# Deploying Machine Learning Models for Public Policy: A Framework*

Klaus Ackermann
Department of Econometrics and Business Statistics, Monash University
klaus.ackermann@monash.edu

Joe Walsh
Center for Data Science and Public Policy University of Chicago
jtwalsh@uchicago.edu

Adolfo De Unánue
Center for Data Science and Public Policy University of Chicago
adolfo@uchicago.edu

Hareem Naveed
Center for Data Science and Public Policy University of Chicago
hareem@uchicago.edu

Andrea Navarrete Rivera
Center for Data Science and Public Policy University of Chicago
andrea.navarrete126@gmail.com

Sun-Joo Lee
Center for Data Science and Public Policy University of Chicago
sunjoo@uchicago.edu

Jason Bennett
Charlotte-Mecklenburg Police Department
jbennett@cmpd.org

Michael Defoe
Charlotte-Mecklenburg Police Department
mdefoe@cmpd.org

Crystal Cody
Charlotte-Mecklenburg Police Department
ccody@cmpd.org

Lauren Haynes
Center for Data Science and Public Policy University of Chicago
lnhaynes@uchicago.edu

Rayid Ghani
Center for Data Science and Public Policy University of Chicago
rayid@uchicago.edu

## ABSTRACT

Machine learning research typically focuses on optimization and testing on a few criteria, but deployment in a public policy setting requires more. Technical and non-technical deployment issues get relatively little[1] attention. However, for machine learning models to have real-world benefit and impact, effective deployment is crucial.

In this case study, we describe our implementation of a machine learning early intervention system (EIS) for police officers in the Charlotte-Mecklenburg (North Carolina) and Metropolitan Nashville (Tennessee) Police Departments. The EIS identifies officers at high risk of having an adverse incident, such as an unjustified use of force or sustained complaint. We deployed the same code base[2] at both departments, which have different underlying data sources and data structures. Deployment required us to solve several new problems, covering technical implementation, governance of the system, the cost to use the system, and trust in the system. In this paper we describe how we addressed and solved several of these challenges and provide guidance and a framework of important issues to consider for future deployments.

**ACM Reference Format:**
Klaus Ackermann, Joe Walsh, Adolfo De Unánue, Hareem Naveed, Andrea Navarrete Rivera, Sun-Joo Lee, Jason Bennett, Michael Defoe, Crystal Cody, Lauren Haynes, and Rayid Ghani. 2018. Deploying Machine Learning Models for Public Policy: A Framework. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3219819.3219911

## 1 INTRODUCTION

Most machine learning research and optimization is based on static data, but deployment often requires a wider range of considerations. Perhaps the best known example is The Netflix Prize [11], where Netflix offered $1 million for a model that reduces the error rate of their movie recommendation system by 10%. Although more accurate, the winning model turned out to be too complex to use in production [2].

For the past two years, the Center for Data Science and Public Policy (DSaPP) at the University of Chicago has worked with multiple police departments to build and deploy the first data-driven Early Intervention System (EIS) for police officers. The EIS identifies officers at high risk of having an adverse incident in a specified time horizon. The time horizon and the definition of an adverse incident is select based on the the available interventions by a department, allowing them to prevent those incidents with training, counseling, or other support [3, 4, 7]. The Metropolitan Nashville Police Department (MNPD) started using the EIS in fall 2016, and the Charlotte-Mecklenburg Police Department (CMPD) became the

*This work was done at the Center for Data Science and Public Policy, University of Chicago

[1] At KDD2015 there were 819 submissions to the Research Track and 189 submissions to the Industry & Government Track [1].

[2] https://github.com/dssg/police-eis/


*KDD '18, August 19–23, 2018, London, United Kingdom*

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3219911

first department to add a user interface and automation in November 2017.

Working with multiple police departments has given us a broad perspective on the issues that may arise around deployment. There is still a need to build, select, and assess models even in a production environment to ensure performance on new and changing data, but there are also many non-technical challenges that need to be solved, such as assigning someone responsibility to ensure the system operates correctly, providing training so the users know to use the system, and making the system as easy to use as possible.

CMPD has run an EIS since 2005, so they have processes in place to handle alerts and interventions. MNPD has had a less developed EIS program, so they needed more policy and system development. In-house experience differs for the two: CMPD has a large technical team with a big investment in IT and custom systems, while MNPD has a smaller IT department and generally uses out-of-the-box software. CMPD collects more data than MNPD, including information such as every training session an officer has attended. On the intervention side, MNPD has a large and robust team of staff psychologists who provide officers with support, while CMPD just recently hired its first staff psychologist. They use different definitions of "adverse incident." These and other issues change how the department could use the EIS in production.

