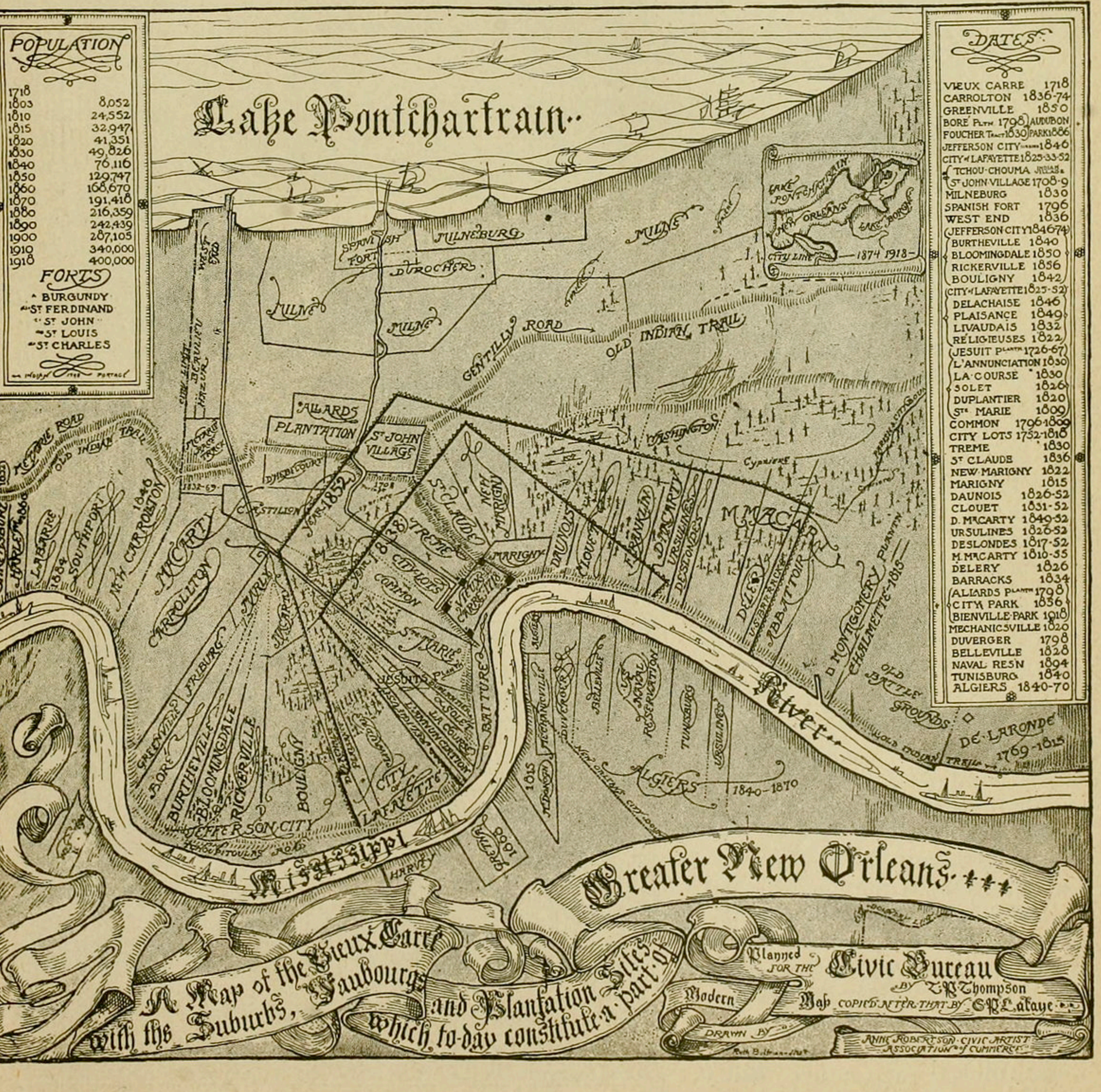


PROGRESS OF THE GROWTH OF NEW ORLEANS.

Optimizing Emergency Response Times in New Orleans

Exploring strategies to improve emergency response times and optimize public safety in the city of New Orleans.

