Franklin Wang

Professor Luetgert

DIGS 30009

12 December 2025

Modeling Policing Data with Machine Learning

Introduction

Machine learning is often presented as a tool that can extract patterns from large datasets and produce accurate predictions. In many professional settings, this promise is demonstrated using datasets that are already well structured, mostly numeric, and designed for modeling. These examples make machine learning appear cleaner and more reliable than it often is in practice. However, administrative data rarely looks like this in real life. Policing data is a clear example. It is created to support routine police work rather than predictive analysis, and it mainly reflects internal practices for how incidents are recorded and classified within the department.

Data Description

Data Source and Concerns

The National Repository of Policing Data (NRPD) is a data initiative that collects police event records from different police departments, including the Chicago Police Department. Event data describes individual police incidents, such as when police were notified of an event, what type of incident was reported, and how the police responded. The NRPD was created to improve access to policing data, which is often fragmented or difficult for the public to obtain.

For my final project, I use a sampled version of the 2018 NRPD event data from Chicago to move beyond standard classroom machine learning datasets and work with a real administrative dataset. The full NRPD event data is large, so sampling makes it easier to explore

patterns and see how different models behave within the scope of a course project. Rather than prioritizing predictive accuracy, I focus on the structure of the data. I also examine how recording practices affect what machine learning models are able to learn.

After defining the scope of the project, it is important to consider what kind of data this analysis is actually working with. The NRPD event data was not originally created for machine learning or research. Consequently, the dataset carries assumptions and limitations that directly affect how machine learning models behave, including technical and ethical concerns.

On the one hand, technical concerns mean that some fields are missing, and others are recorded inconsistently across events. Timestamps are not always reliable, and many variables are categorical rather than numeric. This means the data contains less clear information for models to learn stable patterns from, leading to potentially noisy or low-performing results. On the other hand, ethical concerns relate to what the data actually captures. The event data reflects how police, at their own discretion, choose to record and label incidents. This reality implies models may learn patterns about police behavior instead of the incidents themselves. Because of these technical and ethical concerns, I first examined the data using descriptive statistics and visualizations to better understand what information was actually present and how it was distributed before applying any machine learning models.

Descriptive Statistics and Visualization

The original dataset includes 1,048,001 observations and 33 features before filtering. Even so, the dataset is large. Events are not evenly spread across police radio zones, and some zones appear much more often than others. This uneven distribution is important because patterns in the data may mostly reflect what happens in the busiest zones.

To understand how response patterns vary across these zones, I focused on INCIDENT_DELAY as the main target variable in later models. Incident delay is defined as the number of minutes between when an event was received by dispatch and when officers arrived on scene. Descriptive statistics and boxplots (see Figure A in the Appendix) show that most events have short delays, while a small number have much longer delays, which raises the average. Because of this skew, the median gives a better sense of a typical response time. When comparing incident delay across radio zones, there is no clear linear relationship between zone and delay, so linearity assumptions are weak and should be used with caution when applying linear regression.

Data Cleaning and Preparation

After inspecting the structure and distribution of the NRPD event data, I prepared the dataset for analysis by making several cleaning and preprocessing decisions. Because the data was not originally collected for machine learning, these choices involved tradeoffs between improving consistency and preserving as much information as possible.

I began by cutting the dataset down to a smaller group of variables that actually related to where events happened, what type of incident was recorded, and how long it took for officers to arrive. A lot of the original columns were just IDs or internal codes. Keeping them would not have added much to the analysis, so I left them out. That said, removing these fields also meant losing some context that might matter outside the scope of this project.

Some fields look numeric but function as labels, such as police radio zones and service beats. I treated these as categorical during inspection. For modeling, they had to be encoded so the models could use the information, which is a simplification of what the variables represent. This also highlighted that administrative data rarely works in its raw form without adjustments.

After this initial preparation, I removed any rows that were missing values for key variables used in the analysis. NaNs (Not a Number) appeared because different officers, procedures, and data-entry practices led to gaps or inconsistencies in the records. Negative or impossible values were also removed, as they clearly reflected recording errors. For instance, events with negative INCIDENT_DELAY values were dropped, since a response cannot occur before a call is received. In general, dropping rows with missing values helped clean up the dataset, but it removed a significant number of observations and may have affected some event types or zones more than others.

In addition, I kept extreme delay values and zeros in the dataset, even though they are not realistic. These cases reflect the true limitations and quirks of administrative data. Keeping them preserves the real structure of the dataset, but it also makes modeling more challenging, especially for machine learning algorithms that assume consistent patterns.

