NIGH GOLDENBERG RASO & VAUGHN

Metadata

#Author	actuaryzhang@uber.com	SEMANTIC
#Date Modified	01/28/2019	SEMANTIC
#DateCreated	03/12/2016	SEMANTIC
#Title	Bouncer v3	SEMANTIC
Account	brooke.anderson@uber.com;	SEMANTIC
All Custodians	Anderson, Brooke;Breeden, Tracey;Brown, Greg;Cardenas, Philip;Chang, Frank;Fuldner, Gus;Kaiser, Roger;Kansal, Sachin;Lake, Carley;McDonald, Katy;Parker, Kate;Sheridan, Danielle;Silver, Nick;Sullivan, Joe;	SEMANTIC
All Paths	Anderson, Brooke: \20240905\Anderson, Brooke\Drive\EDISCO-25640_DR_25.zip; Anderson, Brooke: \20240905\Anderson, Brooke\Drive\EDISCO-25640_DR_25.zip; Anderson, Brooke: \MassTort_Category4_DRIVE\MassTort_Category4_DRIVE_6.zip; Anderson, Brooke: \MassTort_Category4_DRIVE\MassTort_Category4_DRIVE_6.zip; Breeden, Tracey: \JCCP_DRIVE002_002\JCCP_DRIVE002_002_18.zip; Breeden, Tracey: \JCCP_DRIVE002_002\JCCP_DRIVE002_002_18.zip; Breeden, Tracey: \MassTort_Category2_DRIVE\MassTort_Category2_DRIVE_9.zip; Breeden, Tracey: \MassTort_Category2_DRIVE\MassTort_Category2_DRIVE_9.zip; Breeden, Tracey: \MassTort_Category4_DRIVE\MassTort_Category4_DRIVE_6.zip; Breeden, Tracey: \MassTort_Category4_DRIVE\MassTort_Category4_DRIVE_6.zip; Brown, Greg: \Drive_003_002_Drive003_001\JCCP_Drive003_001\JCCP_Drive003_001_18.zip; Brown, Greg: \Drive_003_002_Drive003_001\JCCP_Drive003_001\JCCP_Drive003_001_18.zip; Cardenas, Philip: \Drive_003_002_Drive003_001\JCCP_DRIVE003_002\JCCP_DRIVE003_002_1.zip; Cardenas, Philip: \Drive_003_002_Drive003_001\JCCP_DRIVE003_002\JCCP_DRIVE003_002_1.zip; Chang, Frank: \EDISCO-25937_frank@uber.com\EDISCO-25937_frank@uber.com_82.zip; Chang, Frank: \EDISCO-25937_frank@uber.com\EDISCO-25937_frank@uber.com_82.zip; Fuldner, Gus: \JCCP-EDISCO-23800_2019JanMarch\JCCP-EDISCO-23800_2019JanMarch_216.zip; Fuldner, Gus: \JCCP-EDISCO-23800_2019JanMarch\JCCP-EDISCO-23800_2019JanMarch_216.zip; Kaiser, Roger: \JCCP_DRIVE005\JCCP_Drive005_120.zip; Kaiser, Roger: \JCCP_DRIVE005\JCCP_Drive005_120.zip; Kaiser, Roger: \MassTort_Category2_DRIVE\MassTort_Category2_DRIVE_9.zip; Kaiser, Roger: \MassTort_Category2_DRIVE\MassTort_Category2_DRIVE_9.zip; Kansal, Sachin: \EDISCO-25937_sachin.kansal@uber.com\EDISCO-25937_sachin.kansal@uber.com2_65.zip; Kansal, Sachin: \EDISCO-25937_sachin.kansal@uber.com\EDISCO-25937_sachin.kansal@uber.com2_65.zip; Lake, Carley: \EDISCO-24394_Drive\EDISCO-24394_Drive_102.zip; Lake, Carley: \EDISCO-24394_Drive\EDISCO-24394_Drive_102.zip; McDonald, Katy: \JCCP_DRIVE005\JCCP_Drive005_120.zip; McDonald, Katy: \JCCP_DRIVE005\JCCP_Drive005_120.zip; McDonald, Katy: \MassTort_Category2_DRIVE\MassTort_Category2_DRIVE_9.zip; McDonald, Katy: \MassTort_Category2_DRIVE\MassTort_Category2_DRIVE_9.zip; McDonald, Katy: \MassTort_Category4_DRIVE\MassTort_Category4_DRIVE_6.zip; McDonald, Katy: \MassTort_Category4_DRIVE\MassTort_Category4_DRIVE_6.zip; Parker, Kate: \JCCP_DRIVE006\JCCP_DRIVE006_35.zip; Parker, Kate: \JCCP_DRIVE006\JCCP_DRIVE006_35.zip; Sheridan, Danielle: \JCCP-EDISCO-23800_2019JanMarch\JCCP-EDISCO-23800_2019JanMarch_216.zip; Sheridan, Danielle: \JCCP-EDISCO-23800_2019JanMarch\JCCP-EDISCO-23800_2019JanMarch_216.zip; Silver, Nick: \EDISCO-24394_Drive\EDISCO-24394_Drive_102.zip; Silver, Nick: \EDISCO-24394_Drive\EDISCO-24394_Drive_102.zip; Silver, Nick: \MassTort_Category2_DRIVE\MassTort_Category2_DRIVE_9.zip; Silver, Nick: \MassTort_Category2_DRIVE\MassTort_Category2_DRIVE_9.zip; Sullivan, Joe: \EDISCO-25937_joesullivan@uber.com\EDISCO-25937_joesullivan@uber.com_11.zip; Sullivan, Joe: \EDISCO-25937_joesullivan@uber.com\EDISCO-25937_joesullivan@uber.com_11.zip	SEMANTIC
Application	Microsoft 2007 Word Document	SEMANTIC
Attachment Names	word	SEMANTIC
Begin Family	UBER_JCCP_MDL_003231342	SEMANTIC
Collaborators	alexclatterbuck@uber.com; jross@uber.com; bhav@uber.com; mananp@uber.com; butorac@uber.com; brianz@uber.com; dmitriy@uber.com; xin.luo@uber.com; kmcdonald@uber.com; brian.tan@uber.com; doyle@uber.com; rmawas@uber.com; ncf@uber.com; dhruv@uber.com; kchiu@uber.com; sjeon@uber.com; s.team-disabled@uber.com; UNKNOWN; safety-program@uber.com; uber.com	SEMANTIC
Confidentiality	Confidential	SEMANTIC
Date Created	03/12/2016 11:33 pm	SEMANTIC
Date Modified	01/28/2019 10:32 pm	SEMANTIC
DocID	1y4jv644kF23DXdWrpNaVqiXpVgft1vxytSss5cPHUgY	SEMANTIC
Document Type	Electronic File	SEMANTIC
End Family	UBER_JCCP_MDL_003231369	SEMANTIC

Trial Exhibit No.
P-01701

File Path	\20240905\Anderson, Brooke\Drive\EDISCO-25640_DR_25.zip	SEMANTIC
File Size	1942794	SEMANTIC
Filename	Bouncer v3_1y4jv644kF23DXdWrpNaVqiXpVgft1vxytSss5cPHUgY.docx	SEMANTIC
GoogleDocumentType	DOCUMENT	SEMANTIC
Hash Value	3552253cd3bcb66a7bdda30abee7192d	SEMANTIC
Hidden Content	Yes;	SEMANTIC
ILS All Bates	UBER_JCCP_MDL_003231342;UBER_JCCP_MDL_003231343;UBER_JCCP_MDL_003231344;UBER_JCCP_MDL_003231345;UBER_JCCP_MDL_003231346;UBER_JCCP_MDL_003231347;UBER_JCCP_MDL_003231348;UBER_JCCP_MDL_003231349;UBER_JCCP_MDL_003231350;UBER_JCCP_MDL_003231351;UBER_JCCP_MDL_003231352;UBER_JCCP_MDL_003231353;UBER_JCCP_MDL_003231354;UBER_JCCP_MDL_003231355;UBER_JCCP_MDL_003231356;UBER_JCCP_MDL_003231357;UBER_JCCP_MDL_003231358;UBER_JCCP_MDL_003231359;UBER_JCCP_MDL_003231360;UBER_JCCP_MDL_003231361;UBER_JCCP_MDL_003231362;UBER_JCCP_MDL_003231363;UBER_JCCP_MDL_003231364;UBER_JCCP_MDL_003231365;UBER_JCCP_MDL_003231366;UBER_JCCP_MDL_003231367;UBER_JCCP_MDL_003231368;UBER_JCCP_MDL_003231369	SEMANTIC
ILS Document Date	01/28/2019	SEMANTIC
ILS Prod Date	3/3/2025	SEMANTIC
ILS Prod Vol	JCCP_MDL106	SEMANTIC
LINKSOURCEBEGBATES	UBER_JCCP_MDL_001728489; UBER_JCCP_MDL_002221404; UBER_JCCP_MDL_002345578; UBER_JCCP_MDL_002345602; UBER_JCCP_MDL_004208267; UBER_JCCP_MDL_005200631; UBER_JCCP_MDL_005249724; UBER_JCCP_MDL_005442410; UBER_JCCP_MDL_005689136; UBER_JCCP_MDL_005689364; UBER_JCCP_MDL_005689389; UBER_JCCP_MDL_005704558; UBER_JCCP_MDL_005719651	SEMANTIC
Other Custodians	Breeden, Tracey;Brown, Greg;Cardenas, Philip;Fuldner, Gus;Kaiser, Roger;Lake, Carley;McDonald, Katy;Parker, Kate;Sheridan, Danielle;Silver, Nick;Anderson, Brooke;Chang, Frank;Kansal, Sachin;Sullivan, Joe;	SEMANTIC
Primary Date	03/12/2016 11:33 pm	DOC_TYPE_ALIAS
Production Volume	JCCP_MDL106;	SEMANTIC
Redacted	No	SEMANTIC
Sort Date	01/28/2019 10:32 pm	SEMANTIC
SourceHash	0704653c760110e2fbad787b82a22764	SEMANTIC

Bouncer v3

Sunny Jeon
Data Scientist | Trust & Safety

March 28, 2016

Summary: This document provides an overview of *Bouncer*,[1] an intelligent decision system for anticipating and preventing safety incidents on the Uber platform (e.g., accidents, interpersonal conflicts). Statistical machine learning models predict whether drivers will or will not be involved in a safety incident 7, 14, 30, and 60 days into the future. High risk users are targeted for interventions that prevent safety incidents in randomized controlled trials. The key performance indicator is reduction in safety incidents.

