

---

---

# Parking Tickets in Chicago

— Predicting Payment —

---

---

# Overview

- Data from [ProPublica](#)
- City of Chicago Parking Tickets
- Passenger Vehicles only
- Potential to help allocate city resources
- Multiple decades and has over 50 million observations
- Only analyzing first million

# Data

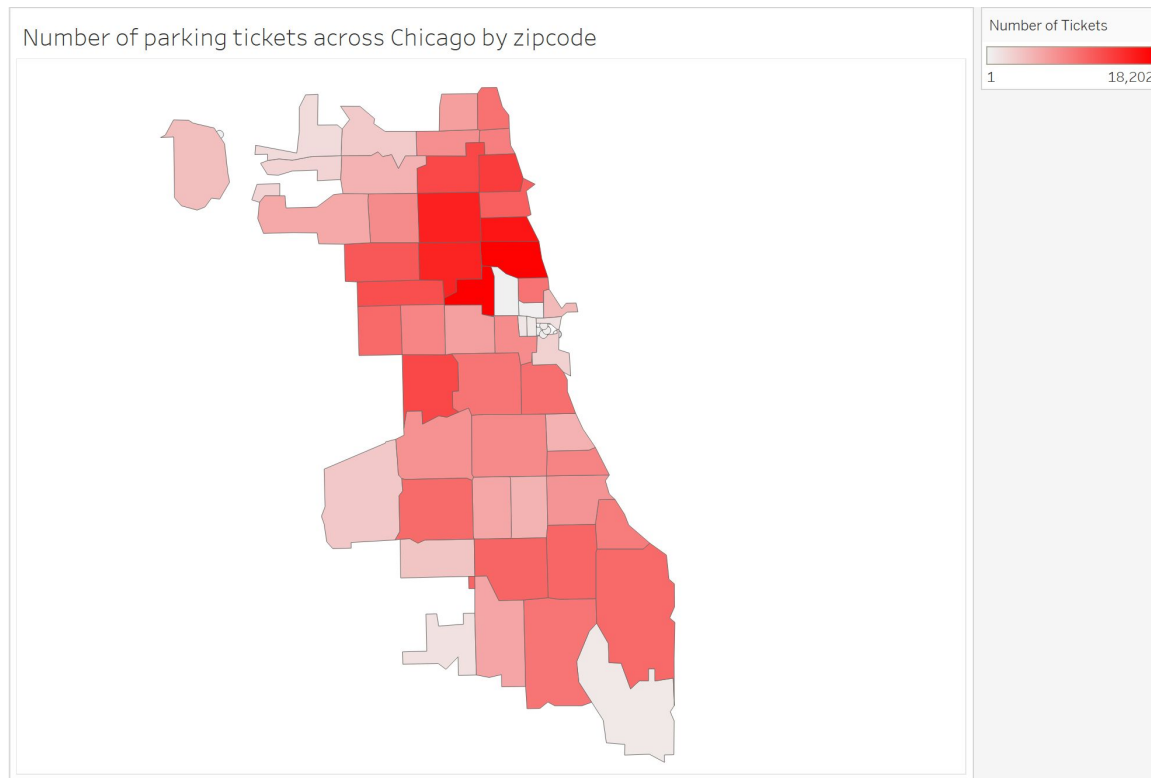
## Target: Payment Status

- Paid if paid
- Not Paid if: Dismissed, Unpaid, Hearing Required, Notice Sent, or Bankruptcy