Finally, categorical fields such as FINAL_DISPATCH_DESCRIPTION were converted into dummy variables for modeling. While necessary for technical reasons, this process greatly increased the number of features and added many columns with mostly zeros, which could make the models harder to learn from and may introduce noise.

Methods

Baseline Models

I chose an unsupervised method to see how events in the NRPD data are grouped, and a supervised method to see whether incident delay could be predicted from the available features. Using these methods for different reasons helped me see what each could and could not show about the data.

K-means (Unsupervised Learning)

I used K-means clustering as an exploratory method to check whether the NRPD event data showed any clear groupings by radio zone and incident delay. I chose this approach because there was no good target variable to predict at this stage, and I wanted a basic sense of the data's structure before moving on to supervised models. K-means works by grouping events that are similar to each other based on Euclidean distance, which makes it useful for an initial scan of patterns in the data.

However, applying it to this dataset makes its assumptions visible. The method assumes that all features are numeric and comparable, that distance between points reflects meaningful similarity, and that clusters are relatively balanced. In this case, a radio zone number is not truly numeric but a categorical label, and encoding it numerically creates a false sense of distance between zones. Additionally, incident delay is highly skewed, with many short delays and a small number of extreme values. When the K-means model was applied, the resulting clusters were driven almost entirely by differences in delay length rather than by radio zone, with most events grouped into a large short-delay cluster and a smaller cluster capturing long delays. This suggests that while K-means is useful for highlighting how response time dominates the dataset, it is limited in its ability to reveal meaningful spatial structure in administrative policing data.

Linear regression (Supervised Learning)

To move from exploration to prediction, I then turned to linear regression to test whether incident delay could be predicted using the available features in the NRPD event data. I chose linear regression as a baseline model because it is simple, interpretable, and commonly used to check whether there is any clear relationship between predictors and a numeric outcome. The goal was not high accuracy but to see whether the data contained enough meaningful patterns to support basic prediction.

Linear regression makes several assumptions that are important for interpretation. It assumes a roughly linear relationship between predictors and the outcome, normally distributed errors, and limited multicollinearity among features. These assumptions do not hold well for this dataset. Incident delay is highly skewed, and earlier visualizations showed no clear linear relationship between radio zone and delay. In addition, many predictors were converted into dummy variables, which increases collinearity and makes coefficient estimates unstable. When the model was applied, it explained very little variation in incident delay and produced low overall performance.

Testing Assumptions

Before fitting the linear regression model, I looked at the distribution of incident delay to see whether it aligned with the assumptions of the method. Most events have very short delays, while a much smaller number have extremely long delays, creating a strongly right-skewed distribution. This does not fit the assumption that errors are roughly normally distributed. Instead, a small number of extreme delays have a large influence on the model, which already limits how well a linear approach can work for this outcome.

After fitting the model, I examined a scatterplot (see Figure B in the Appendix) comparing actual incident delay to the values predicted by the model. In this plot, predictions remain clustered at low values even when the actual delay is longer, showing that the model does not adjust well to extreme cases. I also noticed that many predictors were converted into dummy variables, which creates overlap among predictors and leads to unstable coefficient estimates.

Alternative Methods

As a result, linear regression struggles to capture meaningful patterns in this dataset, especially for longer delays. To see whether models with fewer assumptions could perform

better, I tested random forest and decision tree regression as alternative supervised approaches. These models were chosen to explore whether added flexibility could help capture patterns that a linear model could not.

Random Forest (Supervised Learning)

I first applied a random forest model. Unlike linear regression, random forest does not force the data to follow one overall linear pattern, which seemed useful given how uneven incident delay looked in earlier plots. Based on this, I expected it might pick up patterns that the linear model missed.

In fact, the random forest model did perform slightly better than linear regression, but the improvement was limited. It handled some variation across events more effectively, but it still struggled with very long delays. Most predictions stayed close to shorter delay values, even when the true delay was much larger. This shows that although random forest allows for more flexibility, the available variables still do not provide enough information to predict extreme delays reliably.

Decision Tree (Supervised Learning)

I tried a decision tree regression model as another option. I picked this model because it breaks the data into smaller pieces based on feature values, instead of trying to fit one overall pattern. Since incident delay looked very uneven across events, I wanted to see if a decision tree could handle those differences more directly.

However, the decision tree did not work very well in practice. It fit some parts of the data closely, but the results changed a lot depending on how the data was split. The model was especially unreliable for longer delays and did not generalize well beyond the training data.

