**Report: Event Permit Prediction and Its Role in Traffic Management**

Table of Contents

[1. Introduction 2](#_Toc209297375)

[2. Data and Preparation 2](#_Toc209297376)

[a. Dataset Overview 2](#_Toc209297377)

[b. Feature Engineering 3](#_Toc209297378)

[c. Scaling and Encoding 4](#_Toc209297379)

[d. Correlation Check 4](#_Toc209297380)

[3. Weak Label Construction 5](#_Toc209297381)

[a. Method 5](#_Toc209297382)

[b. Outcome 5](#_Toc209297383)

[4. Model Training 6](#_Toc209297384)

[a. Setup 6](#_Toc209297385)

[b. Results 6](#_Toc209297386)

[c. Confusion Matrix 7](#_Toc209297387)

[d. Curves 7](#_Toc209297388)

[5. Key Findings 7](#_Toc209297389)

[6. Discussion and Application 8](#_Toc209297390)

[7. Limitations and Future Work 8](#_Toc209297391)

[8. Conclusion 9](#_Toc209297392)

# Introduction

Cities often face challenges when public events take place in busy areas. Weddings, film shoots, cultural festivals, and promotional events can block streets or attract large crowds, which disrupts traffic flow and creates congestion. Anticipating these disruptions is valuable not only for city authorities but also for individual road users. If a system can detect where and when an event is likely to happen, it becomes possible to plan detours, send alerts, and even recommend alternative transport options before congestion builds up.

The goal of this project was to develop a predictive framework using event permit data provided by the city. The dataset covers thousands of permits issued between 2014 and 2018 for various types of events across Melbourne. Our objective was to build and evaluate machine learning models that can identify patterns in the permits and predict the likelihood of an event happening on a given date and location. Beyond the technical task, the expected outcome is a system that can support real-world applications such as traffic management and proactive user notifications.

In this report, I will explain the steps we took in preparing and analyzing the data, the models we trained, and the insights we found. Each section connects back to the overall purpose: predicting event-driven disruptions so they can be managed more effectively.

# Data and Preparation

## Dataset Overview

The raw dataset contained 2,827 rows of event permits. Each row included details such as the event start and end date, the location, and categories describing the type of event. Some fields were incomplete, especially Category 2, which was missing in over 97% of rows. The Duration\_days variable, calculated from start and end dates, became one of the key numeric features. Most events lasted only one day, but some stretched to several months, creating a highly skewed distribution.

A graph of a number of different sizes and shapes

Description automatically generated with medium confidence A graph of a column

Description automatically generated

A graph of a bar graph

Description automatically generated A graph with text overlay

Description automatically generated

A graph of a number of events

Description automatically generated

The boxplot and histogram show that about 90% of events lasted one week or less, but a few outliers extended beyond 300 days. These long events, such as exhibitions, are rare but important signals because they indicate sustained disruption over a long period.

## Feature Engineering

We extracted several features from the event start date:

* Year, month, and day
* Day of week (0–6)
* Weekend flag (1 if Saturday or Sunday)

This allowed us to capture seasonal and weekly patterns. For example, weddings peaked in spring months, while promotional events were more spread across the year. Events were also more common on weekends than weekdays.

A graph of a graph of a number of columns

Description automatically generated with medium confidenceA graph of a bar graph

Description automatically generated

These plots confirmed strong seasonality, with a clear dip in winter and peaks in November and March. Year-to-year counts were fairly stable from 2014 to 2017, with only partial data in 2013 and 2018.

## Scaling and Encoding

The numeric features were standardized so that models relying on vector norms would not be biased by large values. After scaling, most durations clustered around zero with outliers far on the positive side, showing that scaling preserved the signal of long events.

A graph of a boxplot

Description automatically generated A graph of a column

Description automatically generated

Categorical features, namely location and Category 1, were one-hot encoded. Because there were hundreds of unique locations, we grouped rare values into an “infrequent” bucket to avoid an explosion of sparse features.

## Correlation Check

Since only one numeric feature was included, the heatmap was trivial with correlation of one. However, this step confirmed that the pipeline was selecting the right columns. With more engineered features, this step will help detect multicollinearity.

A graph of heatmap

Description automatically generated

# Weak Label Construction

One of the biggest challenges was that the dataset only contained positive examples (event days). To train a classifier, we also needed negative examples (days without events). Since explicit non-event labels were not provided, we created **weak labels**.

## Method

We used a combination of anomaly detection and rarity heuristics:

1. Isolation Forest was trained on the feature space and assigned anomaly scores. The top 5% most unusual rows were flagged as candidate negatives.
2. Rare combinations of category and location, and rare category-month counts, were also flagged as potential negatives.
3. If any method flagged a row as negative, we set its target to 0; otherwise, 1.
4. To avoid imbalance, the negative ratio was capped at 10%.

## Outcome

The final distribution was 2,545 positives (90%) and 282 negatives (10%). This ratio was balanced enough for models to learn without drowning in positives, while still realistic given how frequently events occur in city spaces.