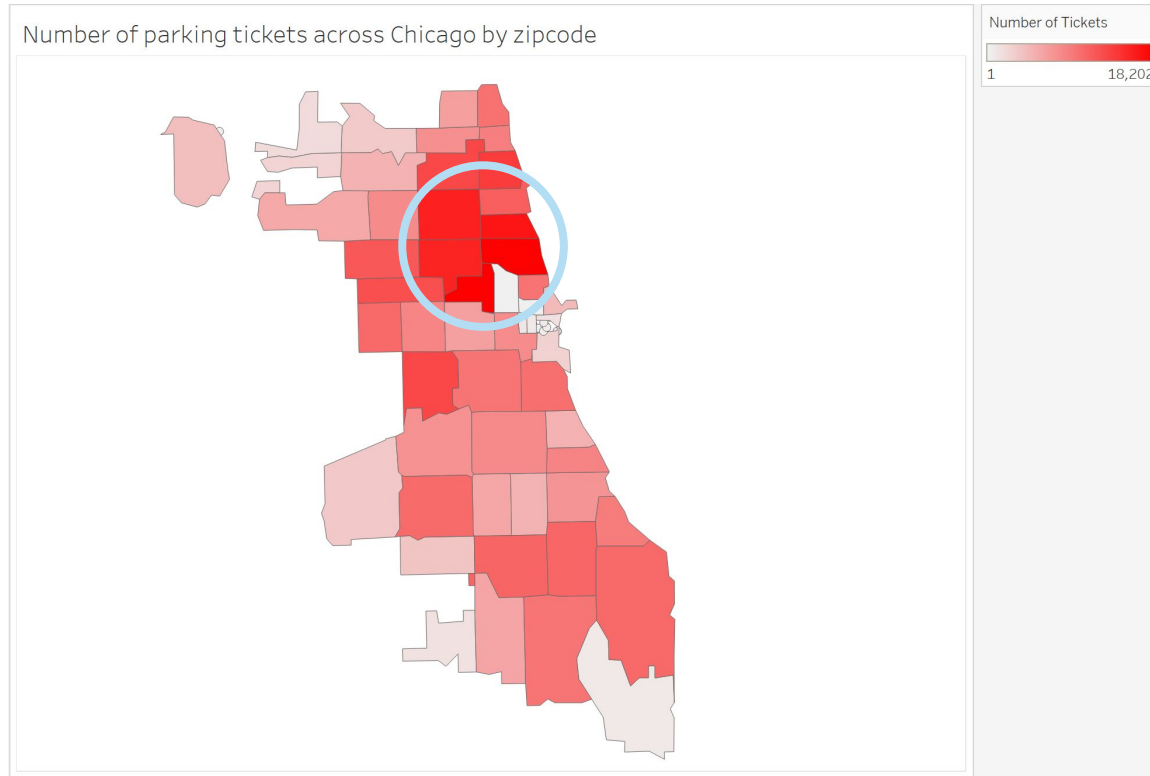
## Features

- License Plate State
- Geolocation (Latitude/Longitude)
- Fine amount (\$\$\$)
- Violation code (what is the ticket for?)
- Count of license plate appearing in data

# EDA

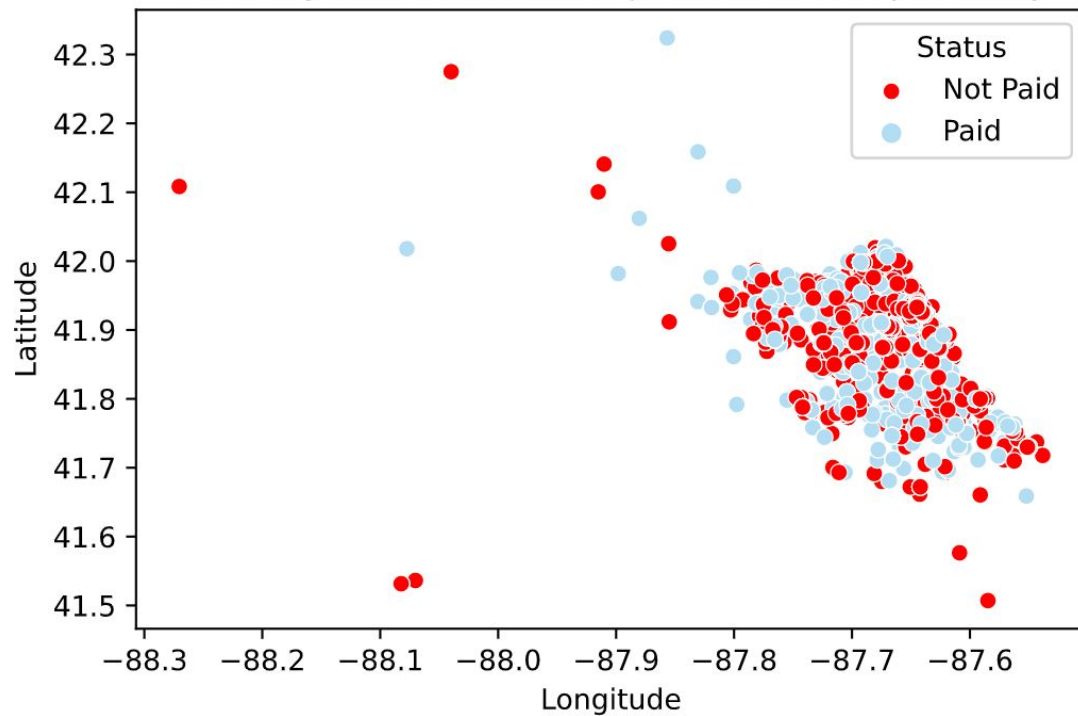


# EDA



# EDA

On a citywide scale, not a great deal of separability



# Thoughts on classifying and potential errors

The data set is imbalanced with paid tickets making up the majority (two to one)