This paper discusses our experience deploying machine learning models for police early interventions [3, 4, 7]. We outline four types of deployment issues—**Governance**, **Trust**, **Cost of Use**, and **Accuracy**—and some of the solutions we have found.

## 2 TECHNICAL IMPLEMENTATION

The input data collected at both departments represent the police officers, the actions they take, and the environment they operate in. For example, dispatch data, and features built from it, represents the environment an officer was placed, while the outcome of a traffic stop, such as a verbal warning or an arrest, represent an action an officer took. While few events significantly alter an officer's risk of future adverse incidents, an accumulation of events can. For example, we found that the number of suicide calls an officer has responded to predicts whether the officer will have an adverse incident. Complaints and policy violations are also important.

Police departments encode their event data differently. We used lookup tables[3] to map each department's codes to a single type while still allowing feature building from the raw data if desired. Categorizing dispatches or arrests allows the creation of a stable feature across historical data, to mitigate possible problems of reading data directly from a operational system. The operational data might change the dispatch encoding of a certain type at one point in time, resulting a distortion. For example, a feature that counts the number of dispatches of suicide calls can become distorted when the raw data encoding changes and so the feature suddenly switches to 0—a problem that occurred while testing the production system. This incident motivated us to build automated checks around such changes as described in section 3.3.

The EIS selects the best model group and provides daily predictions. We define a model group as an algorithm and a set of features and hyperparameters.[4] We define a model as a model group trained on a specific set of data (defined by start and end dates of the underlying data). This enables the evaluation of model groups across time to simulate performance in production mode. The production database environment has the same table definition, except the model group table is linked to the test environment, with the idea that the best model group is selected in the testing environment, and the realization of the model group (i.e. the model) at a certain point time is then used in production.

The analytic system is built on top of a operational database, which requires safeguards to provide consistent predictions. The features for the modeling are built based on a standardized *staging* schema across police departments. Unfortunately, due to inclusion of internal management data, such as internal affairs investigations, past data can change. Data integrity is not necessarily guaranteed between features built at point in time. To mitigate these possible errors, we rebuild the features and matrices every time a selected model group is used. The tradeoff here is processing time versus accuracy, but given that predictions are generated no more than once a day—a low frequency compared to online analytic systems [14]—we favored accuracy.

As of the beginning of February 2018, 102 officers have been reviewed by the professional standards captain, and 26 (25.5%) interventions have been initiated. That proportion is comparable with the test-set performance in the top 100.

## 3 THE FRAMEWORK

This framework is designed to be applied in public-policy settings as an external partner or internal analytic team trying to use machine learning methods in a production environment. At CMPD, DSaPP had a database dump, with all the identifying information removed or hashed by the department. At MNPD, DSaPP had direct access to a department analytics server and the department's operational database through via a database view.

At the beginning of any machine learning deployment in public policy settings, there are at least four issue areas that need to be considered, as shown in Table 1. We recommend thinking about each question as it applies to the policy area.

### 3.1 Accuracy

A necessary condition for deployment is that the model performs well out of sample. While the metric depends on the specific implementation, here we refer to any of those metrics (precision, recall, f1, etc) as accuracy. We acknowledge that we are overloading the definition of the word, especially when its meaning depends on context. There is a resource constraint for police early interventions because the department can only intervene on $k$ officers. On any given date, all officers are scored and ranked. Figure 1 plots out-of-sample precision over time for various model groups at CMPD. Linear models, such as a logistic regression, are more efficient and faster to compute, but this is less a concern in public policy settings where actions are taken by a human after a review process.

Data integrity has been an issue for accuracy. For example, rows get updated without timestamps. This creates a problem for the

[3] https://github.com/dssg/police-eis/blob/master/schemas/populate_tables/lookup/lookup_tables.yaml

[4] See the database definition: https://github.com/dssg/police-eis/blob/master/schemas/create_schemas/CREATE-SCHEMA-results.sql