Contents

1. Motivation
2. Safety Incident Prevention Strategy
3. Predictive Model
4. Predictive Performance in US Markets
5. Improvements From Previous Models
6. Most Important Predictors
7. Challenges
8. Next Steps
9. Team
10. Appendix

Variable List and Definitions (spreadsheet)
Overview of Intervention Framework (deck)
Bouncer Supplementary Analysis (report)

[1] Previous versions of Bouncer: Bouncer v2+, Bouncer v2, Bouncer v1.

1. Motivation [back to contents]

As Uber usage continues to grow, the company may encounter a greater number and variety of safety risks—including fatal automobile accidents and interpersonal conflicts like physical altercations and sexual misconduct. Although safety incidents are rare and may appear random, **many safety incidents are predictable**. That is, safety incidents follow patterns and have precursors that can be leveraged to forecast them before they happen. For example:

- Safety incident rates increase dramatically between 1-3am, on Saturday nights/Sunday mornings, and on holidays and other days of major social gatherings (e.g., World Series, Outside Lands). [See analysis: Safety Hot Times/Days Analysis]
- Users that have caused business critical safety incidents have substantially more safety tickets going into the incident and substantially lower ratings in their first 10 trips than users that have never caused a major safety incident. [See analysis: Precursors to Business Critical Safety Incidents]
- Safety risks travel through social networks. That is, it is possible to predict a driver's safety ticket history using just their friends' ticket history. [See analysis: Safety Across Referral Networks]
- Safety incidents do not occur randomly across space, but are clustered in specific types of neighborhoods. [See analysis: Safety Hot Spots Analysis]

If safety incidents are predictable, they are preventable. This is the insight behind *Bouncer*, an intelligent decision system for anticipating and preventing safety incidents (e.g., accidents, interpersonal conflicts). Statistical machine learning models predict whether drivers will or will not be involved in a safety incident 7, 14, 30, and 60 days into the future. To prevent incidents from occurring, high risk users are targeted for a variety of interventions that have reduced safety incidents in randomized controlled trials. Key results from the project:

- At the **30/60 day forecasting windows**, predictive models correctly identify ~90% of drivers that cause dangerous driving incidents (with ~20% precision) and ~45% of drivers that cause interpersonal conflicts (with ~5% precision) in out-of-sample tests.
- At the **7/14 day forecasting windows**, predictive models correctly identify ~70% of drivers that cause dangerous driving incidents (with ~10% precision) and ~25% of drivers that cause interpersonal conflicts (with ~3% precision) in out-of-sample tests.
- Predictive models for both dangerous driving and interpersonal conflicts can be tuned to have >80% precision at the 30/60 day forecasting windows, but at the cost of recall.
- Interventions based on safety messaging **reduce safety incident counts by >10%** in controlled experiments conducted in the US (see SASSY Project Brief).

Bouncer is still very much a work in progress, but to solicit feedback and suggestions for improvement, this document describes Bouncer's incident reduction strategy, forecasting methodology, and results from performance evaluations.

Figure 1: Bouncer Intervention Framework

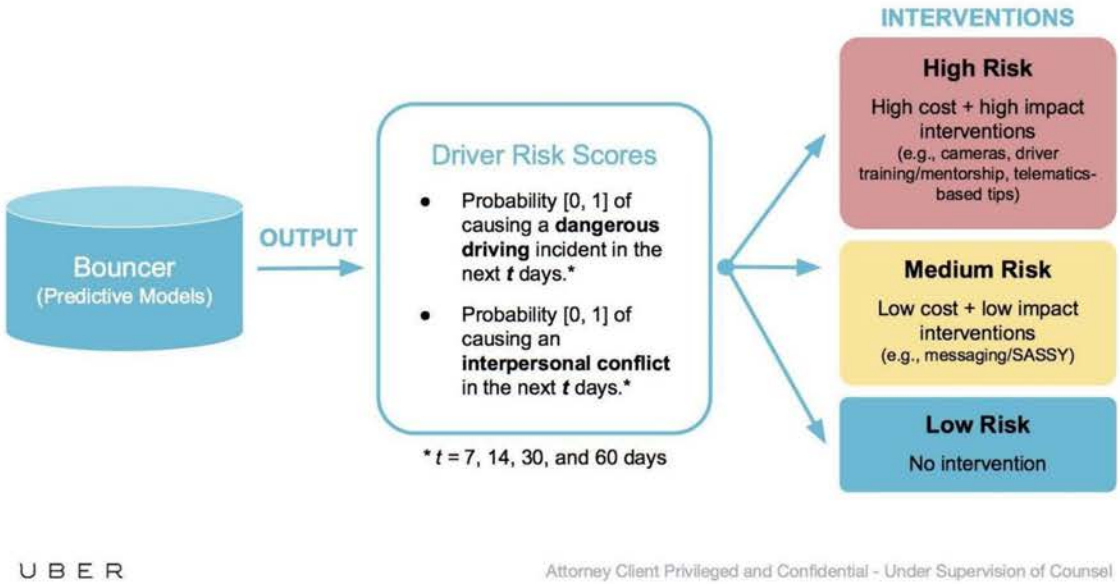
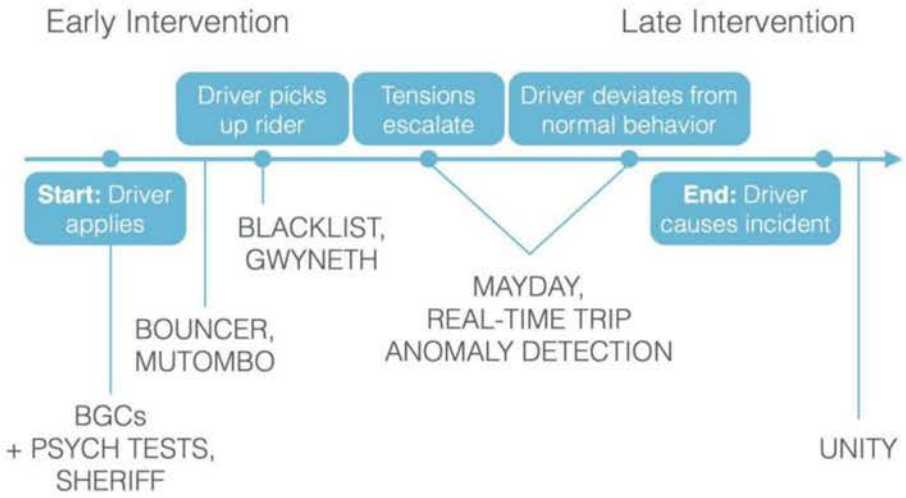


Figure 2: Products for Prevention by Stages of Trip Life Cycle



2. Safety Incident Prevention Strategy [back to contents]

Bouncer reduces safety incidents on the Uber platform by using patterns in historical data to identify which users are at greatest risk of causing a safety incident in the future. These users are then targeted for a variety of preventative interventions based on their specific risk profile. The incident reduction strategy is driven by 3 components:

1. Data Pipeline
2. Predictive Models
3. Automated Interventions

Figure 1 provides a high-level overview of the Bouncer intervention program. Figure 2 depicts where in the trip life cycle Bouncer -- as well as other safety products + initiatives -- intervene.

Data Pipeline

To compute risk scores, Bouncer draws on a data pipeline of internal and external data relevant for detecting high risk users. Data include:

- Zendesk/Bliss ticket history
- Account data from Vertica: e.g., tenure, trip count, ratings, wages, ETA differences, times worked (night v. day shifts), vehicle background, referral network, hours worked per day.
- Holiday / event schedule
- Weather
- Telematics: e.g., hard braking + accelerations.
- Criminal background (Checkr) -- *not yet available*
- Driving record (external vendor: Checkr) -- *not yet available*
- Behavioral (external vendor: MIDOT + HireVue) -- *not yet available*
- Social geography (e.g., location of bars, major roads, census blocks, population density) -- *not yet available*

The full variable list (including definitions and sources) is available here. In Section 5 -- Most Important Predictors, I identify the predictors with the greatest predictive power.

Predictive Models

Bouncer draws on the data pipeline to build a suite of statistical machine learning models for forecasting safety incidents. The current iteration of Bouncer (v3) focuses on identifying drivers/partners that are likely to cause two types of safety incidents **7, 14, 30, and 60 days into the future**:

- **Dangerous Driving:** accidents, distracted driving, poor/erratic driving, and traffic violations.

- **Interpersonal Conflicts:** verbal and physical altercations, inappropriate behavior, and sexual misconduct (instigated by the driver).

To accomplish this predictive task, the models leverage a variety of binary classification algorithms -- including Random Forests, Support Vector Machines, AdaBoost, and Stochastic Gradient Boosting -- to discriminate between drivers that are likely to cause a dangerous driving incident or interpersonal conflict, and those that are unlikely to. The best performing models are identified by systematically back-testing existing data, and predictive performance is estimated using an out-of-sample testing protocol.

Automated Interventions

Once Bouncer identifies drivers that are likely to cause a safety incident, the drivers are targeted for a series of interventions designed to prevent the incident from occurring. In other internal work, I have shown that the following safety messaging interventions are effective at reducing safety incident counts in randomized controlled trials:

- **Positive Reinforcement:** weekly messages acknowledging good behavior or good driving performance (see labs page).
  - Treatment reduces number of safety incidents by **-6%** ($p = 0.08$).

