Ritwik Awasthi

NYC Incident Volume Prediction

03.13.2025

“Better safe than sorry”

People responding to emergencies cannot afford to lose any precious time. **Seconds** can often mean the difference between **life and death**.

In a city like New York, where getting around quickly is a challenge, are there a better way to prepare for emergencies?

The Problem

Given that there exists a significant relationship between time and number of incidents, we can leverage machine learning to build a statistical model to make smarter guesses about how many incidents we might expect to see across the city.

Predicting how the number of incidents vary throughout the day in differents parts of the city can provide essential insight to assist emergency personnel prepare and manage resources accordingly.

A Calculated Guess

Impact

Dynamic police routes to ensure public wellbeing.

Pre-emptive relocation of EMS personnel and resources.

Predictive scaling of emergency systems, such as AI agents.

Insight into future city planning and infrastructure development.

Last but not least, lives saved.

Acquire	NYC Open Data (2018 - Present) Dataset
Aggregate	How can we derive hourly, information from this data? Grouping, cleaning, and data encoding.
Analyze	Model selection, evaluation, and tuning.

The Data

Acquire & Aggregate

Original Data

Each record is a call made to a 911 operator in NYC along with a timestamp, location, and description of incident

Combining

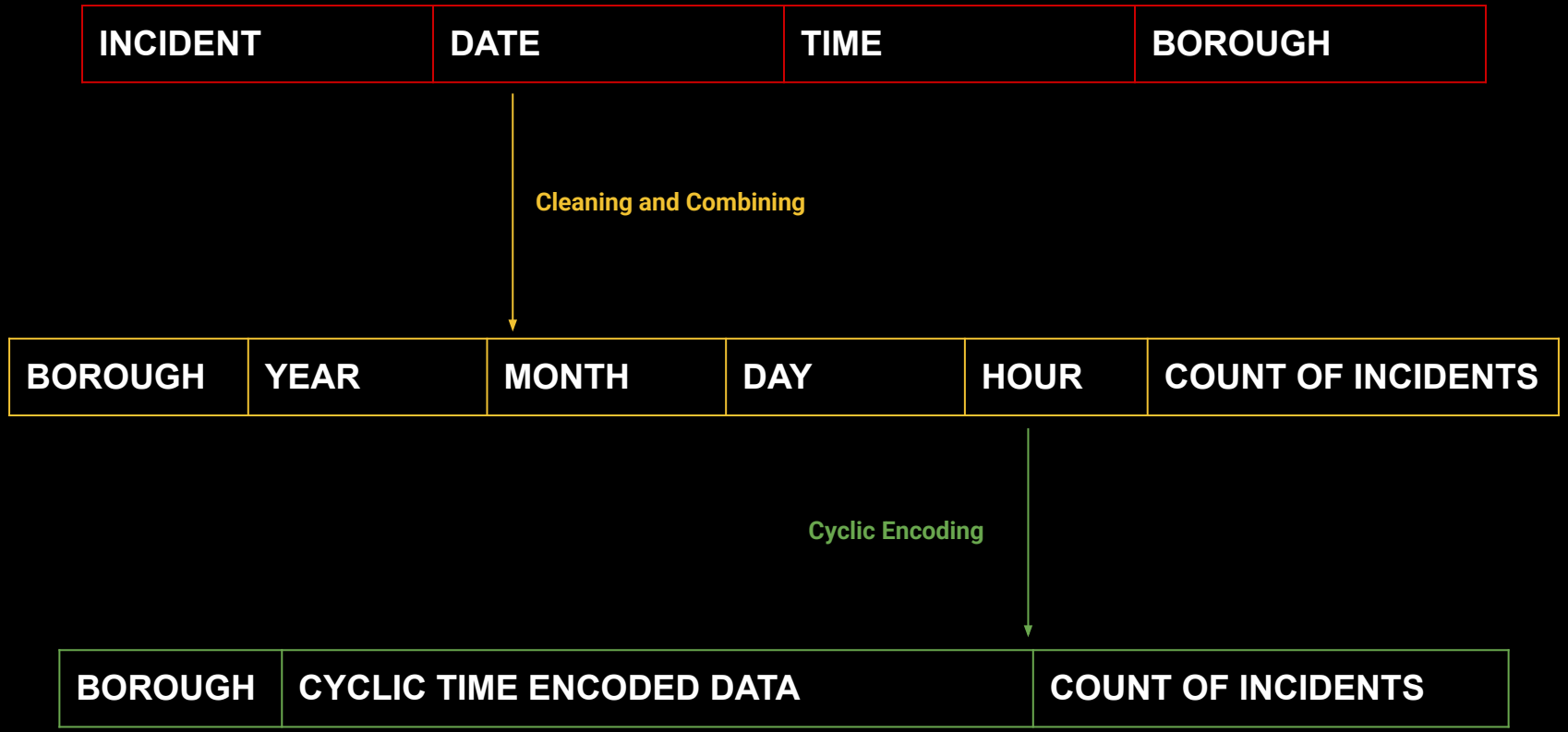
Using the timestamp and location of the incident, we can group incidents by time and borough, resulting in hourly counts of incidents per borough