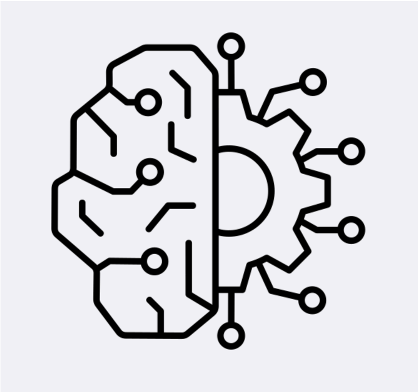
Project Overview

Data Collection and Cleaning

Collect and clean response time data for emergency calls reported to the Orleans Parish Communication District (OPCD) in 2023.

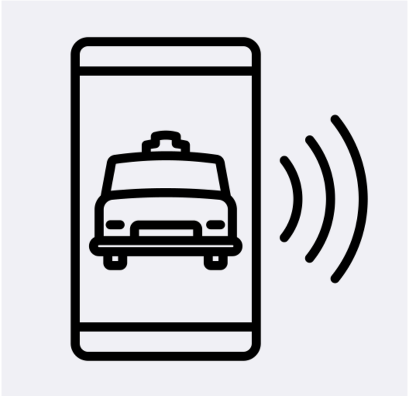
Data Exploration

Investigate and assess the distributions of response times, incident category, and check for balanced class distributions.



Predictive Modeling for Response Times

Develop machine learning models to forecast emergency response times at risk for longer-than-acceptable wait times based on historical data and relevant factors.



Future App Development

The app would integrate with the NOPD's existing systems via APIs, providing real-time updates and response time predictions to support dynamic resource management.

By analyzing historical data, developing predictive models, and identifying key factors, this project aims to provide actionable insights to the OPCD for optimizing emergency response times in New Orleans.

Hypothesis

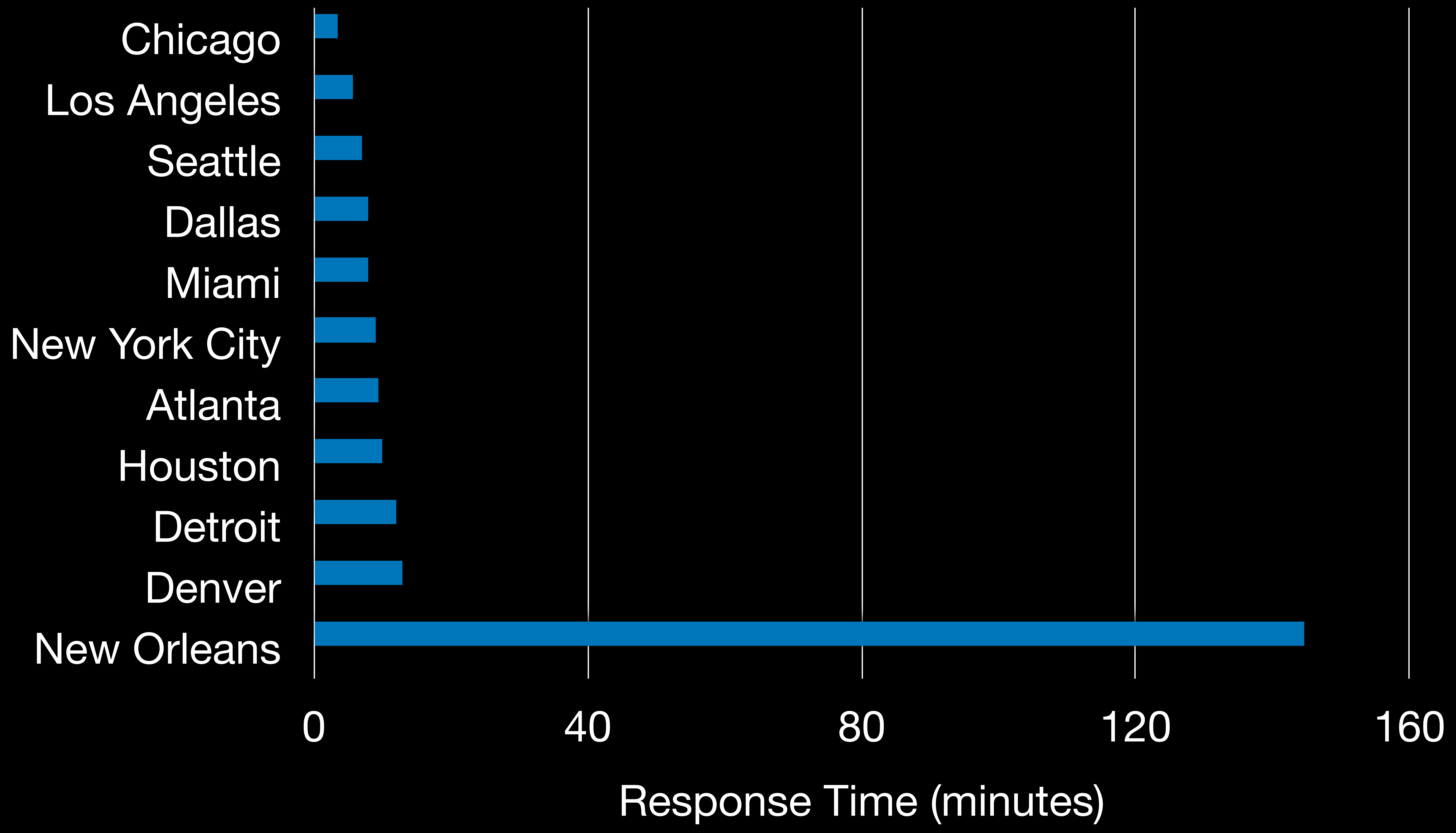
By creating a model that forecasts unusually high 9-1-1 response times, the NOPD and OPCD can coordinate resource allocation when these cases arise in order to improve decision making and enhance public safety.