Compared to the random forest, the single decision tree was more affected by noise and did not offer much improvement for this dataset.

Why I Tried Other Methods

In my machine learning application, I used both an alternative method and a layered modeling approach. The decision tree serves as an alternative to linear regression because it does not rely on a single linear relationship and instead allows different patterns across the data. The random forest builds on this approach in a layered way by combining many decision trees to produce more stable predictions. My justification for choosing these models was to test whether moving away from the strict structure of linear regression would better fit the uneven and skewed distribution of incident delay. The main advantage of these approaches is their ability to handle nonlinear patterns and variation across events, which helps clarify whether weak performance comes from model choice or from limitations in the data itself.

Result Discussion

One result that genuinely surprised me in this final project is how similar the supervised models perform, despite their very different structures. Based on intuition, I expected random forest to clearly outperform linear regression, since it can model nonlinear relationships and interactions. Instead, the test R^2 values are almost the same. Linear regression reaches about 0.18, while random forest slightly drops to 0.16, with a higher test MSE (see Table A in the Appendix). This means that adding complexity did not improve generalization and in some cases made it worse.

What stood out even more to me is that random forest performs slightly better on the training data but not on the test data. This gap shows that the model is learning small patterns that do not carry over to unseen events. In other words, the additional flexibility helps fit the

training set but does not capture anything stable about incident delay. Seeing this made it clear that the issue is not underfitting, but that the features themselves do not contain much useful information for predicting incident delay.

The decision tree results reinforce this conclusion. The decision tree has the lowest test R^2 at around 0.11 and the highest test error. This model should be able to separate out extreme cases by splitting the data into smaller regions, yet it performs the worst. It suggests that the extreme delays are not driven by consistent feature combinations, but by irregularities in the data such as missing or repeated timestamps. The tree ends up chasing these irregularities instead of learning meaningful structure.

What I found most striking is that all three models fail in different ways but arrive at nearly the same outcome. Linear regression averages everything toward short delays. Random forest smooths predictions toward the mean. The decision tree overreacts to noise. Despite these differences, none of them can explain more than about 15–18% of the variation in incident delay. Seeing this same issue across models was unexpected and showed me that the data, not the model choice, is the main limitation.

Rather than identifying a “best” model in the traditional sense for prediction, these results show something more important. The variables available in the NRPD event data are not sufficient for predicting response time, even when using more advanced methods. The models are doing exactly what they are designed to do, and their failure is itself a meaningful result. This was frustrating at first, but it became the most important takeaway of my project.

Criticism and Outlook

Identifying Why the Models Fail

I believe the main problem of my final project is not in the implementation of the methods, but in the structure and content of the data itself. The sampled NRPD event data was not created for predictive analysis, and this becomes clear when trying to model incident delay. Key factors that likely drive response time, such as distance to the scene, traffic conditions, officer availability, or dispatch priority, are not present. Thus, even correctly implemented models struggle to explain why some delays are longer than others.

From a technical perspective, the modeling process itself worked as expected. I did not encounter major runtime issues, crashes, or memory problems while working in Google Colab. However, I did run into practical challenges related to data preparation. Cleaning timestamps, handling missing values, and deciding how to treat zero or extreme delays required repeated checks and judgment calls. These steps did not break the analysis, but they showed how much the results depend on data-entry practices and administrative issues.

Criticism

One limitation of my final project approach is that I treated incident delay as a single, continuous outcome without fully accounting for how it is constructed in the dataset. Delays are calculated from two timestamps that are often incomplete or inconsistently recorded. While I discussed this issue and kept extreme values to preserve the structure of the data, I did not fully explore alternative ways of defining or cleaning the delay variable since I did not have enough time. This choice likely amplified noise and made prediction harder, especially for longer delays.

Another limitation is that most features I had are categorical labels rather than numeric measures. Radio zone, beat, and dispatch type describe how incidents are organized in police systems, not the real-world conditions officers face. Converting these labels into dummy variables allowed the models to run, but it also increased dimensionality and collinearity without

adding much explanatory power. I did not try grouping or simplifying the features further, which might have made broader patterns easier to see.

Methodologically, I focused on mostly standard models taught in class, which was appropriate for demonstrating understanding. But it also limited how deeply I could explore more alternative methods since this is my first time studying machine learning at such an advanced level. For example, I did not try models that are better suited for highly skewed data, or methods that treat extreme delays as a separate case. These choices mean that some structure in the data may remain unexplored.