The models in this project classify unpaid tickets as “positive”

# Thoughts on classifying and potential errors

The data set is imbalanced with paid tickets making up the majority (two to one)

The models in this project classify unpaid tickets as “positive”

**False Positives:** Tickets the model expects to be unpaid, but actually are paid

**False Negatives:** Tickets the model expects to be paid, but actually are unpaid



# Thoughts on classifying and potential errors

The data set is imbalanced with paid tickets making up the majority (two to one)

The models in this project classify unpaid tickets as “positive”

False Positives: Tickets the model expects to be unpaid, but actually are paid

False Negatives: Tickets the model expects to be paid, but actually are unpaid

**Goal: To separate paid and unpaid tickets as cleanly as possible**

**Metric: Use AUC score and confirm with confusion matrix**

# Baselining

- Simple Logistic models based on each feature for smaller data sample
  - AUC scores near 50%, not much better than coin flip
- Simple kNN models similar to above
  - Performance (speed) is terrible on larger data sets, not worth waiting
- Use more features in training logistic, Random Forest, and XGBoost

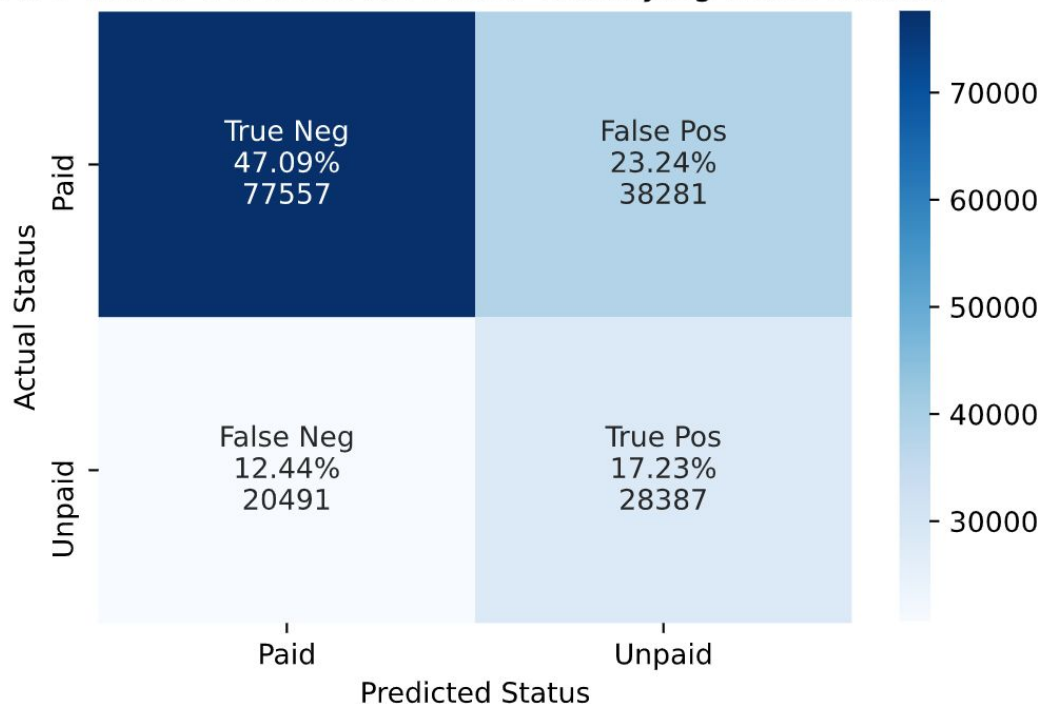
# The top performing models were

- 1) XGBoost
- 2) Random Forest

The top performing models both the full feature set

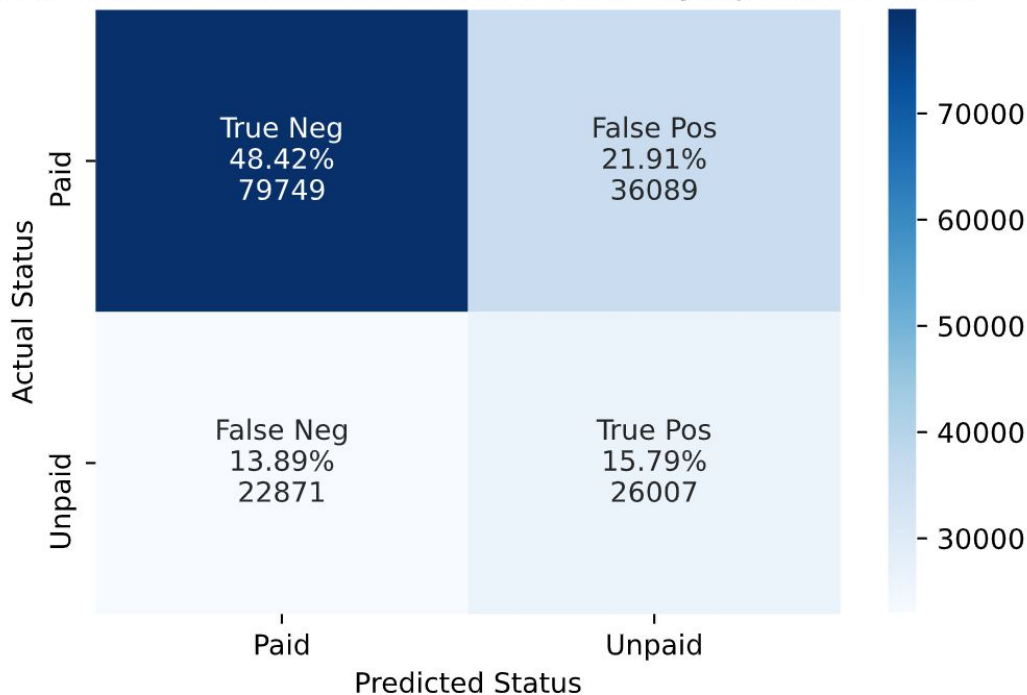
# Random Forest and XGBoost similar on unseen data

How well is the XGBoost model classifying ticket status?



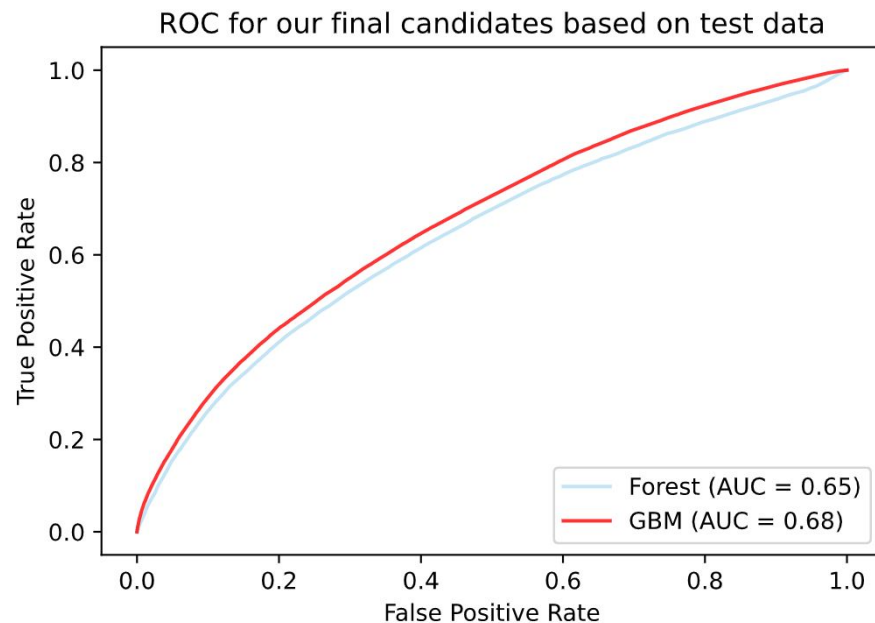
# Random Forest and XGBoost similar on unseen data

How well is the random forest model classifying ticket status?