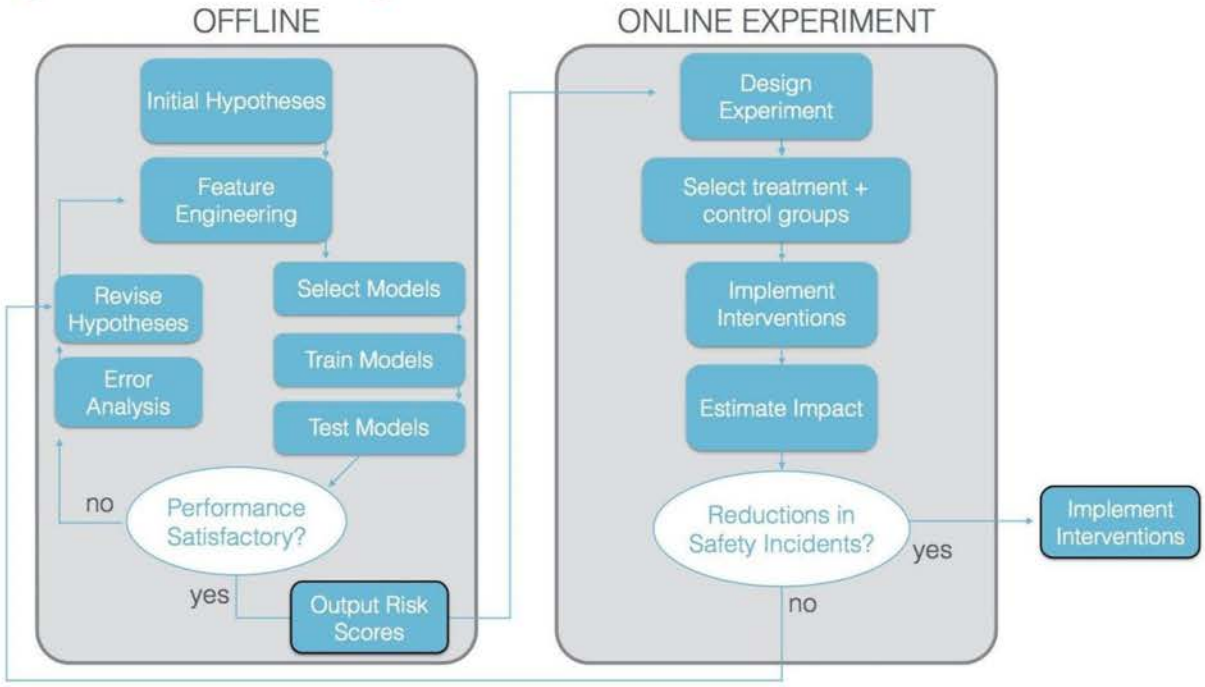
- **Rider Feedback:** weekly messages providing 3 randomly selected (scrubbed) verbatim comments from riders (see labs page).
  - Treatment reduces number of safety incidents by **-8%** ($p = 0.03$).
  - Results driven by impact on dangerous driving tickets, which saw a **-10%** decrease compared to control group ($p < 0.01$).

- **SASSY (Smart + Automated Safety SMS System):** personalized safety messages that are algorithmically composed based on each driver's weekly performance (see SASSY Project Brief for more details).
  - SASSY -- and in particular the reactive bundle of treatments (pro-tips + warnings for repeat violations) -- reduces safety incident counts by **-12%** ($p < 0.01$) and dangerous driving ticket counts by **-14%** ($p < 0.01$).

These messaging interventions have intentionally been designed to be subtle and low-cost because the interventions are targeted towards those that are *predicted* to cause a safety incident -- not necessarily those that have already caused an incident. As such, strong interventions like banning/off-boarding are probably inappropriate. However, if the predictive models can demonstrate reliability over time, it may be worth experimenting with moderately costly but potentially high-impact interventions, like restricting use during high-risk times (1-3am), requiring driver training, or installing cameras in partner

vehicles for audio, video, GPS, and/or telematics monitoring. This slide deck describes the prevention action playbook.

Figure 3: Bouncer Testing Process



Testing Process

Figure 3 depicts the process by which Bouncer is developed and tested. The process begins with offline experimentation, which involves an iterative process of building predictive models, testing them, and revising them based on the test results and findings from error analysis. Once the predictive performance of the models is satisfactory, Bouncer is launched in select cities using a randomized experimental framework that randomly divides drivers into treatment (receive Bouncer interventions) and control (do not receive Bouncer interventions) groups to isolate and estimate the impact of Bouncer on safety incident counts. If the experiment results indicate Bouncer reduces incident counts, it is scaled up to other cities. If the experiment results cannot identify effects from Bouncer, both the interventions and the predictive models are re-designed by repeating the entire offline-online experiment process.

3. Predictive Model [back to contents]

Strategy

Bouncer simplifies the predictive task into a binary classification problem by learning to predict drivers that are and are not likely to cause a (i) dangerous driving or (ii) interpersonal conflict safety incident t days into the future, where t = 7, 14, 30, and 60.

The modeling strategy is designed to address 3 specific challenges to predicting safety incidents:

1. Safety incidents are rare events.
2. Safety incident data is noisy and based on self-reported cases (i.e., measurement error in dependent variable).
3. High cost of false negatives, so need to minimize this and err on the side of caution (i.e., maximize recall).

To address (1.) the rare events problem, I build the predictive models using 3 different subsampling techniques. The first is *simple down-sampling*, which involves selecting a training set that is artificially balanced so that it has an approximately equal proportion of positive cases (high-risk drivers) and negative cases (low-risk drivers). A second approach is *internal down-sampling*, which involves artificially balancing the data in the resampling process used for model training (i.e., balance each of the bootstrap samples). A third approach is *synthetic minority over-sampling technique* (SMOTE), which involves both down-sampling the majority class and upsampling the minority class by synthesizing new cases using K-nearest neighbors.

To address (2.) the noisy data problem, I train models using dichotomized operationalizations of the dependent variables (safety incident counts). The idea here is that, due to noise and reporting biases, drivers with say, 3 dangerous driving tickets may not be that different in terms of safety risk than drivers with 2 or 4 dangerous driving tickets. The biases introduced by measurement error can then be attenuated by collapsing the ticket data into a binary variable indicating whether a driver has received at least 1 dangerous driving ticket or not. Alternatively, due to noise, a nontrivial proportion of low risk drivers may accidentally or incorrectly receive a safety ticket at some point, making measurement error much higher in the group of drivers with 1 incident ticket than drivers with $\geq$2 tickets. In this case, predictive performance could be improved by removing drivers with exactly 1 incident ticket during the model training process, so that the models can better learn how to discriminate between low and high-risk drivers.

To address (3.) the high cost of false negatives, the predictive models are tuned to maximize recall (# of True Positive/Total # of Positive Cases) without sacrificing precision too much by manipulating the probability cutoff used for classification. In future iterations, cost-sensitive learning will be implemented.

Data

The current iteration of Bouncer focuses on detecting high risk drivers in 4 major US markets: San Francisco, Los Angeles, Chicago, and New York City.

To build models for these cities, I collect data on a stratified random sample of 28,986 drivers (6,805 Chicago, 7,517 LA, 7,538 NYC, 7,126 SF). Because safety incidents are uncommon, users with safety incidents are over-sampled and users without safety incidents are under-sampled to balance the data (this dataset is further balanced before and during model training using simple down-sampling, internal down-sampling, and SMOTE, see Figure 4 for details). This sample was drawn by pulling:

- All 705 SF, LA, Chicago, and NYC drivers that have received at least one L3 and L4 tickets between January 1 - September 30, 2015,
- All 4,151 SF, LA, Chicago, and NYC drivers that have received at least one L2 ticket between January 1 - September 30, 2015,
- A 10% random sample of SF, LA, Chicago, and NYC drivers that have received at least one L1 ticket between January 1 - September 30, 2015 (12,134 drivers), and
- A random sample of 11,996 SF, LA, Chicago, and NYC drivers.

For each user, I create a day-level time-series that goes from January 1, 2015 to September 30, 2015, and includes data on a variety of user features built from Zendesk ticket data and user account data (e.g., changes in ratings, ETAs, cancellation rates, trip counts, times worked, fares earned/paid, etc.). These data are operationalized as over 200 distinct predictors. This Variable List provides definitions and sources.

Data Pre-Processing

The data are pre-processed in several ways:

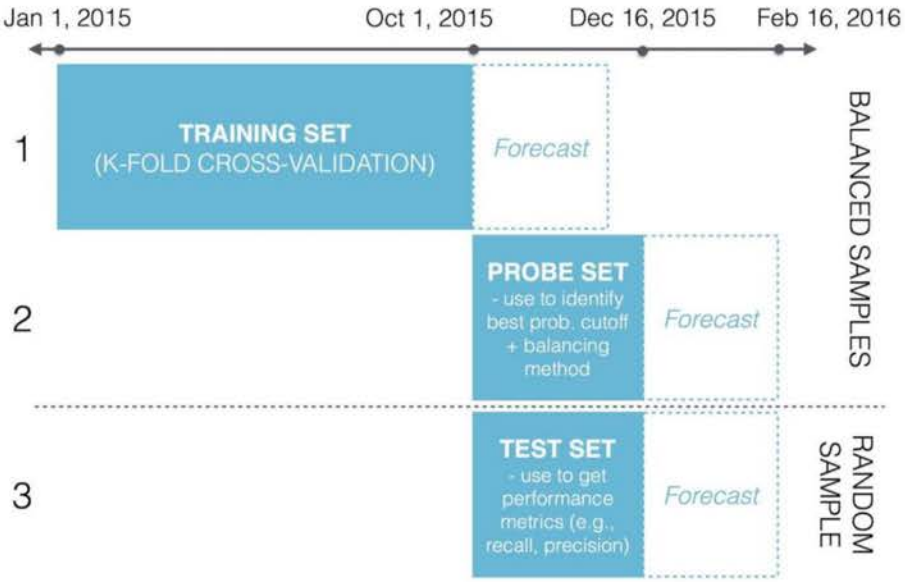
- All predictors are centered and scaled to have mean 0 and standard deviation 1. In the model training process, this normalization is made within resampling loops.
- All predictors with near zero variance are removed. Near zero variance predictors are those that meet two conditions: (i) less than 10% of observations are unique, and (ii) the ratio of frequencies for the most common value over the second most common value is greater than 95/5.
- Highly correlated predictors are removed by computing pair-wise correlations, identifying highly correlated predictors (>0.80), and then removing the predictor with the higher mean pair-wise correlation across the full correlation matrix.