Context

New Orleans continues to face an emergency responder (NOPD and OPCD) staffing crisis and poor 9-1-1 response times compared to national averages.

"Due to the slower response, 34 percent of the time, the caller is gone before the police arrive."

\- wdsu.com



Criteria for Success

- **Decrease** in average emergency service response time.

- **Lowered crime rate** as a secondhand effect of improved NOPD readiness.

- **Positive feedback** and input from key stakeholders.

- Improved NOPD and Orleans Parish Communication District (OPCD) staffing and **resource allocation.**

https://veritenews.org/2024/06/04/new-orleans-911-call-answer-times/

Scope of Solution Space

Resource Allocation	*Response Time Improvement*	*Technology Integration*
• Optimize resource allocation within NOPD by district and time of day. • Budget for live dashboard development and affiliated technology training with NOPD staff is efficient and effective.	• Identify factors contributing to prolonged response times. • Utilize measures like 'InitialCalltoDispatchTime', 'DispatchToArriveTime', and 'ArrivaltoClose' to assess response time efficiency. • Propose strategies such as optimized dispatch protocols and improved staffing distribution based on analysis of 'Type', 'Priority', and 'SelfInitiated' columns.	• Explore the integration of predictive analytics and real-time dashboards to enhance law enforcement operations. • Implement technology solutions to improve emergency service response efficiency.

https://www.neworleans.com/plan/faqs/visitor-safety/

Stakeholders

- NOPD
- Citizens of New Orleans
- New Orleans City Council
- OPCD
- Emergency Services Personnel
- Crime Data Analysts and Researchers

Constraints

- Financial constraints and resource limitations may hinder the scale and effectiveness of intervention programs.

- Social barriers can affect community engagement and support for proposed solutions.

- Elimination of sex offense-related calls (because of confidential coordinate information) creates a hole in the data for that incident category.

- Data entry and mechanical error by OPCD.

- Salary and recruiting limitations of NOPD and OPCD.

The Dataset

- **Downloaded from**:
  - https://catalog.data.gov/dataset/calls-for-service-2023
- **File specs:**
  - “Calls_for_Service_2023.csv"
  - 81.4 MB CSV
  - File Shape: 325,091 rows, 21 columns
- **Data Entry:** Completed by Orleans Parish Communication District
  - Administrative office for 9-1-1
- **Dataset Limitations**
  - Entries can be changed at a later date upon further investigation
  - Possibility of mechanical or human error