A screen shot of a computer

Description automatically generated

This weak labeling process introduced some noise, but it created a starting point to train supervised classifiers. The expectation is that the models can still learn useful patterns, and later refinements can improve label quality.

# Model Training

## Setup

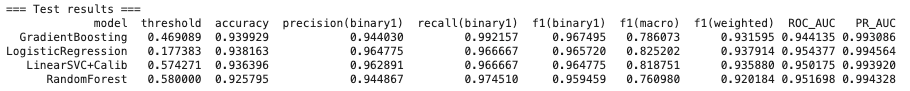
We used four different models to compare performance:

* Logistic Regression (with balanced class weights)
* Random Forest (600 trees, balanced subsample)
* Gradient Boosting
* Linear SVM with probability calibration

Each model was wrapped in a pipeline with preprocessing steps. Thresholds were tuned on validation splits to maximize F1 score for the positive class.

## Results

The test set contained 566 rows. Performance across all four models was strong, with F1 for the positive class above 0.95 in every case.



* Gradient Boosting achieved accuracy 0.940, precision 0.944, recall 0.992, and F1 0.967.
* Logistic Regression achieved accuracy 0.938, precision 0.965, recall 0.967, and F1 0.966, with the highest PR-AUC of 0.995.
* Linear SVM achieved F1 0.965 and PR-AUC 0.994.
* Random Forest achieved F1 0.959 and PR-AUC 0.994.

All models had PR-AUC above 0.99, showing excellent ranking ability between event and non-event days.

## Confusion Matrix

For Gradient Boosting, the confusion matrix showed 506 true positives, 26 true negatives, 4 false negatives, and 30 false positives.



This means the model almost never missed an event, but sometimes falsely predicted an event when there was none. In traffic applications, this is acceptable because a false alarm may cause some inconvenience, but missing a real event could lead to major disruption.

## Curves

The ROC curve had area 0.944 and the PR curve nearly flat near the top with precision above 0.94 even at high recall.

A graph of a curve

Description automatically generated A graph of a curve

Description automatically generated

This demonstrates that the models can operate across different thresholds depending on the tolerance for false alarms.

# Key Findings

The dataset showed that most events lasted only a single day, while a small number extended for weeks or months. These long events are rare but important because they create continuous disruption and stand out as strong signals for prediction.

There was also clear seasonality. Events peaked in spring, especially in November, and dropped in winter, with weddings and public non-ticketed events dominating the busy months. This pattern means that calendar features are valuable for anticipating periods of higher congestion.

Location proved to be another strong factor. A few gardens and public spaces, such as Fitzroy Gardens and Kings Domain, accounted for most permits. These hotspots highlight where disruption is most likely, while many other locations played only a minor role.

Finally, all models achieved very high precision-recall performance despite using weak labels. Gradient Boosting, Logistic Regression, Linear SVM, and Random Forest delivered similar results, showing that the signals in the data are strong. The models naturally favored high recall, rarely missing event days but producing some false positives. This trade-off is acceptable for traffic management, where avoiding missed events is more important than eliminating every false alarm.

# Discussion and Application

The outcome of this project goes beyond accuracy metrics. In practice, such a system can help city authorities and road users manage congestion. For example, if the model predicts a high likelihood of an event at Fitzroy Gardens on a Saturday, traffic controllers can set up detours in advance, and navigation apps can redirect drivers. Public transport agencies can increase capacity on routes near known event venues. Drivers approaching the CBD could receive alerts suggesting they take an alternative road or switch to a train or tram.

False positives, while not ideal, still provide useful information. If a route is flagged as possibly congested but turns out to be clear, the cost to the driver is minor compared to the cost of not warning about a real event. This makes high recall the right choice for threshold tuning.

The project also demonstrates how weak supervision can be used when explicit negative labels are unavailable. By combining anomaly detection and rarity heuristics, we created a balanced dataset that allowed standard classifiers to perform well. This approach can be applied to other domains where only positive examples are logged, such as fraud detection or medical rare-event prediction.

# Limitations and Future Work

There are limitations to this work. The weak labeling introduces noise, especially for rare combinations that may actually be valid events. The dataset also ends in early 2018, so the models may not capture new patterns after that. The split was random across years, which mixes past and future, so a time-based split would provide a more realistic test of deployment.

Future work could focus on:

* Improving label quality with manual audits or additional data sources.
* Adding richer features, such as weather, holidays, or population density near venues.
* Testing temporal splits to check model stability over time.
* Deploying the system as an API that can send alerts to users in real time.

# Conclusion

This project built a complete pipeline to process city event permit data, create weak labels, train machine learning models, and evaluate their ability to predict event-driven disruptions. The models achieved excellent results, with F1 above 0.95 and PR-AUC above 0.99 across all four methods. The findings show that even with weak supervision, meaningful and useful predictions are possible.

The practical value of the system is clear: predicting event days can support traffic management, user notifications, and smarter urban planning. While improvements are possible in labeling and temporal evaluation, the current pipeline already demonstrates strong potential to help reduce congestion and improve the experience of both drivers and event participants.