Model Training and Testing

The model training and selection process is conducted by dividing the balanced data into a ***training set*** and a ***probe set***. Observations between January 1 - September 30, 2015 are

used for model training, and observations between October 1 - December 16, 2015 are set aside to use for the probe set (see Figure 4). In all samples, only data before December 16 is used to build predictors. However, since the models make forecasts 7, 14, 30, and 60 days into the future, the response variables draw on data up to February 16, 2016.

Figure 4: Samples Used for Model Training and Testing



Notes: Figure shows how the time-series data are divided into training, probe, and test sets. The training and probe sets are balanced random samples, and the test set is a random sample.

The purpose of the training set is to identify the optimal tuning parameters for each algorithm, which is done using k-fold cross-validation (with $k = 5$), a procedure that divides the training set into k random and equally-sized subsets, and then sequentially picks one of the k subsets to use as the validation data and the remaining $k - 1$ subsets as the training data. The tuning parameters that maximize the area under the ROC curve are then selected and used to generate predictions. Tuning parameters selected in this way include:

- *Random Forests*: number of randomly selected predictors used in each split.
- *Support Vector Machines* (with Radial Basis Function Kernel): sigma hyper parameter, cost.
- *AdaBoost* (Boosted Classification Trees): number of trees, max tree depth, learning rate.
- *Stochastic Gradient Boosting* (Classification): number of boosting iterations, maximum tree depth, learning rate/shrinkage, and minimum terminal node size.

The purpose of the probe set is to optimize additional model parameters and specifications to fit the model for our purposes, such as the the optimal subsampling method for addressing the rare events problem (simple down-sampling v. internal down-sampling v. SMOTE), and to identify the optimal probability cutoff for maximizing recall without sacrificing precision. The following is a full list of the different specifications I test:

- Subsampling techniques:
  - Simple down-sampling
  - Internal down-sampling
  - SMOTE
- Optimal probability cutoff for maximizing recall without sacrificing precision
- Operationalization of dependent variable
  - Safety incident count ≥ 1 over next 7, 14, 30, 60 days
  - Safety incident count ≥ 2 over next 7, 14, 30, 60 days
  - **Note*: Bouncer v2 predicts incident counts using raw Zendesk ticket data
  - **Note*: Bouncer v1 predicts incident rates (incident count/trip count)
- Subset
  - Subset to drivers that got activated during the sample time period
  - Data aggregated by city
  - Data aggregated across all cities
  - Remove observations where incident count = 1 on the dependent variable (to reduce noise during model training process)
- Predictors
  - Full set of predictors
  - Top 50 predictors (by variable importance scores)

After model specifications are optimized against the probe set, a separate random sample of 17,836 drivers from the same US cities (4,398 Chicago, 4,572 LA, 4,484 NYC, 4,382 SF) is selected for the ***test set***. The accuracy of predictions for the random test set is used to assess the predictive performance of the final models -- which is what we would expect to see "in the wild" if Bouncer were launched.

4. Predictive Performance in US Markets [back to contents]

Performance Metrics

The primary performance metrics are precision, recall, and percent positive predictions on the out-of-sample random test set, where:

$$Precision = \frac{True\ Positives}{True\ Positives\ +\ False\ Positives}$$

$$Recall = \frac{True\ Positives}{True\ Positives\ +\ False\ Negatives}$$

$$Percent\ Positive\ = \frac{True\ Positives\ +\ False\ Positives}{Total\ Predictions\ Made}$$

Because the costs of safety incidents are high, a good model will have high recall to minimize false negatives. An ideal model, however, will also have high precision, particularly when the costs of false positives are nontrivial. Percent positive gives the percent of "positive" (high-risk) predictions. Because one prediction is generated for every driver at every point in time, percent positive gives the percent of the entire driver population that needs to be targeted to reach recall. Larger recall:percent positive ratios thus indicate stronger performance.

Main Results

Bouncer can be optimized for many different purposes and performance metrics. Suppose, for example, that our sole interest is in identifying 100% of drivers that are likely to cause a dangerous driving incident or interpersonal conflict 30 days into the future (to be as risk-averse as possible). Then, the probability cutoff used for classification can be manipulated to maximize recall at the cost of precision. Alternatively, suppose we want to maximize both recall and precision due to some cost produced by false positives.[2] Then the probability cutoff can be manipulated to maximize the F_1 *Score*, which is the harmonic mean of recall and precision (F_1 *Score* = 2*precision*recall/(precision + recall)). To place more weight on recall, the F_2 *Score* and F_3 *Score* are options. The generic formula for computing an F_β Score is:

[2] In the case of Bouncer, the cost of false positives is the unnecessary spamming of safety messages to our drivers/partners. Given the volume of messages that are already sent to our drivers/partners by ops, comms, supply, and many other Uber teams, there is an interest in reducing unnecessary spamming, which would likely diminish the value of each individual message that is sent to drivers/partners (see SMS Volume Report). It would also likely further reduce the already low email open rate (anecdotally at ~30%), and potentially reduce the efficacy of each individual message.

$$F_\beta\ Score\ = (1\ +\ \beta^2)\ \frac{Precision * Recall}{\beta^2 * Precision + Recall},$$

where $\beta > 1$ puts more weight on recall and $0 < \beta < 1$ puts more weight on precision.

Figure 5: Dangerous Driving Predictions

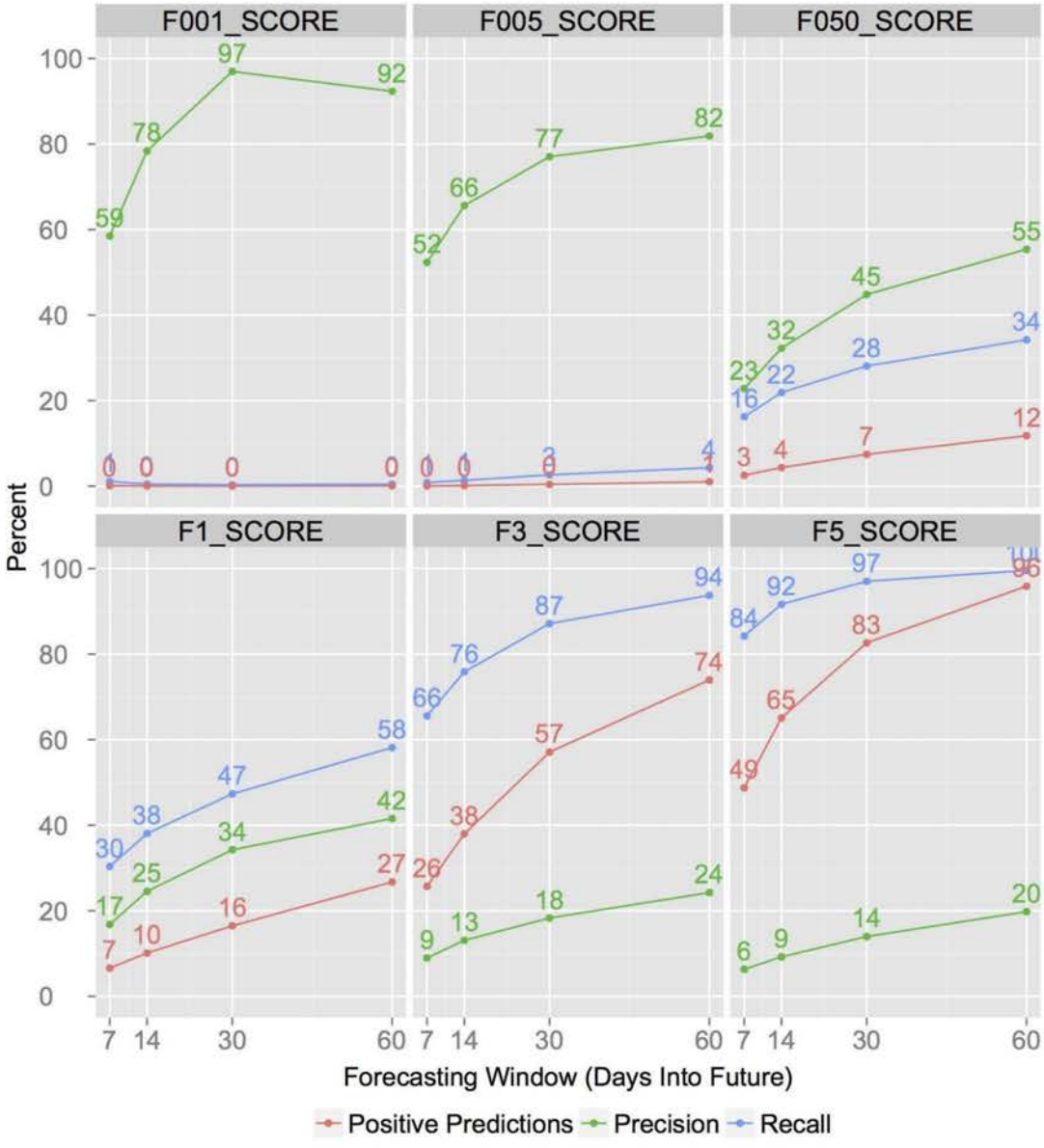
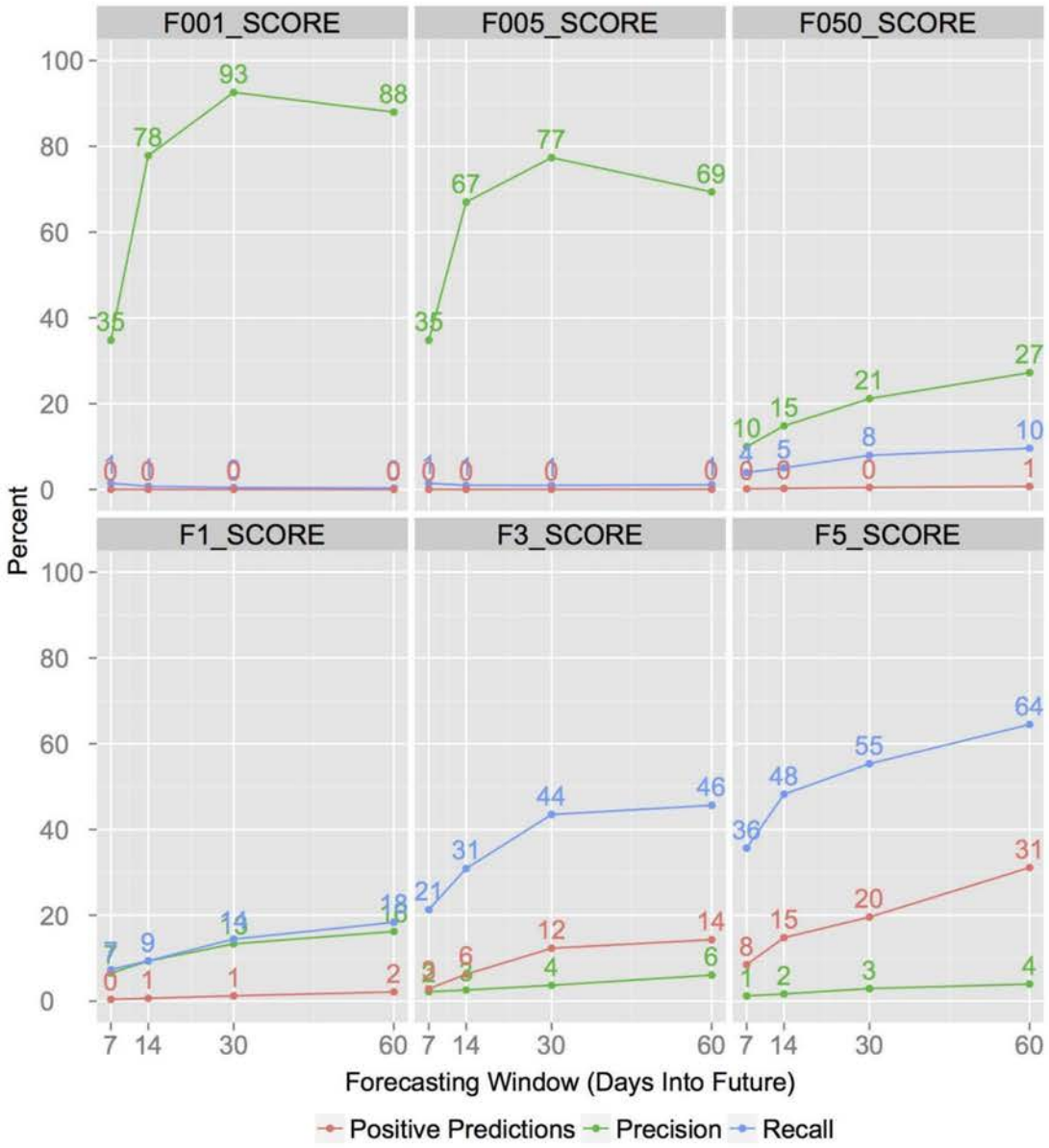