**Table 1: The Framework**

| Area | Example Issues | Potential Consequences |
|---|---|---|
| **Accuracy** | • Models become less precise over time | • The department allocates interventions to the wrong officers<br>• The department misses opportunities to intervene on the highest risk officers |
| **Governance** | • What policies should be in place for managing the model and the people who interact with it?<br>• Who is responsible?<br>• Who is authorized? | • The system does not get used, the policy goal is not reached.<br>• The system gets used incorrectly, resources are wrongly allocated.<br>• The system stops working as planed at some point in time |
| **Trust** | • Black-box nature of the models.<br>• Incorrect interpretation of the outputs.<br>• The model fails to meet impossibly high expectations. | • Actors will not use the system if they do not trust it, even if the results are accurate. |
| **Cost to Use** | • Technical costs, akin to the Netflix Prize.<br>• Human costs, such as a steep learning curve or terrible interface. | • The institution might stop supporting the system, even if it can help their mission and achieve the policy goal.<br>• The supervisors might stop using the model, so opportunities for effective intervention are missed. |

evaluation of the models. To assess the performance of a model group across time, only information available at the time of prediction can be used to simulate the production situation. To address this concern, we employ a conservative **date setting** strategy. We use time stamps for when each event occurred and when it was known. In the case of the label data, which come from Internal Affairs investigations, there are three dates: when the event occurred, when the incident was reported, and when the incident was adjudicated. There can be significant lags between each.

Effective interventions will reduce the value of these statistics. If the EIS were perfectly accurate and the interventions perfectly effective, the EIS would have 0% precision, as the identified officers will receive interventions that prevent the adverse incident from occurring. Interventions have had little effect on historical results simply because so few supervisors acted on flags from a system they did not trust, but it could become a significant problem. Evaluating the effectiveness of the deployed system is an ongoing research project.

## 3.2 Trust

The police department and its supervisors will not use the EIS if they do not trust it. There are several ways that a machine-learning EIS can lose trust. Here are a few and how we are trying to prevent them.

**List stability** At the beginning of the project, we explored a wide range of hyper-parameters and found that random forests with fewer than 1,000 trees often had among the highest precision at the top 100. CMPD then asked if the system would become less accurate if they terminated their threshold EIS (which we were using as a feature). We re-ran the analysis and found that precision in the top 100 remained the same but the lists of high-risk officers differed significantly. We then re-trained the same random forests on the same features and the same data repeatedly, varying only the number of trees and the random seed, and found that the lists differed much more than we had expected. If supervisors look at the EIS list regularly, they could lose trust in the system because the list would have a lot of randomness to it. The problem originates with the algorithm: Bootstrap-based machine learning models are inherently random. We realized we needed to increase the number of trees in the random forest to reduce variance in the list, finding we need at least 10,000 trees to flag over 90% of the same officers from the same data in a reproducible manner (See Figure 2).

Similarly, the list of $k$ officers should change over time, as officers join and leave the department and their risks change [5], but it shouldn't change so much that it appears to be highly random. To test list stability, we simulated a run of the system for every day for 500 days. Figure 3, 4 and 5 compare the results of the consecutive pairs by day, week, and month. Figures 3, 4, and 5 show that the list changes more with time.

**Feature importances** In a previous paper [3], we showed a way to extract directionality from random forest feature importances. But that's at the model level. Police departments have expressed more interest in officer-level feature importances ("risk factors"). The goal has been to suggest where supervisors can look for possible

