Predicting NYPD Response Times after Dispatch

Max Grove | mg6392@nyu.edu

Big Data | Dr. Juan Rodriguez

Tandon School of Engineering

New York University

Outline

- Introduction
- Data
- Pipeline of application
- Loading Data
- Sampling Data
- Cleaning Data
- Exploring Data
- Modeling
- Conclusion

Introduction

Goal

Predict how long it takes a dispatched NYPD officer to arrive at the scene

DataSet

- **NYC's Open Data Initiative**
- **Documented information on NYPD 911 calls and responses**
- **Data is operated by members of the public *and* NYPD members**

Two datasets:
- Historic
  - 2.3 GB size
  - 20 columns
  - 40.7M Rows
- YTD
  - ~900 MB size
  - 18 columns
  - 3.6M rows

Operated by members of the public *and* NYPD professionals

Data Definition

Identifiers	
objectid	Number
cad_evnt_id	Number

Miscellaneous	
nypd_pct_cd	Number
radio_code	Text
typ_desc	Text
cip_jobs	Text

Timestamps	
create_date	Floating Timestamp
incident_date	Floating Timestamp
incident_time	Text
add_ts	Floating Timestamp
disp_ts	Floating Timestamp
arrivd_ts	Floating Timestamp
closng_ts	Floating Timestamp

Location	
boro_nm	Text
patrl_boro_nm	Text
geo_cd_x	Number
geo_cd_y	Number
latitude	Number
longitude	Number
location	Point

Pipelining Data and Technology

Collecting and understanding Data

Hosting data and indexing

Cleaning data
Exploring data
Modeling data

Loading Data to MongoDB

```
api_endpoint_ytd = "https://data.cityofnewyork.us/resource/n2zq-pubd.json"

api_endpoint_historic = "https://data.cityofnewyork.us/resource/d6zx-ckhd.json"
```

- Create an App Token with NYC Open Data
- Fetch the data from the APIs - 20,000 data points per call
- Insert into a Mongo DB collection

Key MongoDB Features Used

- Add indexes on INCIDENT_DATE and cad_evnt_id
- Add in INCIDENT_YEAR and INCIDENT_DATE fields to sample by year-month
- Add indexes on INCIDENT_YEAR and INCIDENT_DATE