Figure 6: Interpersonal Conflict Predictions



To demonstrate these different applications of Bouncer, Figures 5-6 present the precision, recall, and percent positive for out-of-sample predictions when selecting the probability cutoffs that maximize the $F_{0.01}$ Score, $F_{0.05}$ Score, $F_{0.50}$ Score, F_1 Score , F_3 Score, and F_5 Score.

When placing slightly more weight on recall (e.g., F_3 Score models), Bouncer predicts dangerous driving with 66% recall (9% precision) at the 7 day forecasting window, with 76% recall (13% precision) at the 14 day forecasting window, with 87% recall (18% precision) at the 30 day forecasting window, and with 94% recall (24% precision) at the 60 day forecasting window. Interpersonal conflicts are predicted with 21% recall (2% precision) at the 7 day forecasting window, with 31% recall (3% precision) at the 14 day forecasting window, with 44% recall (4% precision) at the 30 day forecasting window, and with 46% recall (6% precision) at the 60 day forecasting window. These recall metrics can be boosted even further by using an F_β Score with $\beta > 3$ (e.g., F_5 Score models).

Recall and precision is substantial lower for the interpersonal conflict models, indicating that interpersonal conflicts (altercations, inappropriate behavior) are harder to predict. However, the interpersonal conflict predictions are more efficient in the sense that they do not need to cast as wide of a net to correctly anticipate a significant proportion of future safety violations. In the F_3 Score models, for example, the recall:positive predictions ratio is 94% / 74% = 1.27 for dangerous driving at the 60 day forecasting window. The recall:positive predictions ratio for interpersonal conflict predictions is 46% / 14% = 3.29 at the same forecasting window. The substantive implication is that for every 1% increase in the proportion of drivers predicted to cause an incident in the future (percent positive), Bouncer identifies almost 3x more drivers causing interpersonal conflicts than drivers causing dangerous driving incidents (recall) on the average.

Bouncer can also be tuned to maximize precision at the cost of recall. These types of high precision -- low recall forecasts can be useful for high impact -- high cost interventions that require a high degree of confidence due to the cost of false positives (e.g., audio/video monitoring, deactivation/waitlisting). When maximizing the $F_{0.05}$ Score, the dangerous driving predictions have 52% precision (1% recall) at the 7 day forecasting window, 66% precision (1% recall) at the 14 day forecasting window, 77% precision (3% recall) at the 30 day forecasting window, and 82% precision (4% recall) at the 60 day forecasting window. The interpersonal conflict predictions have 35% precision (1% recall) at the 7 day forecasting window, 67% precision (1% recall) at the 14 day forecasting window, 77% precision (1% recall) at the 30 day forecasting window, and 69% precision (1% recall) at the 60 day forecasting window. These precision metrics can be boosted even further by using an F_β Score with $\beta < 0.05$ (e.g., $F_{0.01}$ Score models).

Predicted versus Actual

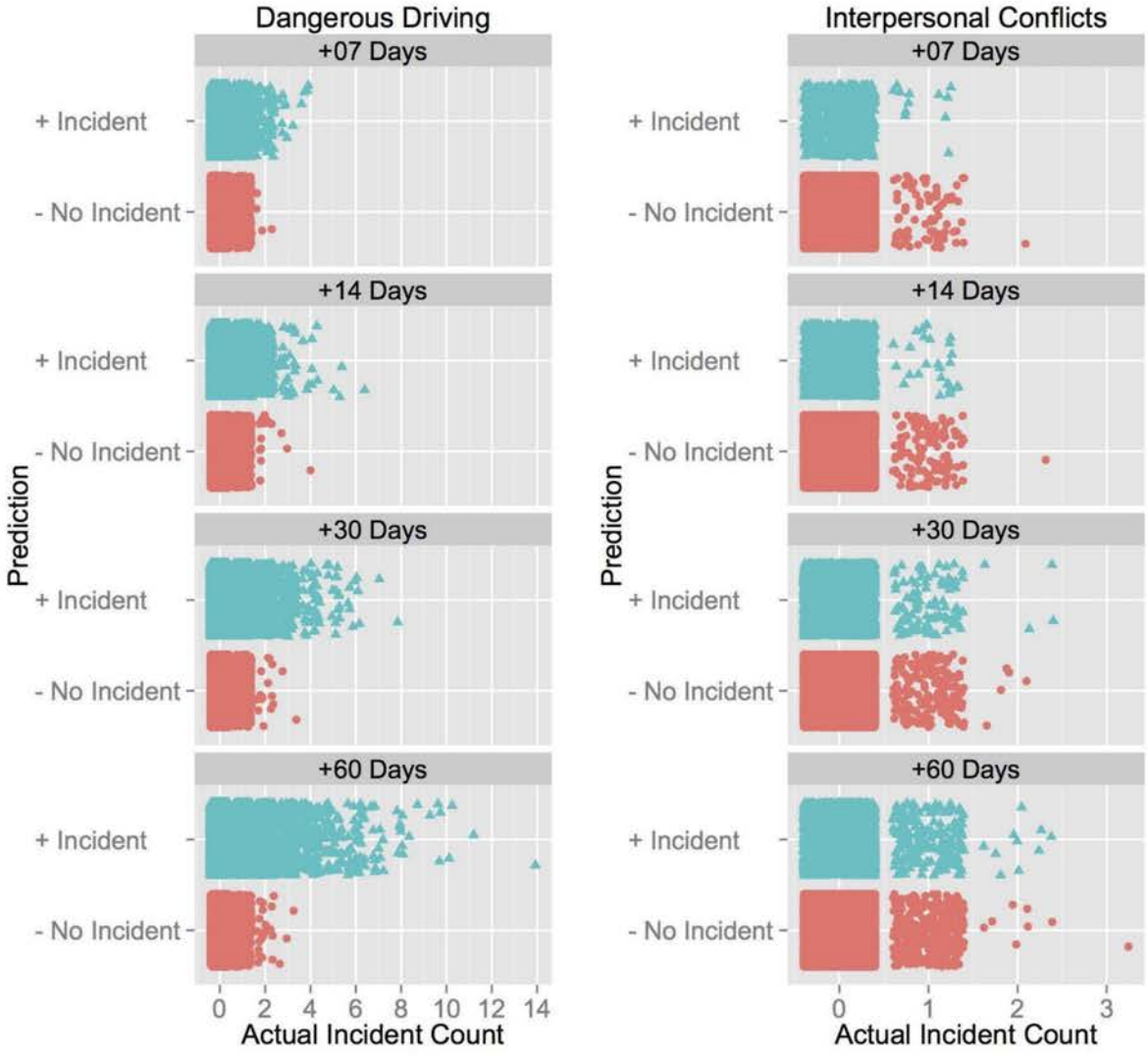
To dig deeper into the results, Figure 7 plots out-of-sample predictions generated for the days October 10, November 10,and December 10, 2015 on actual incident counts for dangerous driving and interpersonal conflicts by city (using the recall models that maximize the F_3 Score). Key takeaways:

- The dangerous driving model does not do well at predicting drivers with few actual incidents (1-2). **But it does very well at predicting incident counts greater or equal to**

3. In fact, it correctly anticipates nearly 100% of drivers with ≥3 tickets in the SF, LA, Chicago, and NYC test sets. These results are favorable, as it implies the model is good at anticipating the highest-risk drivers -- the ones we should care most about.