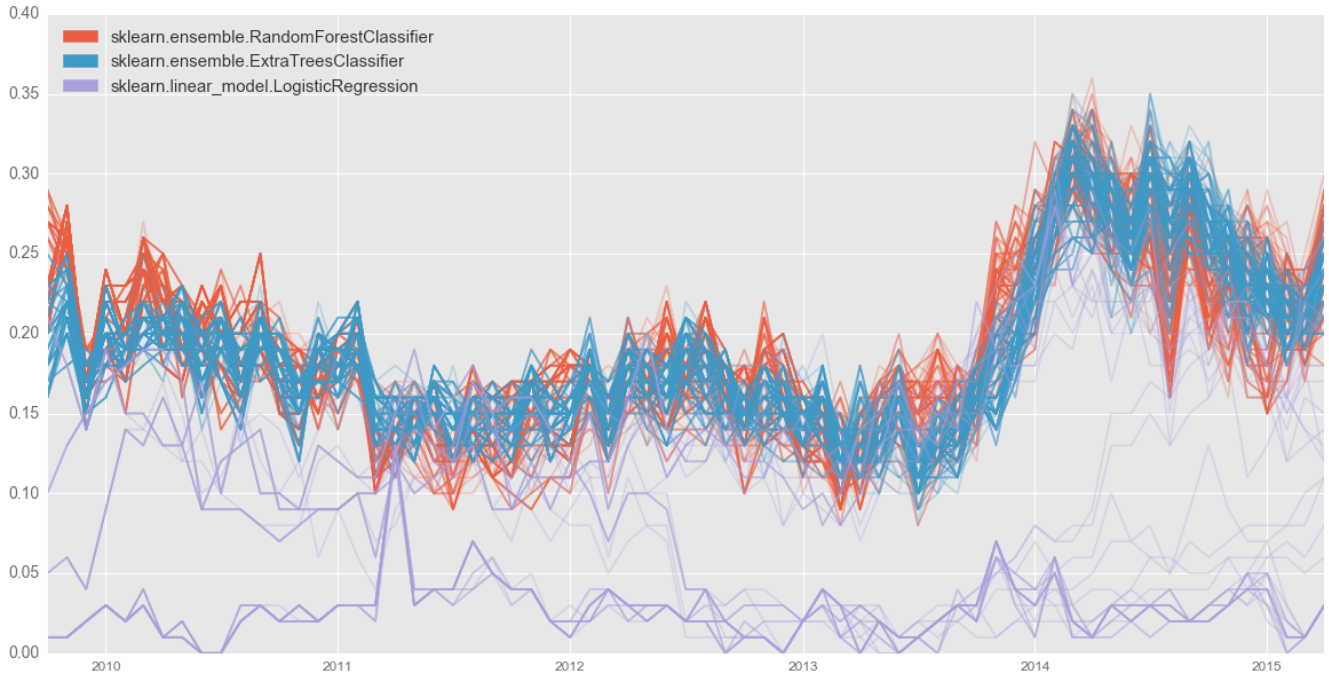


**Figure 1: Extra trees and random forest tend to have the highest precision in the top k (100 officers). Logistic regression is the most accurate for a few time periods but is not as stable across test dates. Each line represent a specific model group (a set of hyper-parameters and features) across time.**

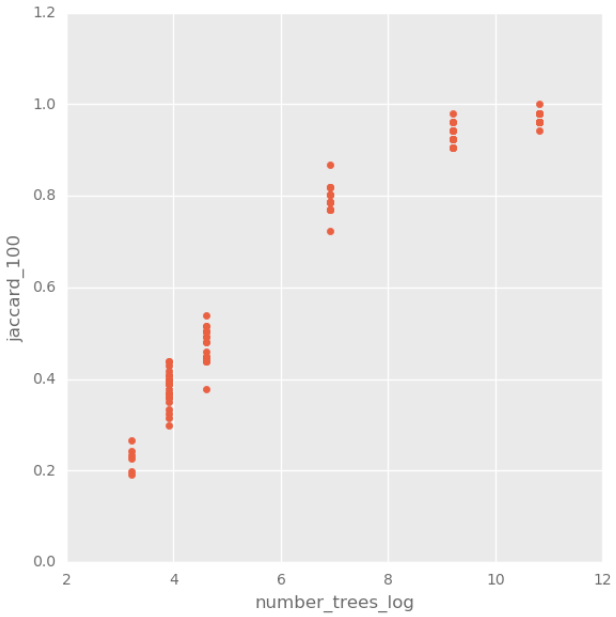


**Figure 2: The proportion of officers who appear on two lists generated by the same random forest model with varying random seeds. The more trees in the random forest, the more stable the list of the top 100 officers becomes.**

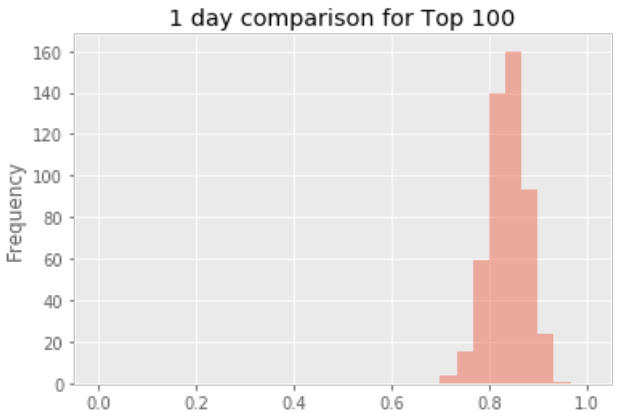


**Figure 3: The distribution of Jaccard similarities comparing each list to the previous day's list.**

issues to address, rather than identifying the "cause" for the officer's rank.