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 325091 entries, 0 to 325090
Data columns (total 23 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   NOPD_Item         325091 non-null  object
 1   Type              325091 non-null  object
 2   TypeText          325091 non-null  object
 3   Priority          325091 non-null  object
 4   InitialType       325091 non-null  object
 5   InitialTypeText   325091 non-null  object
 6   InitialPriority   325091 non-null  object
 7   MapX              325091 non-null  int64
 8   MapY              325091 non-null  int64
 9   TimeCreate        325091 non-null  object
 10  TimeDispatch      157279 non-null  object
 11  TimeArrive        262397 non-null  object
 12  TimeClosed        325091 non-null  object
 13  Disposition       325082 non-null  object
 14  DispositionText   325082 non-null  object
 15  SelfInitiated     325091 non-null  object
 16  Beat              323521 non-null  object
 17  BLOCK_ADDRESS     325090 non-null  object
 18  Zip               320441 non-null  float64
 19  PoliceDistrict    325091 non-null  int64
 20  Location          325091 non-null  object
 21  Latitude          325091 non-null  object
 22  Longitude         325091 non-null  object
dtypes: float64(1), int64(3), object(19)
memory usage: 57.0+ MB
```

Data Transformation

Column Derivation

- **Geological:** Latitude / Longitude columns for future ArcGIS visualization in Tableau

- **Categorical:** "Incident Category" feature created in order to observe various types of emergency incident trends

- **Temporal:** New calculated measures of response time

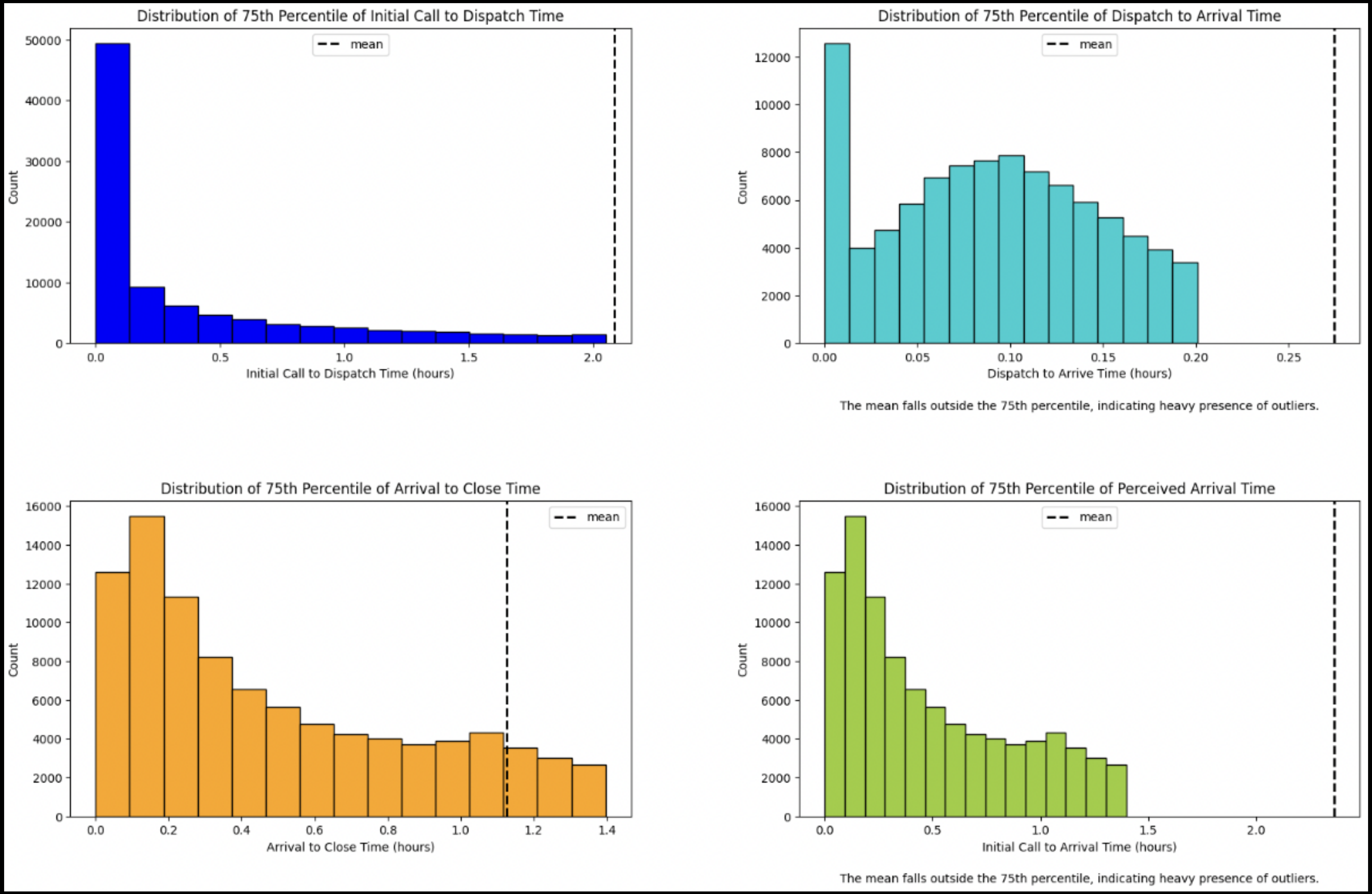
2e.1 Dispatch Response Time = TimeDispatch - TimeCreate