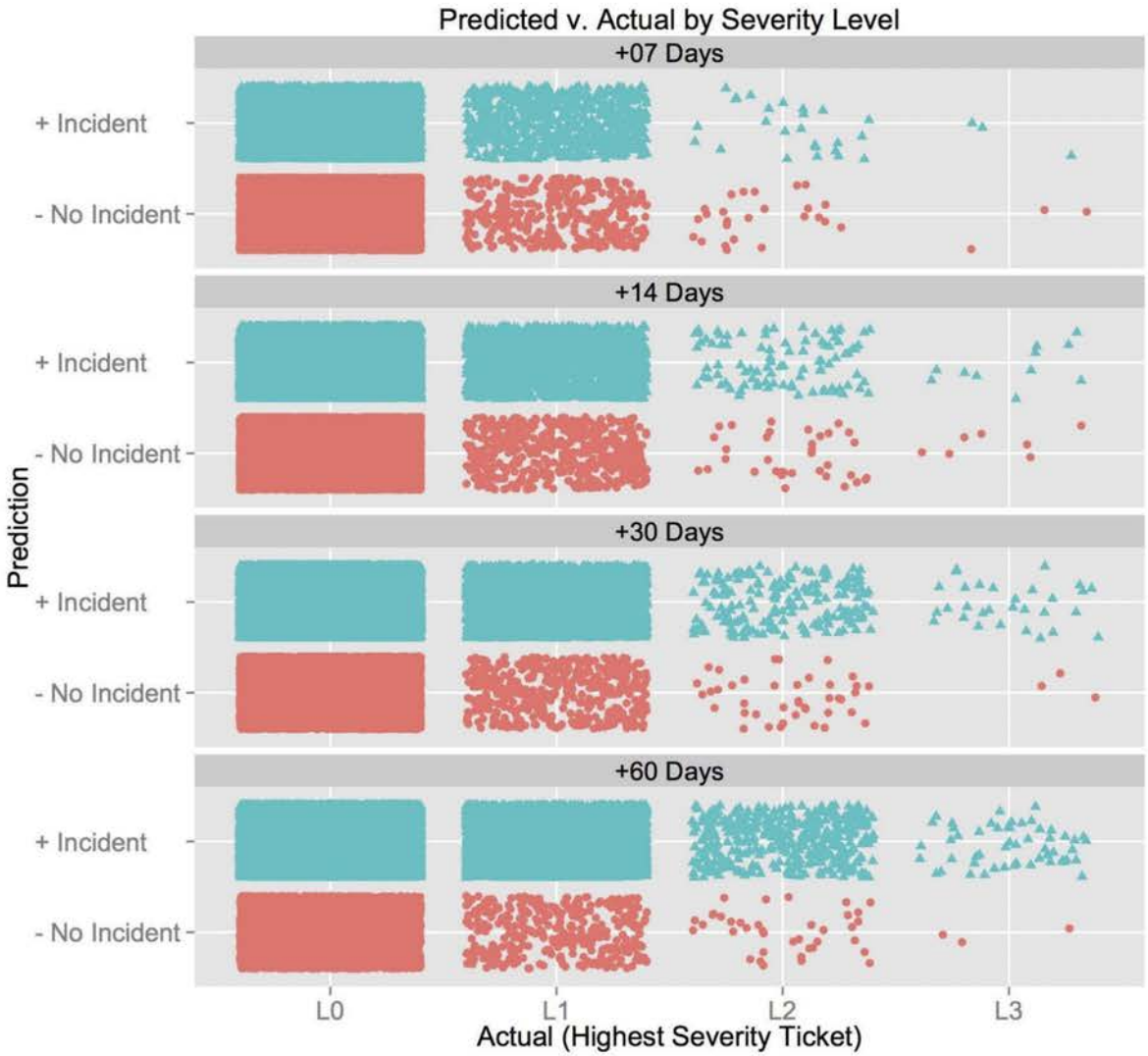
- Results for the driver interpersonal conflict models are not as strong.

Figure 7: Predicted v. Actual Incident Count



Notes: Figures show out-of-sample predictions from October 10, November 10,and December 10, 2015 (y-axis) and actual incident counts (x-axis) for dangerous driving (left) and interpersonal conflicts (right) at the 7, 14, 30, and 60 day forecasting windows for all cities in sample using model specifications that maximize the F_3 Score.

Figure 8: Predicted v. Actual by Severity Level



Notes: Figures show out-of-sample predictions from October 10, November 10,and December 10, 2015 (y-axis) and actual incidents by highest severity level (x-axis) at the 7, 14, 30, and 60 day forecasting windows for all cities in sample using model specifications that maximize the F_3 Score.

Table 1: Performance of Naive Forecasting Strategies

	Naive Strategy 1 (+ if + last period)			*Naive Strategy 2* (+ if + in lifetime)		
	Precision	Recall	Positive Percent	Precision	Recall	Positive Percent
Dangerous Driving						
Next 7 days	11.02%	11.49%	2.71%	4.83%	72.03%	38.94%
Next 14 days	17.23%	18.47%	4.87%	8.85%	71.09%	38.94%
Next 30 days	26.20%	29.86%	8.35%	16.23%	69.46%	38.94%
Next 60 days	33.23%	41.98%	10.86%	26.13%	67.36%	38.94%
Interpersonal Conflicts						
Next 7 days	1.19%	1.19%	0.19%	0.58%	23.48%	7.49%
Next 14 days	2.04%	2.04%	0.35%	1.15%	23.38%	7.49%
Next 30 days	3.79%	4.08%	0.67%	2.40%	23.25%	7.49%
Next 60 days	6.16%	7.69%	1.00%	4.48%	22.74%	7.49%

Comparison to Naive Forecasting Strategies

The predictive power of Bouncer can be further validated by comparing them against benchmarks set by naïve forecasting strategies that use simple heuristics and no model, such as:

- **Naive Strategy 1**: predict that a driver will cause at least 1 incident over the next t = {7, 14, 30, 60} days only if the driver committed an incident in the past t = {7, 14, 30, 60} days.
- **Naive Strategy 2**: predict that a driver will cause at least 1 incident over the next t = {7, 14, 30, 60} days only if the driver committed an incident in the past (lifetime).

Table 1 presents the precision, recall, and positive percent metrics for the 2 naive forecasting strategies. For dangerous driving, **Naive Strategy 1** has recall ranging from 11-42% across the forecasting windows, with recall increasing in the size of the forecasting window. **Naive Strategy 2**, on the other hand, has recall ranging from 67-72%, with recall

decreasing in the size of the forecasting window. For driver instigated interpersonal conflicts, **Naive Strategy 1** has recall ranging from 1-8% across the forecasting windows, with recall increasing in the size of the forecasting window. **Naive Strategy 2** has recall of ~23%, with recall decreasing in the size of the forecasting window. These figures indicate that interpersonal conflicts are substantially harder to predict than dangerous driving.

When compared to the performance of Bouncer on the same out-of-sample test set, it is clear Bouncer offers significant improvements from the status quo (that is -- no predictive model), as Bouncer can be tuned to outperform both naive strategies on any performance metric.

Consider, for example, recall. Naive strategy 2 has the highest recall, ranging from 67-72% for dangerous driving predictions and ~23% for interpersonal conflict predictions. Bouncer can be tuned (e.g., F_5 Score models) to have recall > 80% for dangerous driving predictions and >30% for interpersonal conflict predictions while maintaining similar levels of precision (6-20% for dangerous driving; 1-4% for interpersonal conflicts).

On precision, the naive strategies do not come close to reaching the performance of Bouncer, which can be tuned (e.g., $F_{0.01}$ Score models) to have > 85% precision at the 30/60 day forecasting window, and >35% at the 7/14 day forecasting window for both dangerous driving and interpersonal conflict predictions. Naive strategy 1 has higher precision than naive strategy 2, and it produces precision ranging from 11-33% for dangerous driving predictions and 1-6% for interpersonal conflict predictions.

Altogether the results demonstrate that Bouncer provides actionable intelligence that would otherwise be unavailable.

5. Improvements From Previous Models [back to contents]

With every iteration of Bouncer, it is a useful exercise to assess performance improvements to get a sense of whether the iterations are being made in the right direction and inform next steps. To do this, I compare the predictive performance of 3 different versions of Bouncer:

1. **Michelangelo feature set**
    a. Features in HDFS only. The most basic predictors, like trip counts, cancellation rates, ETAs, ratings, safety ticket history, vehicle type.
2. **Version 2**
    a. Michelangelo feature set + average of first N ratings + features based on attitude tickets.
3. **Version 3 (current)**
    a. Version 2 features + telematics + weather + text analysis on feedback.

* See Bouncer Variable List for full list of features and definitions.