Our first approach was to use one model for prediction (random forest) and another model for explanation (logistic regression), but because it's also being trained on the outcome (rather than the prediction), the explanation model provides little insight about why an officer is being flagged. We discussed the potential tradeoff between accuracy and interpretability using a single model might

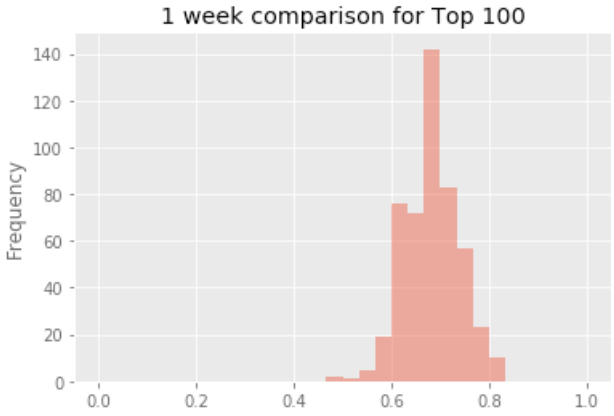


**Figure 4: The distribution of Jaccard similarities comparing each list to the previous week's list.**

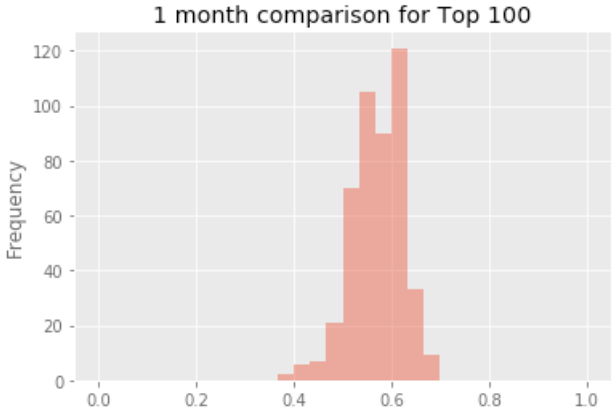


**Figure 5: The distribution of Jaccard similarities comparing each list to the previous month's list.**

bring, and the departments agreed that logistic regression's higher error rates were not worth the gain in interpretability it might offer.

We have since worked to find better ways to extract explanations from the predictive model. We selected the top 1% (equals 30 with 3000 features) features from the model-wide feature importance list and reported the five on which each officer is the biggest outlier. This approach selects features that are reasonably important to the model's predictions and that distinguish the officer from most others.[5] The aim is to provide the human reviewer with a starting point for investigation. In the production system the user can agree or disagree with, and comment on, those factors. Over time, we will be able to use this information to optimize the process and try alternative methods.

**Supervisor Training** Police supervisors don't have the time to study machine learning and the intricacies of how this machine learning EIS works. Both departments provided training to the supervisors so they can interpret the outputs and use the system correctly. For example, the supervisors learned that whether an officer is in the top 100 of the list is more important than whether an officer jumps from 20 to 10 (because there's some randomness in their position on the list). The training covered the importance of supervisor feedback to the system so the EIS can learn from their expertise and reduce false flags. The training also helps set expectations, so the supervisors don't expect perfect predictions.

**Professional Standards captain** Trust is probably most important for the Professional Standards captain. Because he must approve all alerts before an intervention can be made, he is the gatekeeper for frontline supervisor reviews and interventions, and losing his trust will lead to missed opportunities for intervention. He joins our regular meetings so that he can express concerns and we can address them as quickly as possible.

## 3.3 Governance

Policies are critical to the success of an EIS. The policies may be formal (enacted and codified by department leadership) or informal (decided by the people using the system). Policies are critical to the success of the EIS. For example, many EISs, including in Charlotte-Mecklenburg, have gone unused without department mandates.

**Changing review process.** Both CMPD and MNPD have modified their EIS flag review process (and may do so again) to meet their operational needs.

In Nashville, the department provided frontline supervisors with summary analyses for officers who met thresholds, and then the supervisors could decide whether to intervene. The department decided to formalize the process by sending notification letters to supervisors. They began with 100 letters every six months but have moved to 50 letters every three months. They plan to move to weekly letters as the department grows more comfortable with the system.

In Charlotte, the old system sent each flag directly to the officer's supervisor for review and possible intervention, and those supervisors treated the flags quite differently. Some supervisors never intervened, while others always did. Because the flags were handled in a distributed manner, department leadership did not have an easy way to monitor the performance of the system and how it was being used, and a lot of frontline supervisor time was being wasted. They were especially interested in monitoring the new EIS because they had never used a machine-learning system and weren't sure how it would work. CMPD addressed both issues by adding a layer to its review process. Now all flags must pass a preliminary review from the CMPD's Professional Standards captain before forwarding to the officer's supervisor. That way, CMPD gets a more consistent human review for each flag and makes a single person at headquarters responsible for monitoring system performance. It also increases efficiency because a person who reviews many cases can do each quickly, while frontline supervisors need more time to review each because they don't do it as often. The Professional Standards captain can review about 5% of officers a month, so we changed our models to optimize on that target.