2e.2 Scene Arrival Response Time = TimeArrive - TimeDispatch*

2e.3 Time Arrive to Time Closed = TimeClosed - TimeArrive

2e.4 Time Create to Arrival Time = TimeArrive - TimeCreate

Exploratory Analysis



Feature Engineering for Classification Models

Median
New Orleans
9-1-1 Call Wait Time:
21.30 minutes*

* Defined as the threshold

- **Target Variable:** binary measure of "high risk" or "low risk"
  - "Risk" defined as a case where long wait times could result in a negative outcome
  - "high risk" for above threshold expected wait time
  - "low risk" for below threshold expected wait time

Final Model

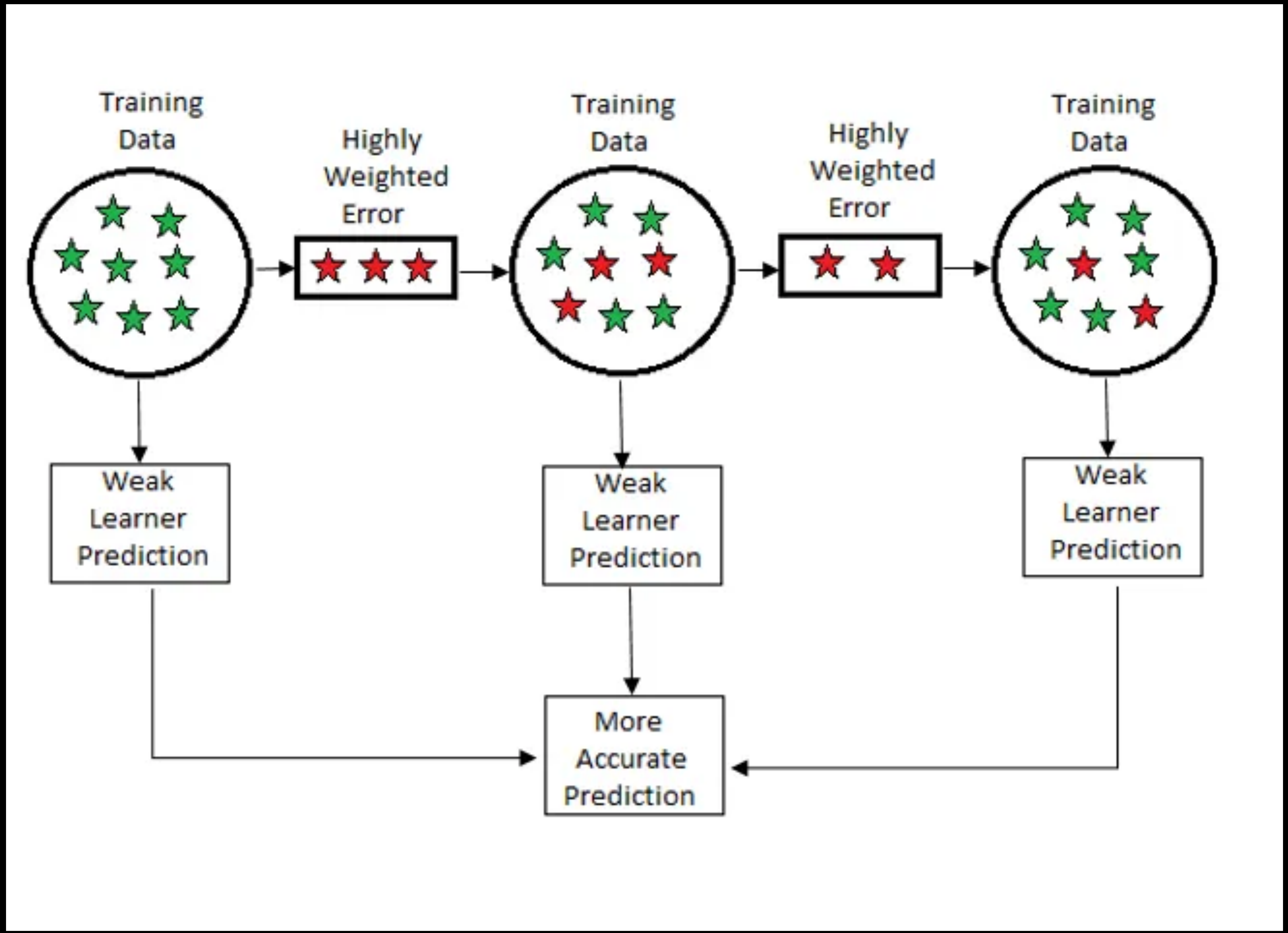
Why Gradient Boosting Classifier?

Handling Complex Relationships:

911 call data often involves factors that might not have a simple linear impact on dispatch time category (e.g., type of incident, location, time of day).

Robust to Noise and Outliers:

This dataset, even after cleaning, has potential for data entry errors. It has also been confirmed to contain outliers as seen in the data cleaning notebook.



https://pub.towardsai.net/fully-explained-gradient-boosting-technique-in-supervised-learning-d3e293ca70e1

Assessing Model Performance

Gradient Boosting Classification Report:

	precision	recall	f1-score	support
high_risk	0.92	0.88	0.90	12502
low_risk	0.89	0.93	0.90	12570
accuracy			0.90	25072
macro avg	0.90	0.90	0.90	25072
weighted avg	0.90	0.90	0.90	25072

Precision for High Risk: 0.92

92% of the calls flagged as high risk truly are high risk. This is critical in a 911 context because false positives (incorrectly labeling a low-risk call as high risk) could lead to unnecessary resource allocation.