I estimate the performance of each model by computing the Area Under the Receiver Operating Characteristic Curve (AUROC), which plots the sensitivity (e.g., recall / true positive rate) of the models against their false positive rate, which is defined as 1 minus their specificity (true negative rate):

$$Sensitivity = Recall = True\ Positive\ Rate = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$Specificity = True\ Negative\ Rate = \frac{True\ Negatives}{False\ Positives + True\ Negatives}$$

$$False\ Positive\ Rate = 1 - Specificity = \frac{False\ Positives}{False\ Positives + True\ Negatives}$$

Generally, the AUROC curve is computed using predictions made against the out-of-sample test set. However, because the out-of-sample test set is highly skewed due to the rarity of safety incidents, meaningful performance improvements will not be as detectable in the test set AUROC curve.[3] As such, I use cross-validation AUROC to compare model performance (since training set is balanced). For completeness, though, Table A.2 in the Appendix replicates the exercise for the test set. Results are consistent.

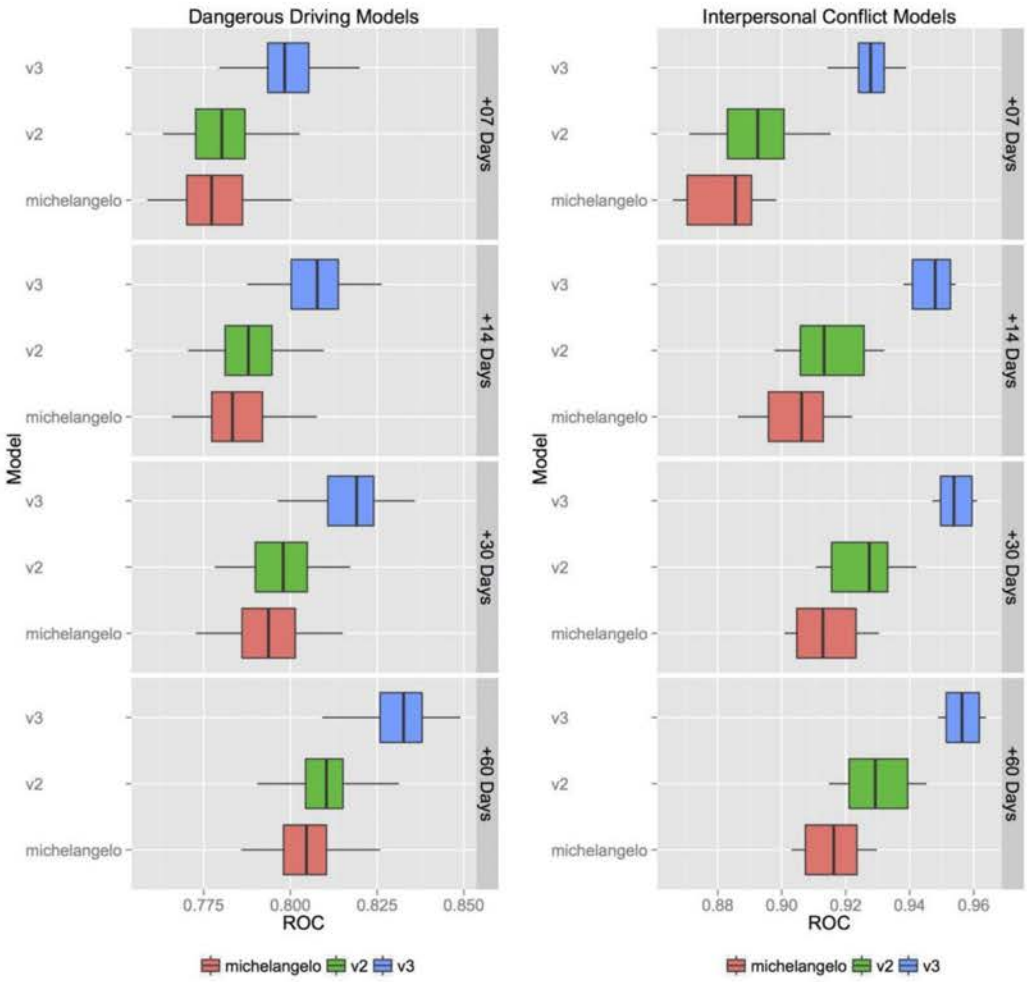
Figure 9 provides box plots describing the area under AUROC curves for the Michelangelo, v2, and v3 models separately, by response variable and forecasting window. Because separate models are trained for each city, the box plots illustrate the distribution of AUROC

[3] See Davis and Goadrich (2006) at http://www.autonlab.org/icml_documents/camera-ready/030_The_Relationship_Bet.pdf.

metrics for each model type (25th, 50th - median, 75th percentiles, with whiskers giving 95% confidence intervals).

The results reveal that **the v3 model performs significantly better than the v2 model, which performs better than the Michelangelo model**. The v3 model has a median AUROC that is 0.02-0.03 greater than the Michelangelo model when predicting dangerous driving, and a median AUROC that is ~0.04 greater than the Michelangelo model when predicting interpersonal conflicts. The median AUROC of Michelangelo is ~0.80 for dangerous driving and ~0.90 for interpersonal conflicts. So improvements of 0.02-0.04 represent quite major improvements.

Figure 9: Comparing Model Performance (Cross-Validation AUROC)



Notes: Box plots of area under the AUROC curve for all models, aggregating city-specific results for each model type (version x response x forecasting window). Box plots provide 25th, 50th (median), and 75th percentiles with whiskers extending to 95% confidence intervals.

6. Most Important Predictors [back to contents]

To identify the most important predictors and risk factors, I compute variable importance scores by systematically permuting each variable and calculating the change in classification accuracy -- as defined by changes in the generalized cross-validation (GCV) estimate of error. I compute these scores for each algorithm and each type of safety incident separately, and then use the median variable importance score across all cities and forecasting windows to rank each variable's relative importance for forecasting dangerous driving and interpersonal conflicts.

Table 2 lists the 50 predictors identified to have the greatest predictive power using this procedure. See the Bouncer Variable List for variable definitions. Figure 10 depicts these results as box plots to facilitate visual exploration of relative importance. Broadly speaking, the features with the most predictive power are those based on:

- Cancellation percent.
- Previous safety and attitude related complaints from riders (e.g., support tickets with issue type attitude, dangerous driving, inappropriate behavior).
- Ratings and rider feedback.
- Telematics data (e.g., hard brakes + accelerations, speed).
- Trip count / duration on trips / duration during high risk times.
- Weather (e.g., precipitation, snow).

It is important to note here that -- although aggregating the city-specific rankings makes the results more digestible -- it also masks interesting variation across cities (e.g., relative impact of weather on safety incidents in New York City versus Los Angeles). The box plots in Figure 10 provide a sense of the spread in the distribution of variable importance scores.

Table 2: Top 50 Most Important Predictors

Rank	Dangerous Driving	Interpersonal Conflict
1	num_dp_cm	perc1star_last20
2	perc5star_cm	days_w_precipitation_next_60
3	trip_count_lastmonth	num_attitude_cm
4	rating_cm	precipitation_next_30
5	perc1star_last20	night_perc_lastweek
6	L1_cm	perc1star_cm
7	trip_count_lastweek	temperature_next_60
8	weeks_active	trip_count_lastmonth
9	telematics_p90_accel_avg_cm	rating_cm
10	telematics_p90_brake_peak_cm	weeks_active
11	perc1star_cm	perc_trips_w_comments_cm
12	L1_lastmonth	night_perc_lastmonth

13	temperature_next_60	rating_first_50
14	telematics_median_accel_avg_cm	telematics_median_brake_avg_cm
15	duration_lastweek	telematics_p90_accel_avg_cm
16	telematics_median_brake_avg_cm	perc5star_cm
17	trip_per_week	telematics_median_brake_peak_cm
18	avg_speed_cm	workday_perc
19	night_perc	day_perc
20	trip_count_cm	rating_given_cm
21	num_attitude_cm	telematics_p90_accel_peak_cm
22	distance_cm	eta_diff_cm
23	rating_first_50	trip_per_week
24	perc_trips_w_comments_cm	telematics_median_accel_peak_cm
25	p90_speed_cm	cancellations_perc_cm
26	telematics_median_accel_peak_cm	telematics_median_accel_avg_cm
27	city_driver_ipc_mom	night_perc
28	night_perc_lastweek	num_behavior_cm
29	day_perc	perc_offensive_cm
30	workday_perc	p90_speed_cm
31	num_attitude_lastmonth	avg_speed_cm
32	telematics_median_brake_peak_cm	rating_first_25
33	cancellations_perc_cm	trip_count_lastweek
34	eta_diff_cm	duration_lastweek
35	rating_given_last50_lastmonth	weekend_perc
36	telematics_p90_accel_peak_cm	perc_super_positive_cm
37	rating_first_25	distance_cm
38	num_super_positive_cm_lastmonth	telematics_p90_brake_peak_cm
39	spd_dangerous_cm	duration_cm
40	weekend_perc	num_attitude_lastmonth
41	cancellations_cm	rating_given_last50_lastmonth
42	rating_given_cm	num_super_positive_cm_lastmonth
43	precipitation_next_60	cancellations_cm
44	perc_super_positive_cm	rating_first_10
45	duration_lastweek_d1m	night_perc_lastmonth_d2m
46	trip_count_lastweek_d1m	num_super_positive_cm_lastmonth_d2m
47	night_perc_lastmonth	trip_count_cm
48	duration_cm	precipitation_next_60
49	snow_next_7	duration_lastmonth_d2m
50	night_perc_lastmonth_d2m	vehicle_age

