and one that contains configuration unique to the given police department:

- Database credentials are stored in a YAML file `default_profile` in the root directory. Use `example_default_profile` as a template:

```
PGPORT: 65535
PGHOST: "example.com"
PGDATABASE: "example"
PGUSER: "janedoe"
PGPASSWORD: "supersecretpassword"
DBCONFIG: "example_police_dept.yaml"
```

- `DBCONFIG` refers to a configuration file containing details of the individual police department, such as unit/district names and what data sources exist for feature generation `example_police_dept.yaml`.

1.2.4 Example Police Department Setup

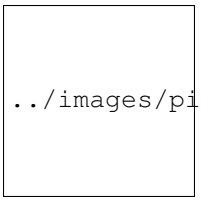
Content

1.2.5 Running Models

To run models on this new dataset, edit `default.yaml`.

Details of experiments, e.g. ranges of hyperparameters as well as features to be included, are stored in a YAML file - example in
python -m eis.run default.yaml

The following flowchart describes the process of model generation in relation to the ETL and feature generation work:



The model code takes a config file with lists of window sizes, dates to use as "fake todays" for temporal cross validation, model types, and lists of hyperparameters (to search), feature groups, and runs every possible combination.

At the moment, some of the feature generation work must be done by hand in adapting the SQL queries to the schema used for a given police department's dataset. In addition, new features can be added as described in the following section.

Note: This pipeline covers the steps from when the data is dumped into a centralized database for analysis.

1.2.6 Adding Features

To add a new feature or make other code changes, you should follow the instructions in `contributing.md`.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search