Recall for High Risk: 0.88

88% of the actual high-risk calls are correctly identified. A higher recall means that the model is effective at capturing most of the high-risk cases, which is crucial for ensuring timely response to critical incidents.

Precision and Recall for Low Risk:

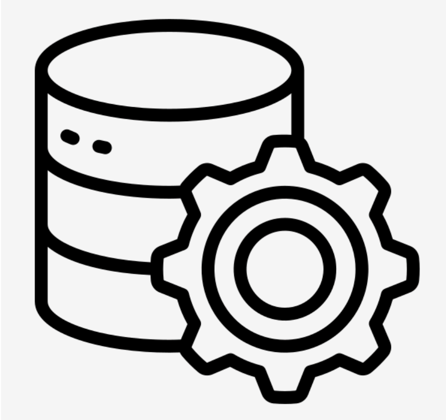
The precision and recall are balanced here as well (0.89 precision, 0.93 recall). This balance ensures that low-risk calls are correctly identified, reducing the chances of underestimating the risk of a call that might actually be urgent.

Future App Development

Front End and Back End Development

Design a user-friendly interface for visualizing trends, hotspots, and response times. Implement interactive dashboards using JavaScript libraries like D3.js or Plotly.

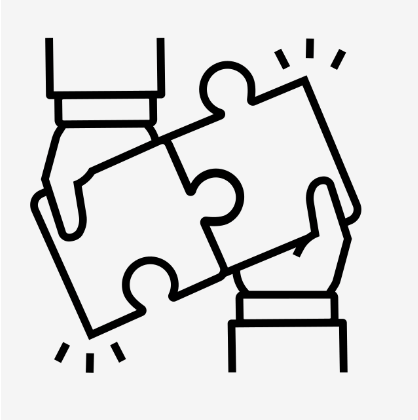
Develop RESTful APIs using Flask or Django to handle data requests. Integrate with the predictive model to serve response time predictions.



Database Management

Set up and configure the database (e.g., AWS RDS).

Design the database schema to store incident data, model predictions, and user information. Implement ETL processes to ingest and preprocess data from various sources.



Integration & Testing

Integrate front-end and back-end components. Perform unit testing, integration testing, and end-to-end testing to ensure the app works as expected.

Conduct load testing to ensure the app can handle high traffic and large volumes of data.

Deployment

Deploy the app to a cloud platform like AWS or Azure. Use containerization tools like Docker for easy deployment and scalability.

Set up CI/CD pipelines for automated testing and deployment.


Thank you.