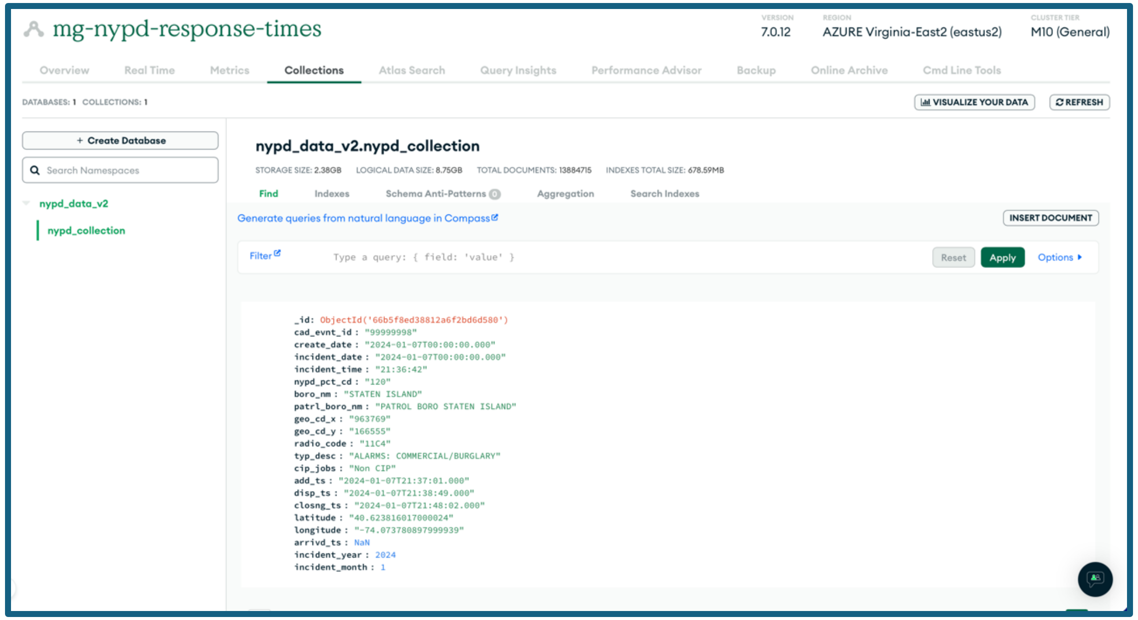
Sampling Data from MongoDB

Pull all available month-year combinations from MongoDB

Pull 1,000 rows from each month-year down into the local JupyterHub environment

Key Benefits

- Save on space
- Ensure we get data from all time periods
- Utilize indexed rows
- Don't have to rely on API and data is backed up



Cleaning and Preparing Data

Initial Schema

22 columns and 22,000 rows

_Id	Typ_desc
Cad_evnt_id	Cip_jobs
Create_date	Add_ts
Incident_date	Disp_ts
Incident_time	Closng_ts
Nypd_pct_cd	Latitude
Boro_nm	Longitude
Patrl_boro_nm	Location
Geo_cd_x	Incident_year
Geo_cd_y	Incident_month
Radio_code	Arrivd_ts

Cleaning

- Dropping unnecessary columns
- Dropping missing values
- Cleaning dates

New Features

- Tokenize words and create feature flags
- New time features
- Linking data to NYC Neighborhood data
- Key data: Time to arrive from dispatch time