Figure 10: Box plot of Variable Importance Scores for Top 50 Predictors

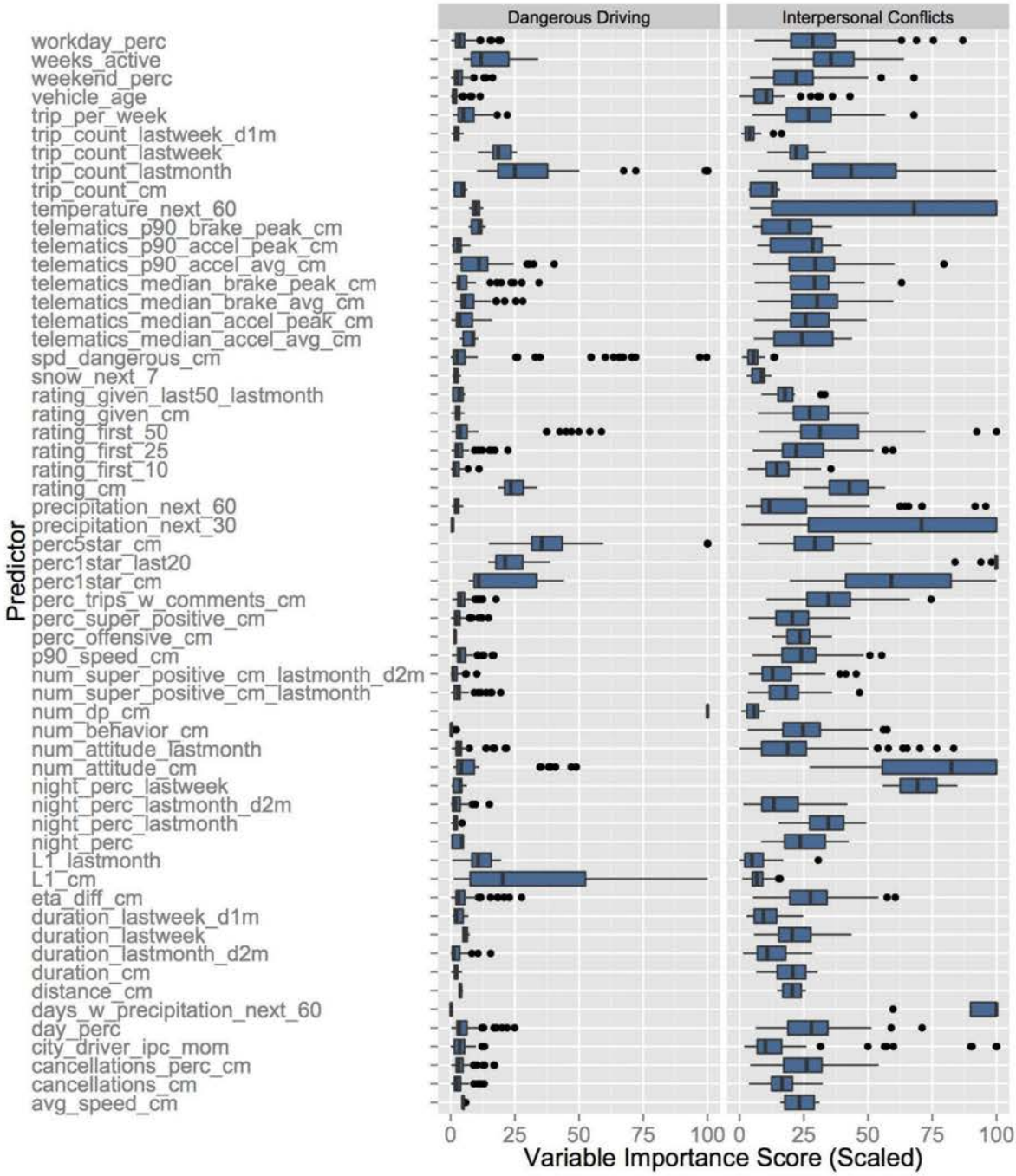


Table 3: Bouncer Development and Roll-Out Schedule

	Q4 2015	Q1 2016	Q2 2016	Q3 2016
VERSION	v2	v3	v4	v5
TARGET	• Driver-instigated incidents			• Driver-instigated incidents • Rider-instigated incidents
DATA PIPELINE	• Trips • Safety tickets • Driver background	• Telematics • Weather • BGCs • Text analysis on feedback	• NLP on feedback • Hot spots • Referral networks • Event schedules • External	
COVERAGE	SF, LA, CHI, NYC	SF, LA, CHI, NYC	All US Cities	US, Europe, ANZ
ENGINEERING	v2 data pipeline	v2 data pipeline	v3 data pipeline	v4 data pipeline
KPIs	• Safety incidents (counts + rates) • Lawsuits / settlements (number + cost) • Predictive performance (precision + recall)			

7. Challenges [back to contents]

- Data integrity — lots of measurement error in safety data (e.g., Zendesk ticket classifications).
- Safety incidents are rare events (e.g., <0.10% of observations), creating highly imbalanced data.
- Model training process very slow despite leveraging parallel processing.

8. Next Steps [back to contents]

- Productionize predictive models using Michelangelo.
- Error analysis. Figure out why some high risk users are being incorrectly classified as low risk and why some low risk users are being incorrectly classified as high risk. Then build new features to capture this variation.
- Maximize precision while holding recall as close to 100% as possible. Strategies:
    - Increase sample size.
    - Build better features that distinguish safe users from unsafe users.
    - Better tuning parameters.

    - In model training, increase costs for misclassification of positive cases (i.e., cost sensitive learning).
- Build features for Bouncer v4 (features listed on this variable list).
- Develop models for other cities outside of US. Need to figure out if there's variation in predictive performance across cities, and if so, what explains the variation (e.g., young v. mature market, regional/cultural differences, etc.)
- Build models for predicting **rider instigated interpersonal conflicts** -- one of the most common causes of L3 and L4 safety incidents.
- Figure out how best to operationalize Bouncer's predictions. Lots of potential interventions. Need to test them using randomized experiments to identify the most effective strategies.
- Table 3 describes the Bouncer development and roll-out plan.

9. Team [back to contents]

Data: Sunny Jeon, Spencer Boucher, David Purdy
Engineering: Rami Mawas, Yisheng Liang, Ron Tal, Jeremy Hermann
Safety Ops: Jeana Williams, Becky Mar
Product: Dhruv Tyagi, Dima Kovalev
Legal: Justin Suhr, Seth Schreiberg

Appendix [back to contents]

Figure A.1: Predicted Probability versus Actual Incident Count

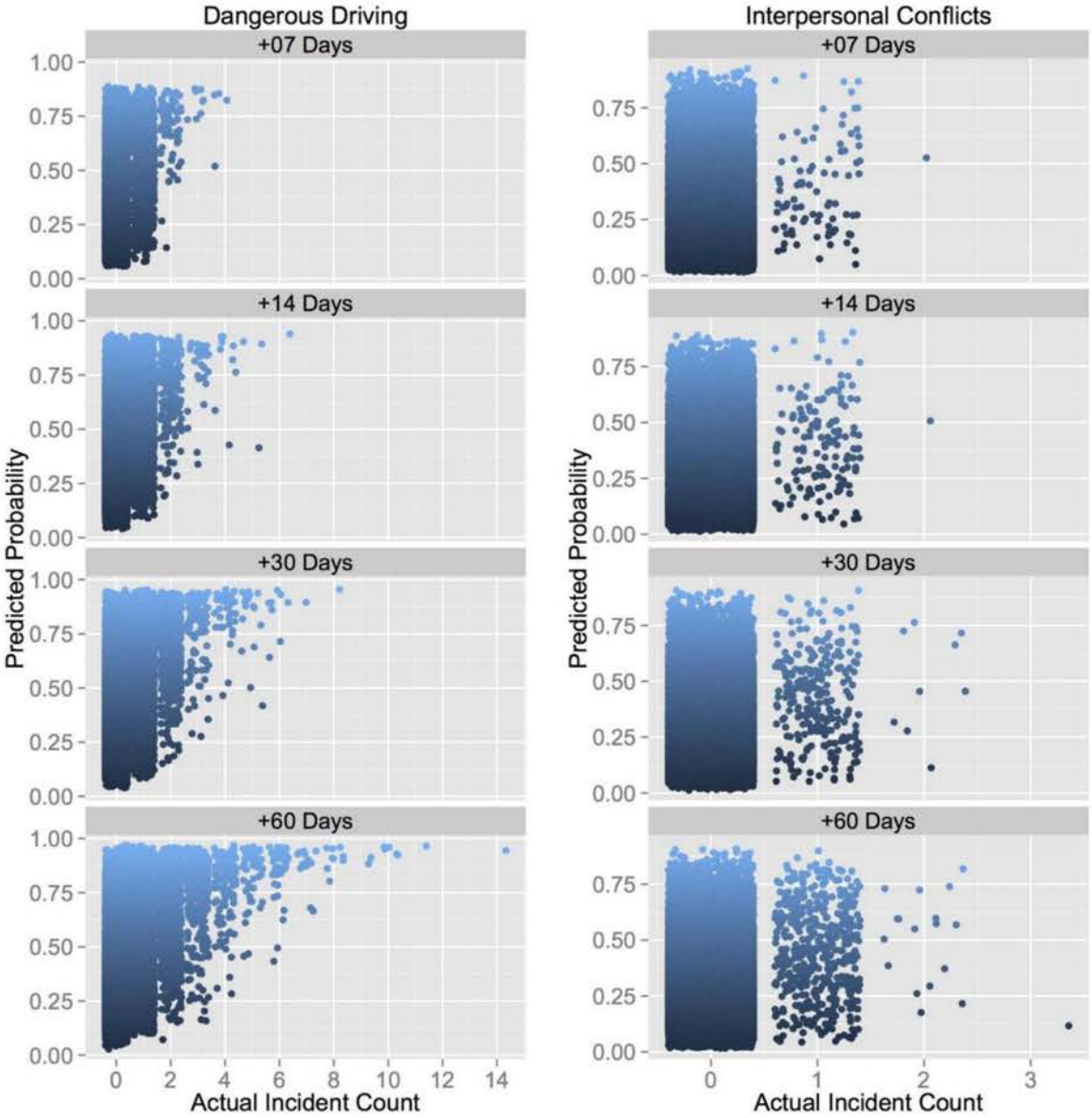


Figure A.2: Comparing Model Performance (Test Set AUROC)