Cyclic Encoding

Using trigonometric functions to map linear periodic information to a cyclical representation.

Scaling and Cleaning

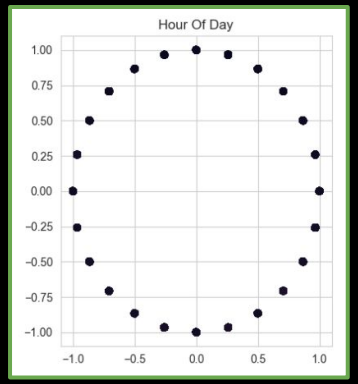
Scaling the data to ensure performance on distance based algorithms. Cleaning is performed to ensure model integrity.


INCIDENT
DATE
TIME
BOROUGH
Cleaning and Combining
BOROUGH
YEAR
MONTH
DAY
HOUR
COUNT OF INCIDENTS
Cyclic Encoding
BOROUGH
CYCLIC TIME ENCODED DATA
COUNT OF INCIDENTS

Cyclic Encoding

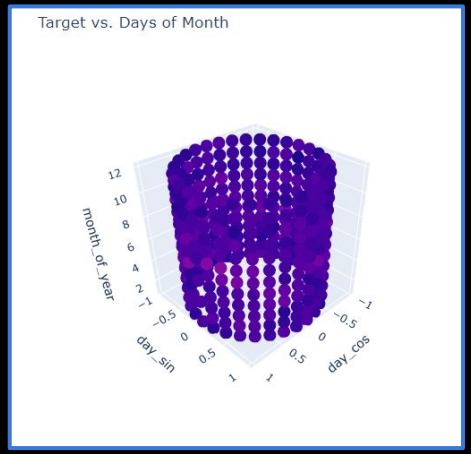
The idea behind cyclic time encoding is that instead of treating time as a linear variable, we transform time into a representation that "wraps around" after a certain period. This method is useful when implementing models that are distance-based such as K-means.