As we prepared for deployment and looked at officer histories, we saw some officers making large jumps on the list. One officer who caught our attention rose almost 700 spots on the list almost overnight. It turned out that the officer had an IA incident that

[5]We thank Kit Rodolfa for the idea.

reasonably raised his rank. To ensure other large jumps make sense, CMPD requested flags for officers who made large jumps, so we started tracking rank changes over a day, week, month, quarter, and year, and we wrote a trigger that flags the $x$ largest jumps for review, where CMPD can change $x$ depending on its available resources.

Both departments now require supervisors to agree or disagree with flags and to provide reasons for dismissing a flag. In speaking with dozens of departments, we've learned many police supervisors believe they can more accurately identify officers at risk than a machine learning model. In other domains, such as Supreme Court [10, 12] and student-retention [15] forecasting, humans have proved more accurate at the top and bottom of the list and worse at the middle. Supervisor feedback provides a way to improve the model and test whether and how they are more accurate. If they are more accurate than the data-driven EIS, the department can stop using the EIS and work to encourage supervisors to act on their knowledge. If the EIS is more accurate, then we will have evidence that helps address some departments' concerns. We need at least six more months of data, though, because we made predictions for one year ahead.

Both departments are now requiring supervisors to provide information about the interventions they provide for flagged officers or a reason why they did not provide an intervention. Relatively little work has investigated the effectiveness of early interventions, especially to sub-groups of officers. The standard early interventions—counseling and training—may work better for some officers than for others. Indeed, there is anecdotal evidence that interventions are less effective for some officers, who have adverse incidents despite having received interventions. The lack of systematically collected data has hindered formal analysis, but CMPD's and MNPD's new policies will change that.

**Changing label definitions.** When we started this work, we had a general understanding of the predictive goal but few specifics. Each department discussed what types of incidents they were concerned about, which we used to define labels. CMPD wanted to focus on "major" adverse incidents, while MNPD expressed concern about any type of adverse incident. It was a good start, but both departments ended up modifying the label definitions. CMPD created a list of 55 policy numbers covering, among other things, unjustified use of force, abuse of position, bias and profiling, truthfulness, and unsatisfactory performance on which a formal finding would constitute an "adverse incident." MNPD changed its label definition to multi-day suspensions and terminations because officers with minor violations were dominating the list.

**Frequency of prediction** CMPD has had an EIS that can flag officers every day for more than 12 years, so they have been ready to score and act on daily flags. They generate officer predictions every night. MNPD started with biannual lists, so they they only needed to score officers twice a year. They have increased the letter frequency to every three months, so they only need to score officers four times a year. They plan to increase the scoring frequency to every week.

**Frequency of retraining** In Charlotte, we ran several experiments varying the time between the end of the train set and the beginning of the test set. Performance clearly degrades over time, starting seven days out. We have been retraining models every day, and we monitor for sudden changes in performance or new model types that may perform better. In Nashville, we retrain before each round of letters (currently every three months).

**Poor Model Performance and Unusual Changes** Existing EISs have performed poorly partly because they are not monitored and updated. CMPD is probably the most technically sophisticated police department in the country. They operated a threshold system for 12 years, and although they knew that it was inaccurate, no one knew just how inaccurate it was, let alone how many flags it had thrown. They had no policies or procedures to monitor system performance and change the system if performance failed to meet standards.

To help police departments avoid using a deficient EIS, we have built several automated checks. First, we started flagging potential data-integrity issues. Before implementing data-integrity checks, some of the data stopped updating, and the model started to produce strange results two months later. We now check that the tables update every day with the correct number of rows. But we also implemented checks on historical records. At one point, CMPD changed how it stores use-of-force data, including for past records, which meant the models we built were potentially vulnerable to bad performance. We added a time-based checksum to track changes to historical records so we know when those changes happen.

Second, we monitor model performance in the top $k$. As Figure 1 shows, precision in the top $k$ can vary significantly over time. Precision can change due to a changing baserate—for example, precision will likely change if the department modifies its policies—but it can also be due to data errors. We are flagging historically unusual changes in precision. We are also monitoring the performance of competing models, with a flag to notify us if one of them gets a higher precision in the top $k$.

Third, we monitor jaccard similarities and rank-order correlations over time. Although individual officers can jump quickly, the overall list should change more slowly. A list that changes too quickly or too slowly may indicate data issues, so every night, after scoring is complete, we store jaccard similarities (in the top 100) and rank-order correlations (the entire list) over the last day, week, month, quarter, and year in the database, and the system throws a flag if any of those numbers are in the $x\%$ of the tails of the historical distribution (where departments can choose $x$ to match the number of checks it is able to handle).