Lastly, my analysis treats the 2018 NRPD event data largely at face value. Although I acknowledge known problems in policing data, I did not validate the delay values using external sources. As a result, it is unclear whether observed delays reflect real response behavior or issues in how the data was recorded. This limits how confidently any findings about delay can be interpreted as policing behavior rather than data quality issues.

Outlook

With more time and resources, my final project could be expanded into a larger study on what policing data can realistically support in terms of prediction and analysis. One clear improvement would be rethinking the outcome variable. Predicting raw incident delay proved difficult because delay values are often messy and unreliable. Many extreme and zero delays appear to come from data-entry issues rather than real response times. A larger project could focus on simpler outcomes or filter out cases with clearly incorrect timestamps. This would shift the analysis away from precise timing and toward more stable patterns. Another major direction would be feature enrichment. The variables used here mostly describe how incidents are recorded, not the conditions that shape police response. A thesis-level project could integrate

additional data sources. Even basic measures could help test whether the models fail because important information is missing, not because the methods are wrong.

Incorporating basic natural language processing (NLP), which is a set of methods for analyzing and comparing text, on dispatch data could also be interesting, especially using the initial dispatch descriptions and final disposition descriptions. In this project, I treated these fields as categorical labels and mostly ignored them, but they contain more information than that. The wording of an incident often changes from the initial dispatch to the final disposition as officers gather more information. A future study could compare the initial dispatch description with the final disposition using simple similarity scores to measure how much a call changes during the response process. Keyword analysis could show which terms appear more often at dispatch versus disposition, helping identify patterns where calls are escalated, downgraded, or reclassified. Even these basic NLP methods can capture how incidents are interpreted over time, which is information lost when text fields are reduced to dummy variables.

Conclusion

In the end, even with all the limits in the dataset, I still feel that my final project shows I understand how to apply machine learning. Working with the National Repository of Policing Data made me see how much missing information, strange values, and the way incidents are recorded affect what a model can actually learn. I realized that performance is not just about the algorithm; it is about the data itself. Predicting incident delay was harder than I expected because the timestamps were messy and many variables did not fully capture reality. Even with the more flexible models, the patterns I expected never really showed up. This project showed me that machine learning is not just about getting predictions to work. I now feel more confident in my coding and in making sense of messy, real-world data.

Appendix

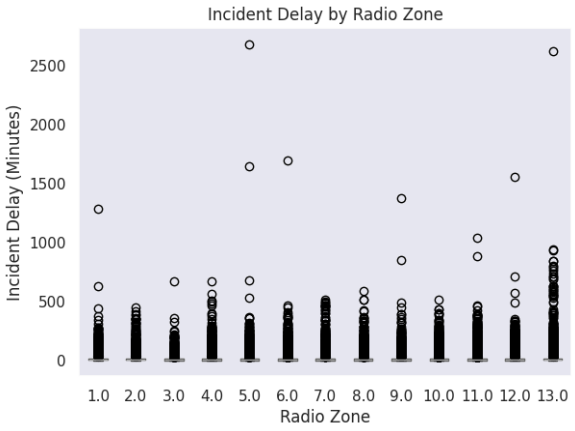

Figure A. Incident delay by radio zone.
Scatterplot of incident delay (minutes) across police radio zones.

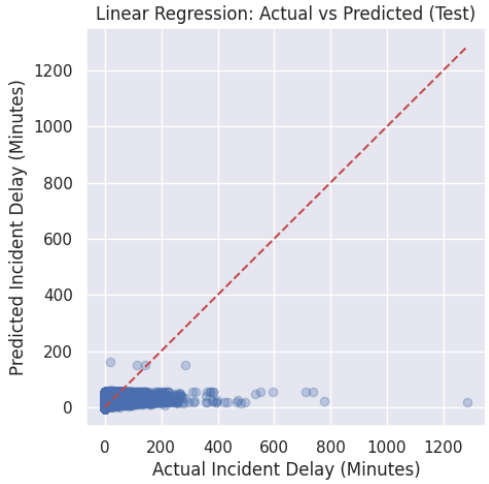

Figure B. Linear regression predicted vs actual incident delay (test set).
Predictions cluster near low delay values, indicating poor fit for extreme cases.

Model	Train R^2	Test R^2	Validation MSE	Test MSE
Linear Regression	0.16	0.18	448.42	525.54
Random Forest	0.17	0.16	463.44	536.49
Decision Tree	0.10	0.11	487.66	570.19

Table A. Model performance comparison on incident delay prediction.
All models show low and similar test performance across approaches.