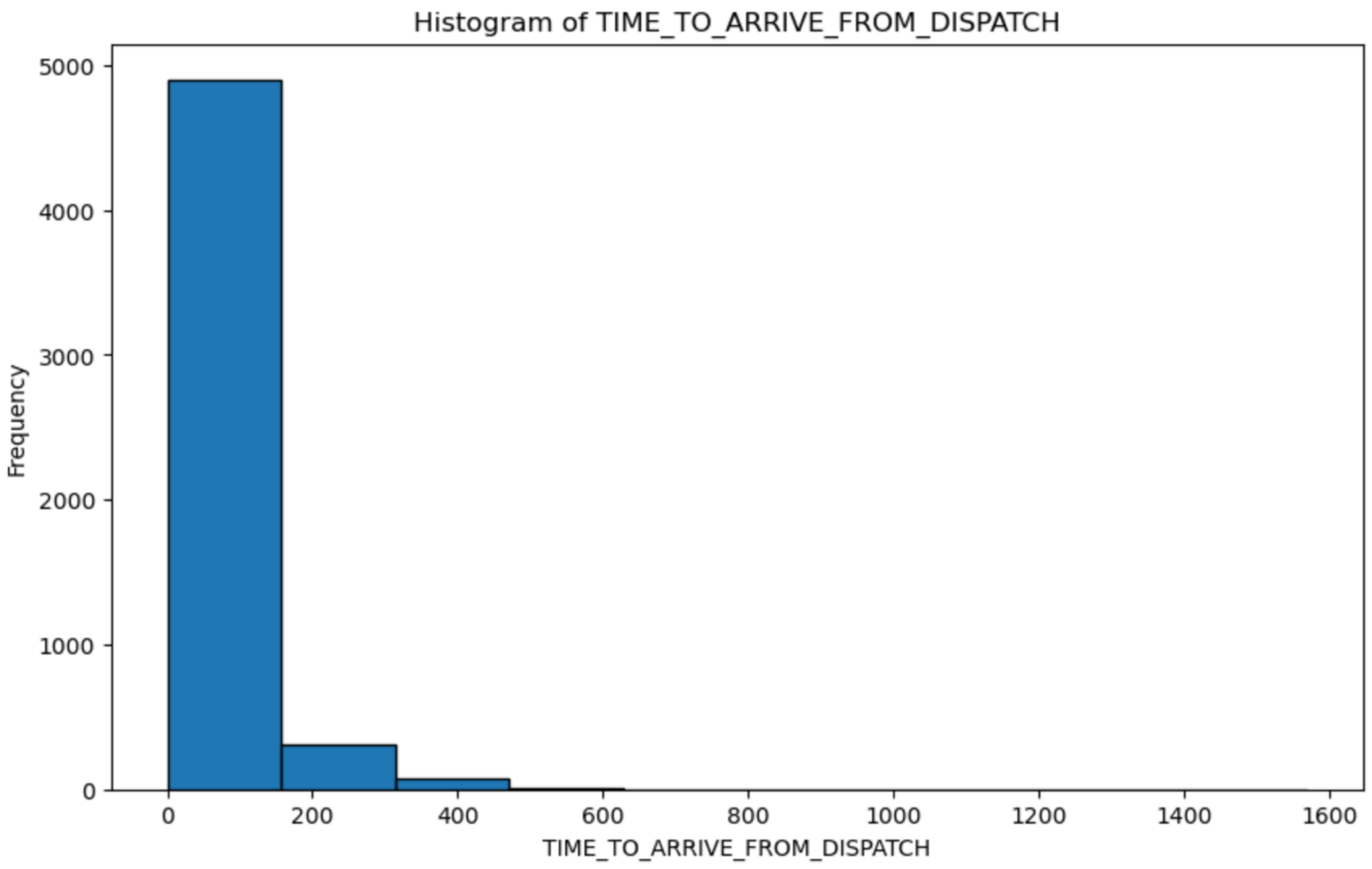
Cleaned Schema

22 columns and 5,291 rows

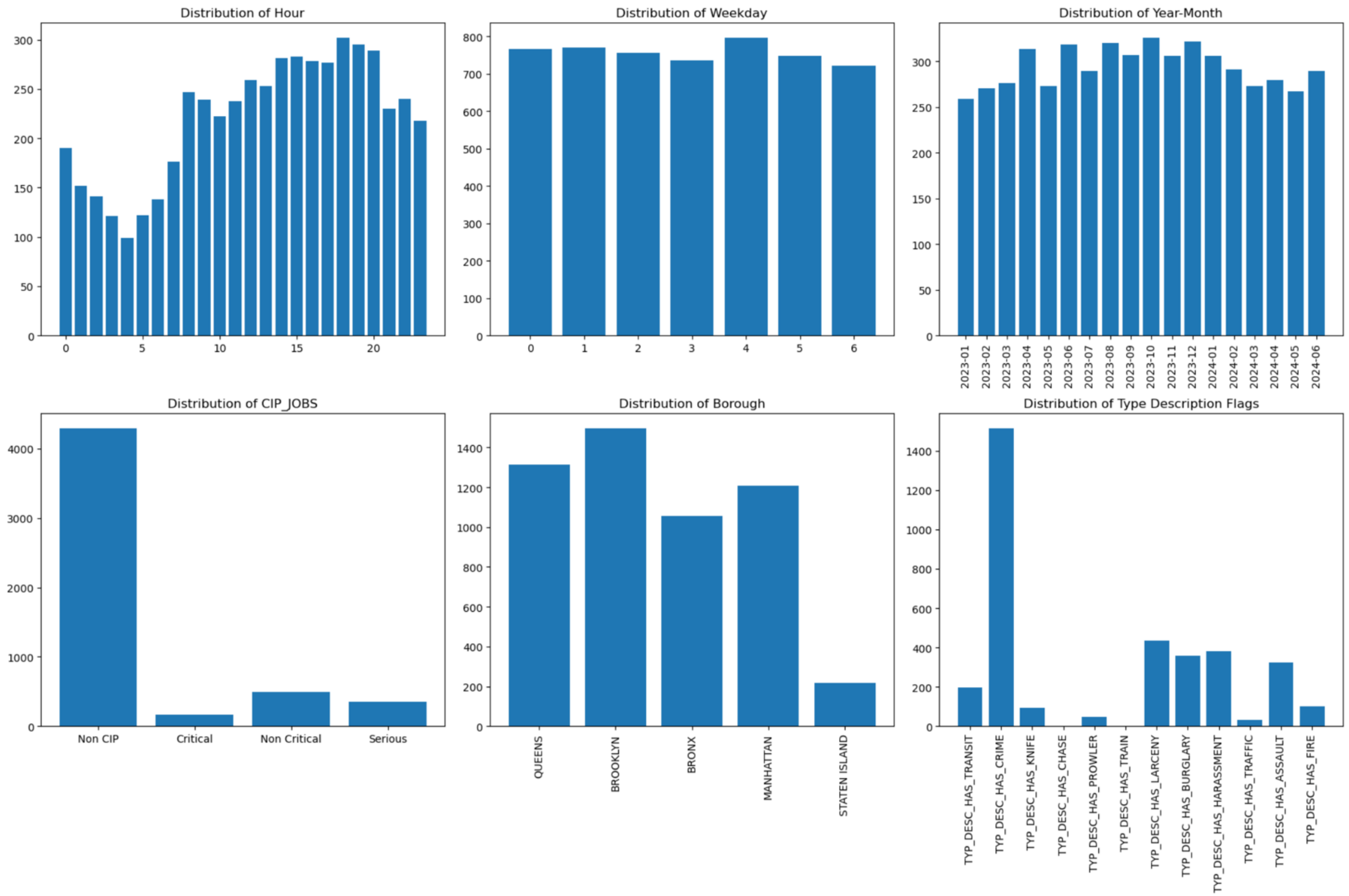
NYPD_PCT_CD
BORO_NM
RADIO_CODE
CIP_JOBS
INCIDENT_YEAR
INCIDENT_MONTH
TYP_DESC_HAS_{SELECTED_WORDS}
HOUR
WEEKDAY
TIME_TO_ARRIVE_FROM_DISPATCH
NEIGHBORHOOD