Human monitoring is also important but rarely happened under the old EIS. CMPD and MNPD have now assigned responsibilities and authorities to ensure the smooth operation of the system. In Charlotte, the IT supervisor is responsible for the application functionality, and the Captain is responsible for the EIS program and making sure supervisors process alerts. Supervisors are responsible for reviewing flags, and IT staff members are authorized to keep the EIS running.

While the departments are responsible for their EISs running on their own systems, both have authorized DSaPP's continued involvement. Both departments employ IT and database experts, but they have relatively little machine learning experience, so DSaPP continues to support model selection, assessment, and interpretation as the departments build those skills.

Each department's policy has affected how DSaPP supports the EIS. CMPD does not grant access to its system, but they do dump

data to DSaPP's system so we can run a parallel EIS on our side. While that slows development and analysis, it also helped us discover many subtle data issues, such as the wrong timezone in DSaPP's database, which threw our events counts off. We learned timestamps are important for matching records and learned to regret using serial primary keys because our keys can differ across sites.

MNPD gave us direct access to a server on their network, so development and analysis has been faster, but data integrity is still critical, and we are still checking the number of rows, tables, and events to ensure that the data import correctly.

### 3.4 Cost to Use

There are at least two types of costs to using a police EIS: technical and human.

*3.4.1 Technical costs.* A number of technical costs have been discussed elsewhere (for example, see Sculley, et al. [13]), but some are unique to this project. DSaPP's EIS is written to work on a linux operating system, PostgreSQL database (9.3+), Python (3.6), Github repositories [6], with enough CPUs and RAM to train and test models. Neither of our partners had those resources, so they purchased servers and installed the software, and DSaPP helped train them on those tools. There has been a learning curve for police IT. None of our partner departments were using linux, Python, Postgres, or Github. To reduce those costs, CMPD used Information Builders to transfer data from their operational databases into the EIS database. MNPD installed a foreign data wrapper to bring the data into Postgres, which is fast enough and enables the department's IT staff to continue to use their database of choice, SQL Server, for most tasks.

Even with Information Builders and foreign data wrappers and our code being open source, the technical costs are too high for many less technologically resourced departments, so DSaPP licensed its code to Benchmark Analytics, a private company that builds an electronic management system for police departments. Benchmark provides departments with a standard records system that the DSaPP EIS can operate on.

*3.4.2 Human costs.* Our system is designed to be used by supervisors for police officers. The other barrier to adoption is that the system has to have an intuitive user interface. Either the interface is similar to what the supervisors are used to working with, or the interface is designed very simply.

For CMPD, the system was incorporated into their existing employee management interface. They were able to use the codebase and modify the production schema and build an interface that was easy for supervisors to use as they were already familiar with it. Nashville, in contrast, used the generated list to send emails to the officers who were identified as at-risk by the system. Since the department did not have an existing system in place, it was the best to involve a low-tech solution to encourage adoption.

CMPD built the EIS to look like its other interfaces, so supervisors will feel comfortable using it. (See Kumar et al.'s research to learn more about the benefits of using a familiar interface [8].) We also wrote code to translate our feature names (e.g. `ir_id_1y_interventionsoftype_counseling_sum`) into English (`number of counseling interventions the officer has received in the last year`). They held focus groups and received feedback on what the interface should show and how it should show it.

CMPD's interface profiles each flagged officer's history, which saves time and improves feedback and understanding.

We originally provided risk scores but settled on providing ranks. Ranks make more sense because our models are optimized for relative risk (these officers are more likely to have an adverse incident than the rest of the officers) rather than absolute risk (e.g. an officer has a 50% chance of having an adverse incident). Not only are risk scores potentially misleading (or incorrectly interpreted as probabilities), but an officer's risk score may change significantly even if his/her rank in the department does not.

An early version of CMPD's EIS interface showed an officer's EIS risk history scaled by the officer's rank range. If the officer has consistently been in the top 10, his/her rank changes will appear to be large even though they're not. (The officer remains in the top 0.5%.) The y axis should have a minimum range (e.g. 1 to 100) so small changes don't look bigger than they are.

We needed to decide what to show supervisors as an officer's risk changes. For example, if the EIS stops flagging an officer before the supervisor review, we decided to no longer show the flag. Similarly, if an officer's risk rank changes before the supervisor review, the EIS should present the supervisor with the most recent information.

We built Tyra [9] to help our partners explore model performance. Tyra is a webapp that pulls standard analyses and reports from the database and presents them in an interactive manner, including precision at $k$ over time, precision and recall curves, ROC plots, the ranked list of officers and their corresponding labels, and feature distributions for the positive and negative label groups.