# XGBoost generally outperforms Random Forest

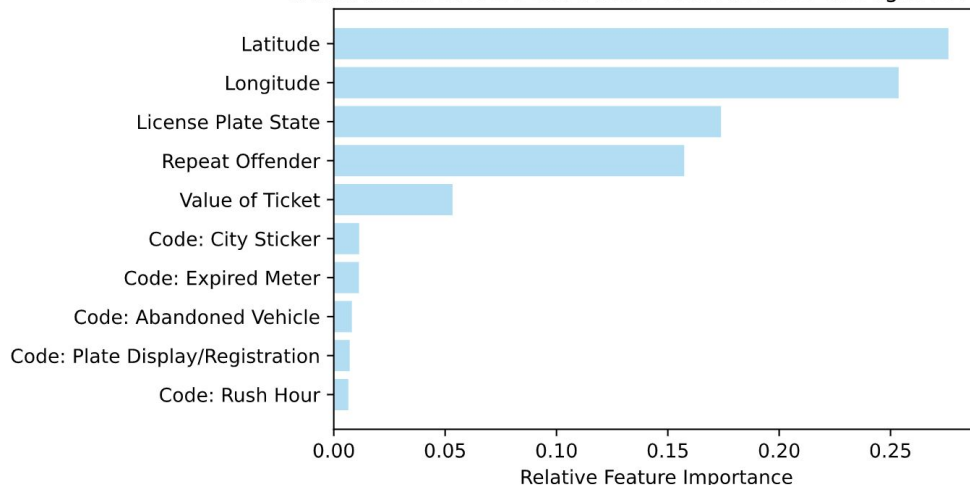
Score	Random Forest	XGBoost
AUC	.65	.68
Accuracy	.64	.64
Recall	.53	.58
Precision	.42	.43
F1	.47	.49



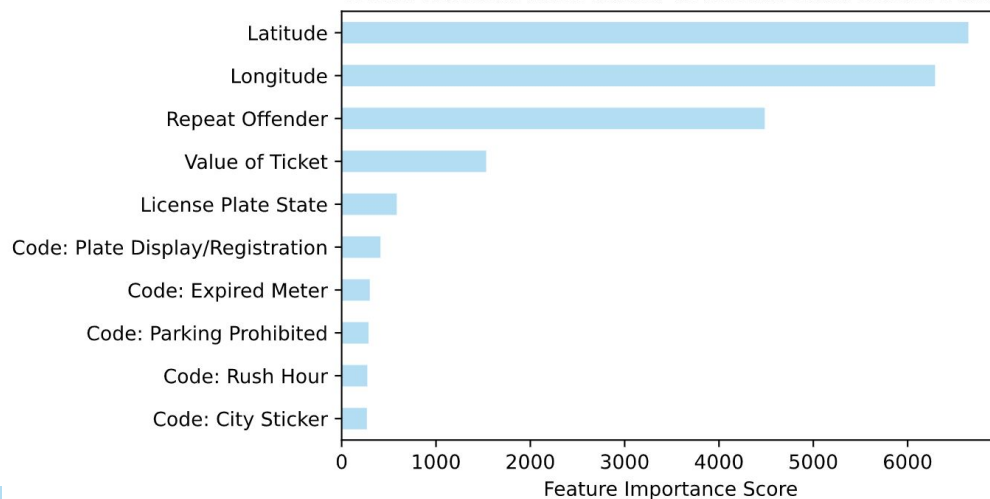
# Feature Importance

- General agreement across models
- Ticket location used most
- Many violations, importance spread disaggregated

Which features were used most often in trees throughout forest



Which features contributed to the XGBoost model most



# Future Work

Use cloud computing to handle full dataset



# Future Work

Use cloud computing to handle full dataset

**Continue tuning hyperparameters to improve the model performance**

# Future Work

Use cloud computing to handle full dataset

Continue tuning hyperparameters to improve the model performance

**Rather than paid or not paid, dig deeper:**

**Given these features, predicting if ticket goes to court**

**Given ticket goes to court court cases, predicting judgements**

**Given these features, predicting car seizure**

# Future Work

Use cloud computing to handle full dataset

Continue tuning hyperparameters to improve the model performance

Rather than paid or not paid, predict:

- Given these features, predicting if ticket goes to court

- Given ticket goes to court court cases, predicting judgements

- Given these features, predicting car seizure

**Create additional model(s) for red light ticket data**

# Future Work

Use cloud computing to handle full dataset

Continue tuning hyperparameters to improve the model performance

Rather than paid or not paid, predict:

- Given these features, predicting if ticket goes to court

- Given ticket goes to court court cases, predicting judgements

- Given these features, predicting car seizure

Create additional model(s) for red light ticket data

**Create an applet to allow for ticket info to be entered and output a prediction**

Thank you!