* New Features

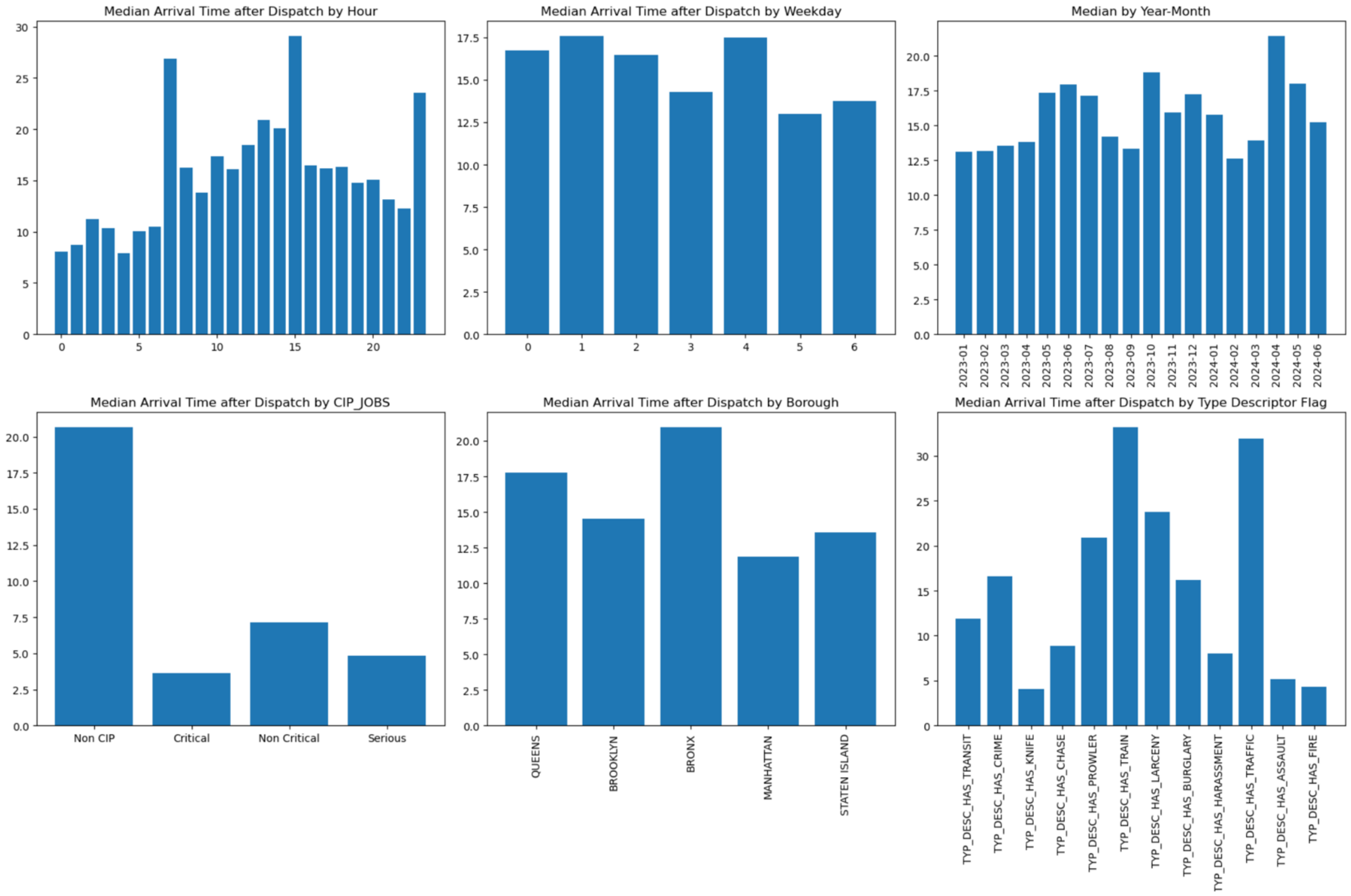
Evaluating and Understanding Data



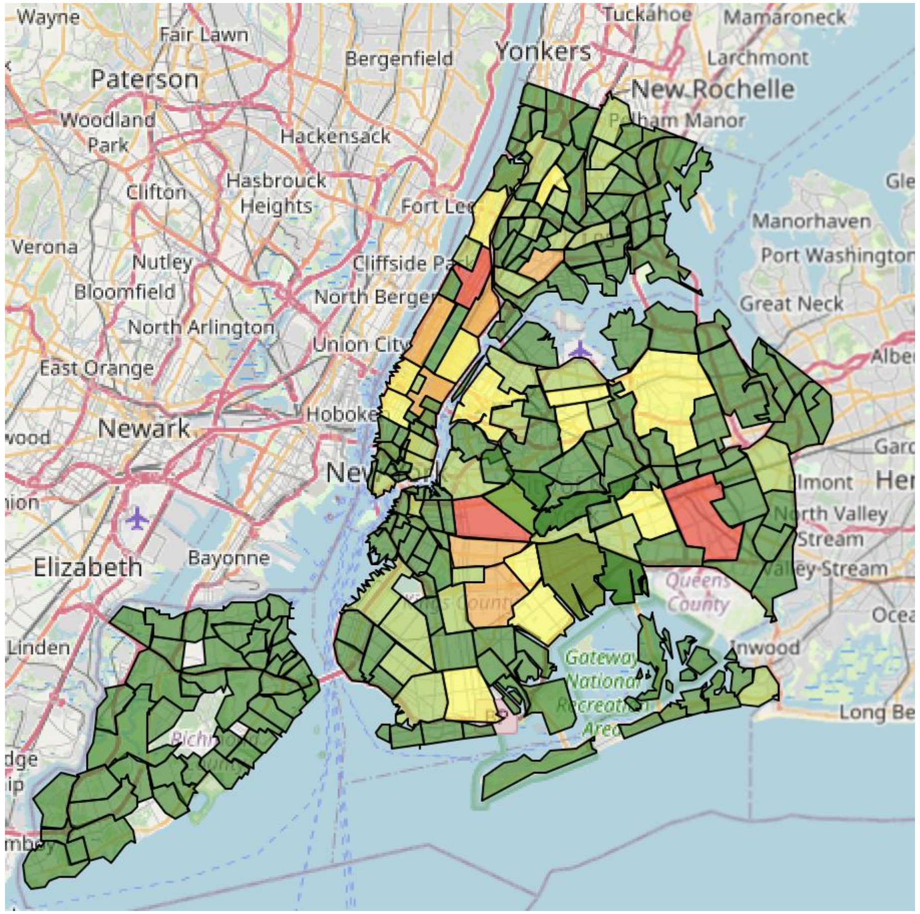
Evaluating and Understanding Data



Evaluating and Understanding Data

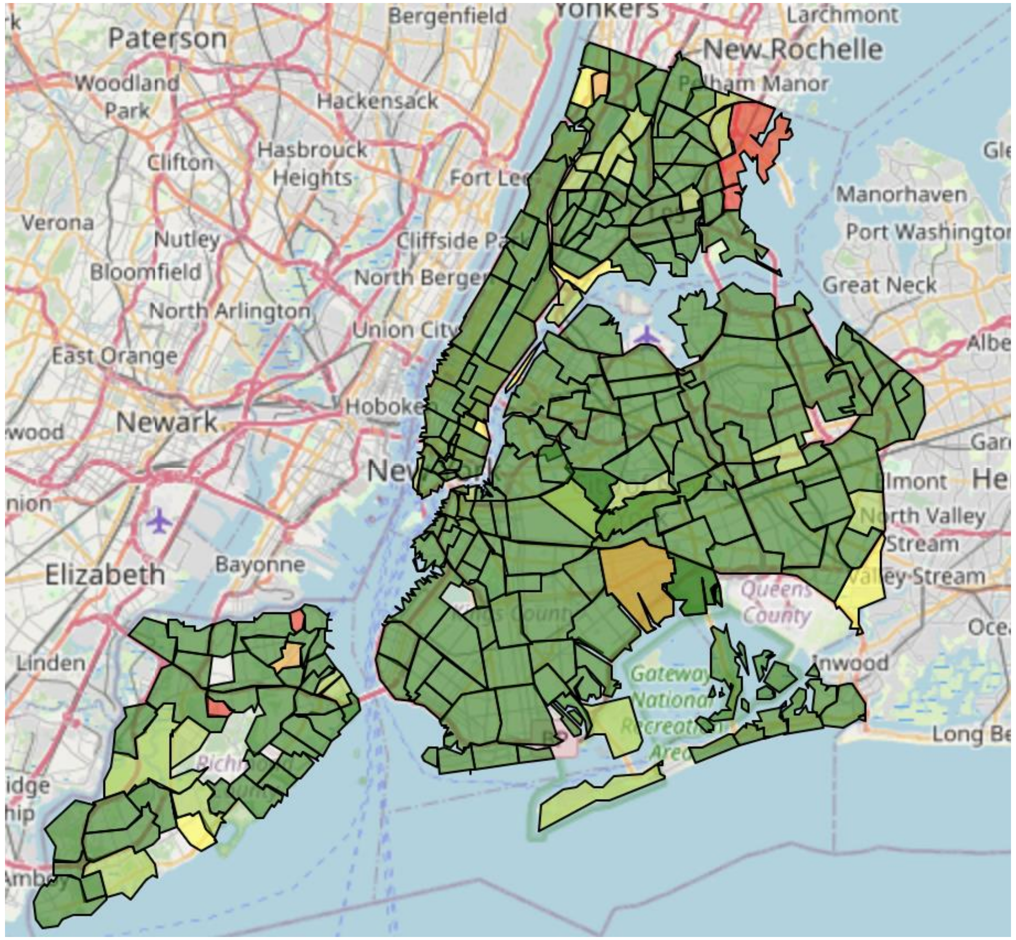


Evaluating and Understanding Data



- Red: 100th Percentile
- Orange: 80th Percentile
- Yellow: 60th Percentile
- Yellow-Green: 40th Percentile
- Green: Bottom 20th Percentile

Evaluating and Understanding Data



- Red: 100th Percentile
- Orange: 80th Percentile
- Yellow: 60th Percentile
- Yellow-Green: 40th Percentile
- Green: Bottom 20th Percentile