## 4 CONCLUSIONS

This paper provides a framework and lessons learned from deploying machine learning models for public policy. We provided 4 pillars—**Governance**, **Trust**, **Cost of Use**, and **Accuracy**—that every team should consider before moving toward deployment. The deployment of a machine learning-based Early Intervention System for police officers at the Charlotte-Mecklenburg and Metropolitan Nashville Police Departments served as a use case.

The next goal of this ongoing project is to improve the system. Incorporating supervisor feedback into the models should increase model precision over time. We plan to develop and test additional algorithms to extract officer-level feature importances. And we expect to analyze intervention effectiveness based on these machine learning predictions, which should be able to help supervisors assign the best interventions possible.

## ACKNOWLEDGMENTS

We would like thank the leadership and officers of the Charlotte-Mecklenburg Police Department and the the Metropolitan Nashville Police Department for sharing data, expertise, and feedback for this project, as well as Rob Mitchum for comments on early drafts. We especially would like to thank Matt Morley from the Metropolitan Nashville Police Department for his contributions to, and implementation of, this work.

# REFERENCES


[1] 2015. *KDD '15: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA.

[2] Xavier Amatriain and Justin Basilico. 2012. Netflix Recommendations: Beyond the 5 stars (Part 1). (2012). https://medium.com/netflix-techblog/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429 [Online; accessed 10-February-2018].

[3] Samuel Carton, Jennifer Helsby, Kenneth Joseph, Ayesha Mahmud, Youngsoo Park, Joe Walsh, Crystal Cody, CPT Patterson, Lauren Haynes, and Rayid Ghani. 2016. Identifying police officers at risk of adverse events. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 67–76.

[4] Crystal Cody, Estella Patterson, Putney Kerr, Jennifer Helsby, Joe Walsh, Rayid Ghani, Samuel Carton, Kenneth Joseph, Ayesha Mahmoud, and Youngsoo Park. 2016. Building Better Early Intervention Systems. *Police Chief Magazine* (2016).

[5] Wikipedia contributors. 2018. 2015 Texas pool party incident — Wikipedia, The Free Encyclopedia. (2018). https://en.wikipedia.org/wiki/2015_Texas_pool_party_incident [Online; accessed 10-February-2018].

[6] Tristan Crockett, Eric Potash, Jesse London, Matt Bauman, Erika Salomon, Avishek Kumar, Kit Rodolfa, Adolfo De Unanue, Tzu-Yun Lin, Klaus Ackermann, Rayid Ghani, Hannes Koenig, Andrea Navarette, and Benedict Kuester. 2018. Triage. https://github.com/dssg/triage/tree/v2.2.0https://github.com/dssg/triage/tree/v2.2.0. (2018).

[7] Jennifer Helsby, Samuel Carton, Kenneth Joseph, Ayesha Mahmud, Youngsoo Park, Andrea Navarrete, Klaus Ackermann, Joe Walsh, Lauren Haynes, Crystal Cody, et al. 2016. Early intervention systems: Predicting adverse interactions between police and the public. *Criminal Justice Policy Review* (2016), 0887403417695380.

[8] Mohit Kumar, Rayid Ghani, and Zhu-Song Mei. 2010. Data mining to predict and prevent errors in health insurance claims processing. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 65–74.

[9] Tzu-Yun Lin, Tristan Crockett, Adolfo De Unanue, and Rayid Ghani. 2017. Tyra. https://github.com/dssg/tyra/tree/v0.2. (2017).

[10] Andrew D Martin, Kevin M Quinn, Theodore W Ruger, and Pauline T Kim. 2004. Competing approaches to predicting supreme court decision making. *Perspectives on Politics* 2, 4 (2004), 761–767.

[11] Netflix. 2009. Leaderboard. (2009). https://www.netflixprize.com/leaderboard.html [Online; accessed 10-February-2018].

[12] Theodore W Ruger, Pauline T Kim, Andrew D Martin, and Kevin M Quinn. 2004. The Supreme Court forecasting project: Legal and political science approaches to predicting Supreme Court decisionmaking. *Columbia Law Review* (2004), 1150–1210.

[13] D. Sculley, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, and Michael Young. 2014. Machine Learning: The High-Interest Credit Card of Technical Debt. (2014).

[14] Parikshit Shah, Akshay Soni, and Troy Chevalier. 2017. Online Ranking with Constraints: A Primal-Dual Algorithm and Applications to Web Traffic-Shaping. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 405–414.

[15] James Soland. 2013. Predicting high school graduation and college enrollment: Comparing early warning indicator data and teacher intuition. *Journal of Education for Students Placed at Risk (JESPAR)* 18, 3-4 (2013), 233–262.