Questions?

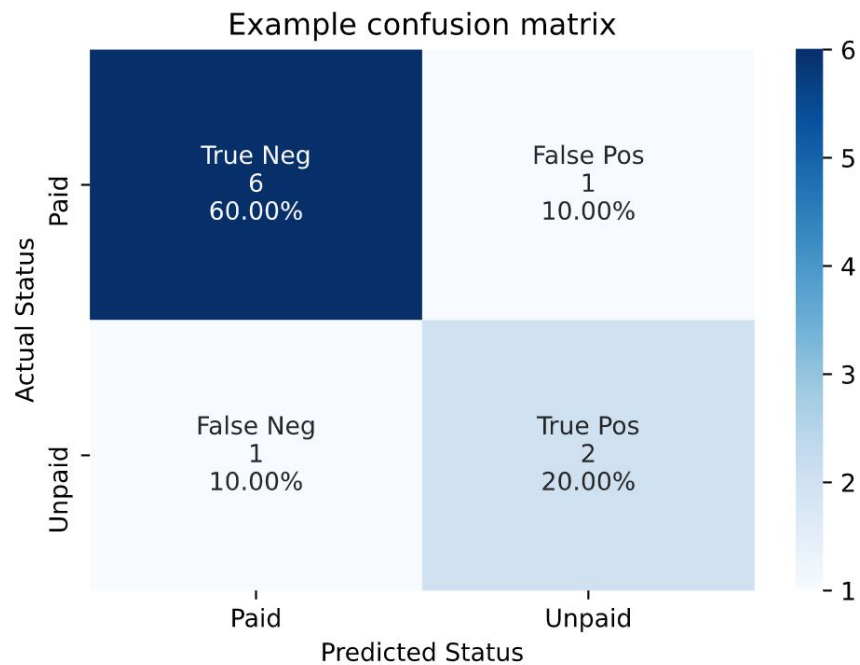
# Appendix

# Use cases

Predict whether a given parking ticket will be paid or not paid

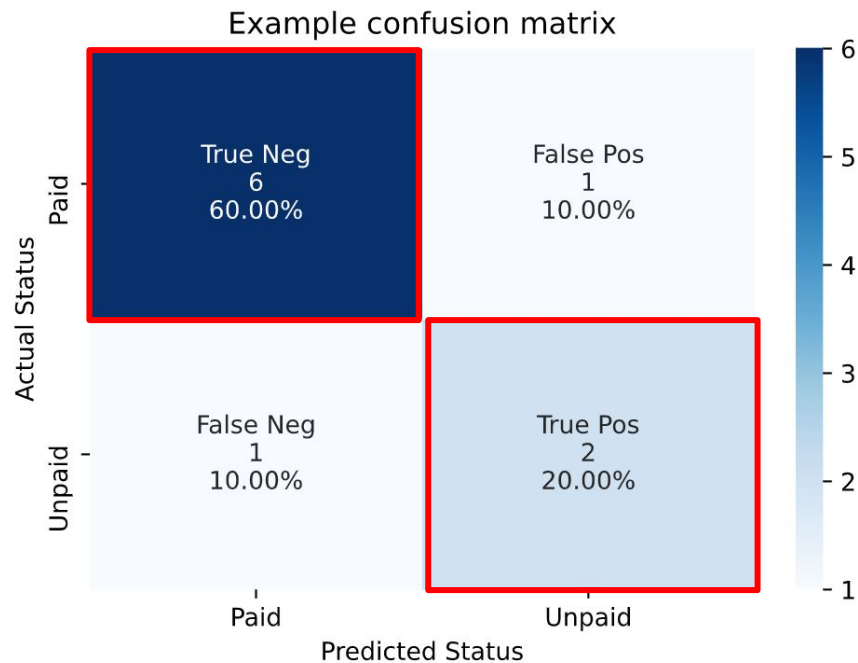
- A) Since tickets are only written if infractions are found, the city must decide how to allocate employees. To generate more revenue, identify areas and tickets more likely to yield payment.
- B) Since tickets are intended to be consequences, the city should be looking to monitor its citizens fairly, so no change should occur to ticket writing. However, it would be useful to know if the city could flag a ticket as being more likely to be delinquent and need following up.