YEAR	MONTH	DAY	**HOUR**	COUNT OF INCIDENTS

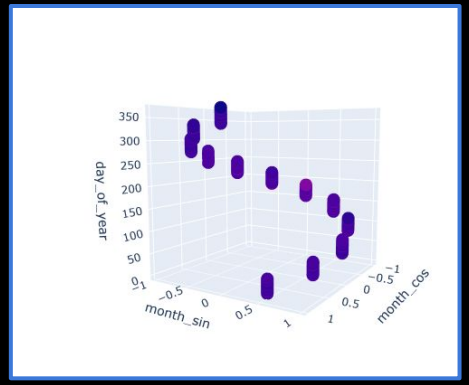


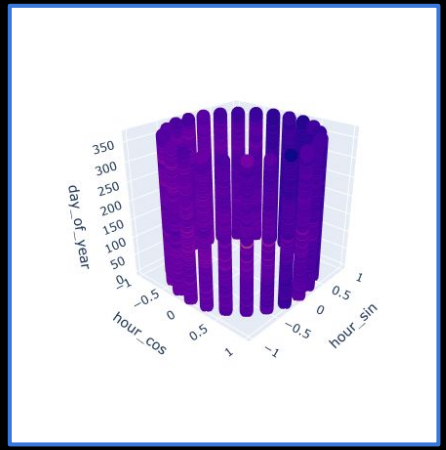
Cyclic Encoding

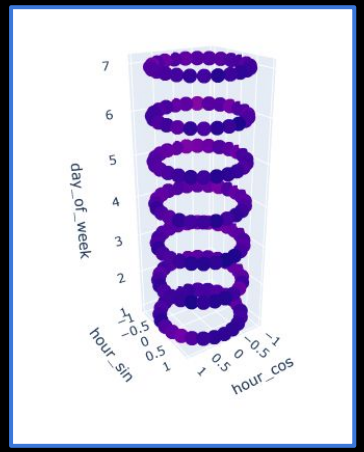
Model Input

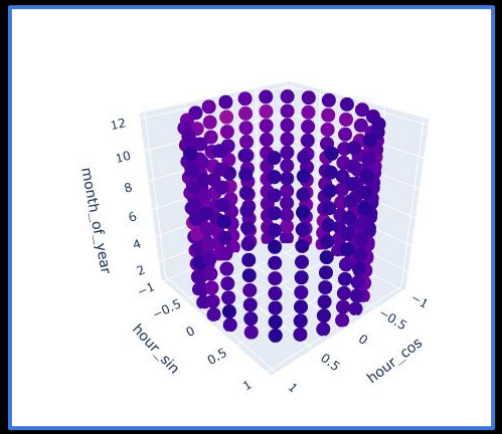


The intricate patterns that we are expecting our model to learn.









Model Selection:
Accuracy Of Predictions

44%

Naive Model

If our model just estimates the average across all records of data.

Computationally cheap
Not accurate at predicting fluctuations.

75%

Linear Regression

Simple linear regression model utilizing cyclic encoded data.

Good performance
Computationally cheap
Simple and explainable

81%

K-Nearest Neighbor Regression

Model expected to benefit most from cyclic encoding. Distance based ML algorithm.

Great performance
Computationally expensive
Overfitting

86%

Gradient Boosting

Cutting edge ML algorithm based on the idea that many weak learners make a strong predictor.

Best performance
Computationally expensive
Minimal overfitting

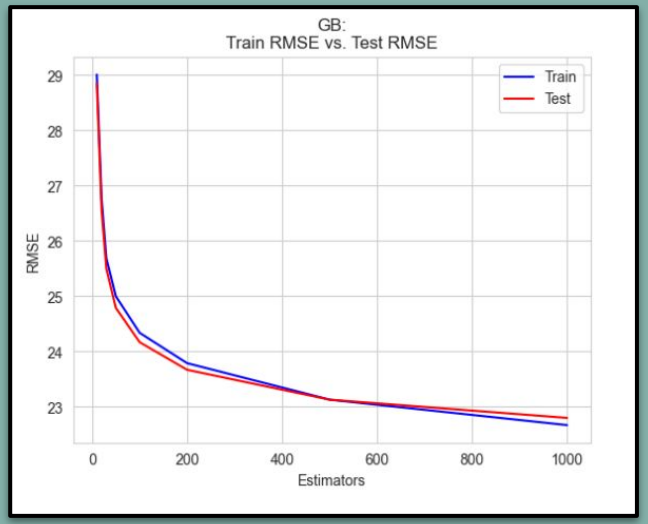
Model Tuning:
Optimizing Gradient Boosting

Gradient Boosting is very sensitive to hyperparameters, therefore, A Grid Search Cross-Validation was implemented in order to efficiently search for the optimal hyper parameters.

Given that we are working with gradient boosted trees, we must ensure that the weak learners that we develop are actually weak learners, therefore, the max depth of trees is limited to 3.

Not only do we have to maximize the predictive performance, we must also consider the performance of the model in an implementation setting: A more complex and bigger model will take longer to make predictions. A 20% increase in computational cost might not be worth predicting few more incidents correctly.

Lastly, we need our model to generalize well and not “learn” the random noise in the data.



Model Implementation:
What would a real world use case look like?

The application featured here showcases the model running on a dashboard predicting the number of incidents in NYC per borough. It has the ability to go live or to search by month, day, and hour. A sample alert is issued for Brooklyn in the example below as the number of incidents has surpassed the threshold of 300.

At a larger scale, and with more granularity, the predictive capabilities of this model has the power to reinvent how EMS and police services are offered throughout the cities.

[Link to Application]