# Metrics



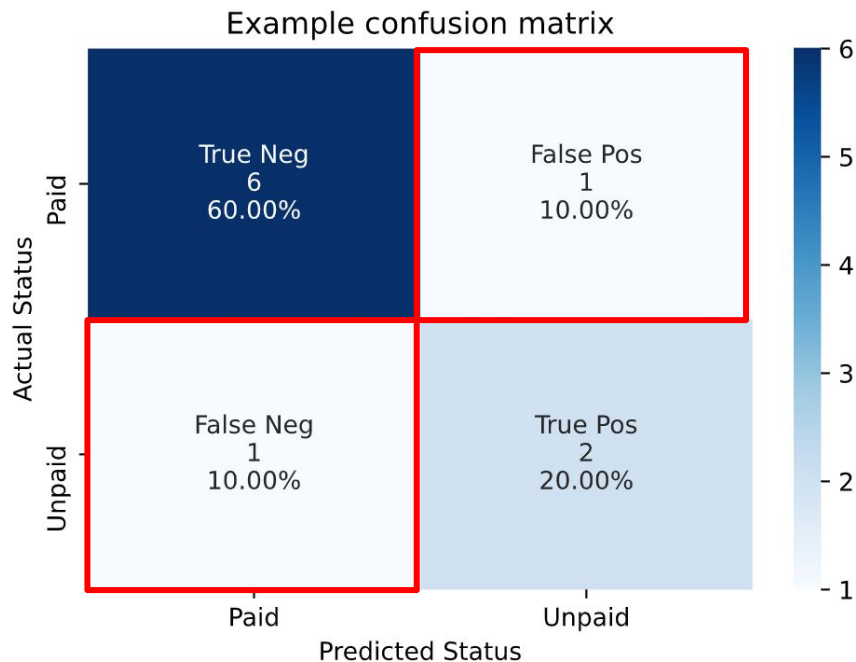


# Metrics



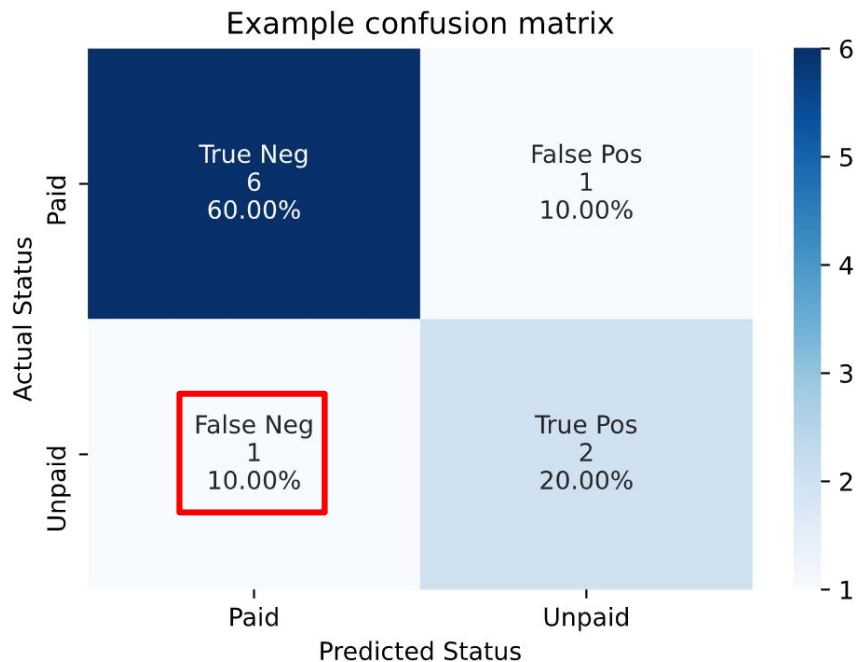
- Correct predictions on the main diagonal

# Metrics



- Correct predictions on the main diagonal
- False Positives and Negatives shown on the other diagonal

# Metrics



- Correct predictions on the main diagonal
- False Positives and Negatives shown on the other diagonal
- Displays number classified and percentage of the total

# ROC Comparison (On Validation Data)