Machine Learning Models

Filter out actual arrival times greater than 40 minutes and less than 3 minutes

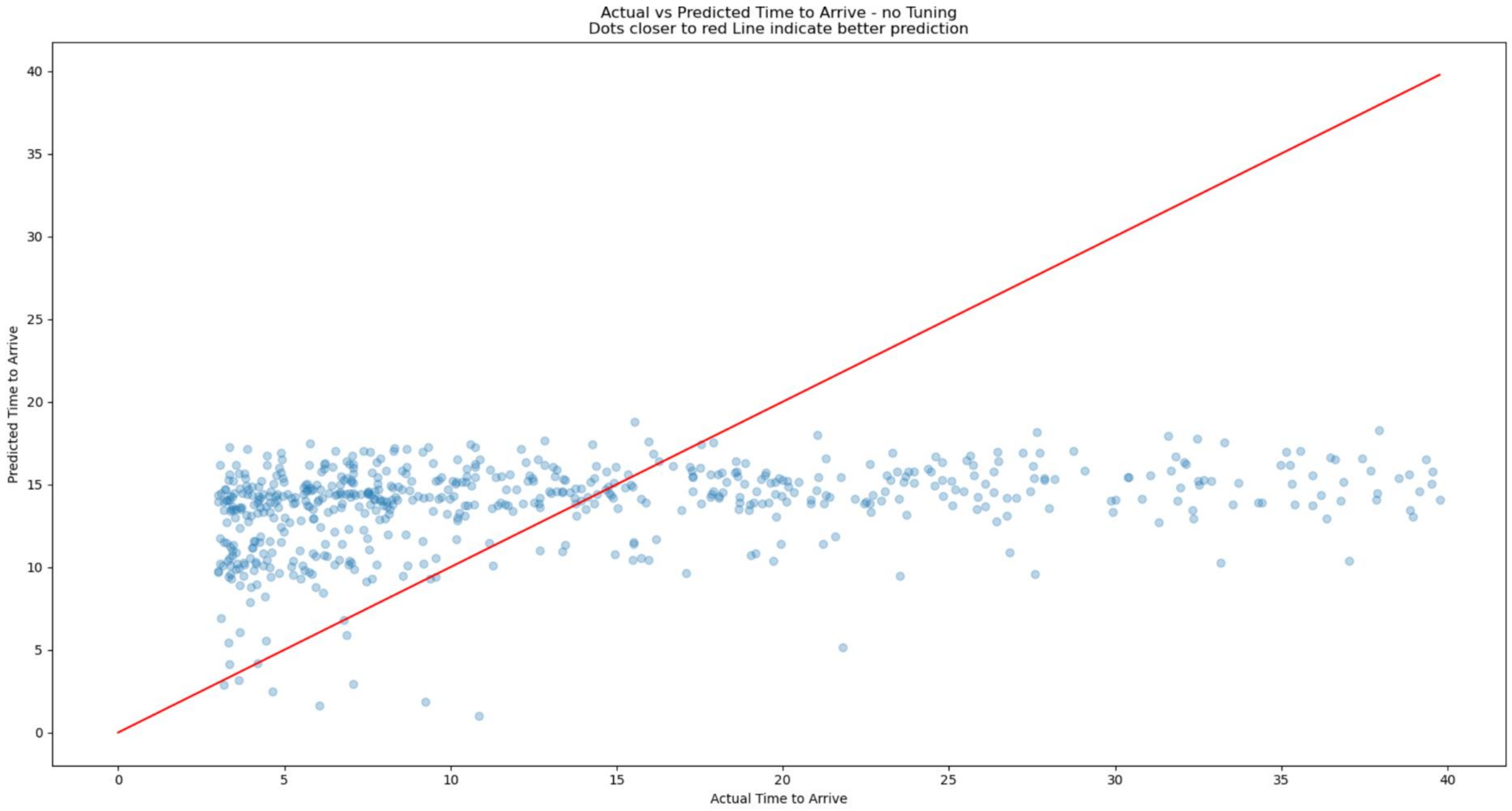
```
columns = [
   "BORO_NM",
   "RADIO_CODE",
   "CIP_JOBS",
   "NEIGHBORHOOD"
]
```

Create index values off of String columns

Create a test and training data set

Training data set: 80%
Test Data Set: 20%

Machine Learning Models



RMSE: 9.27

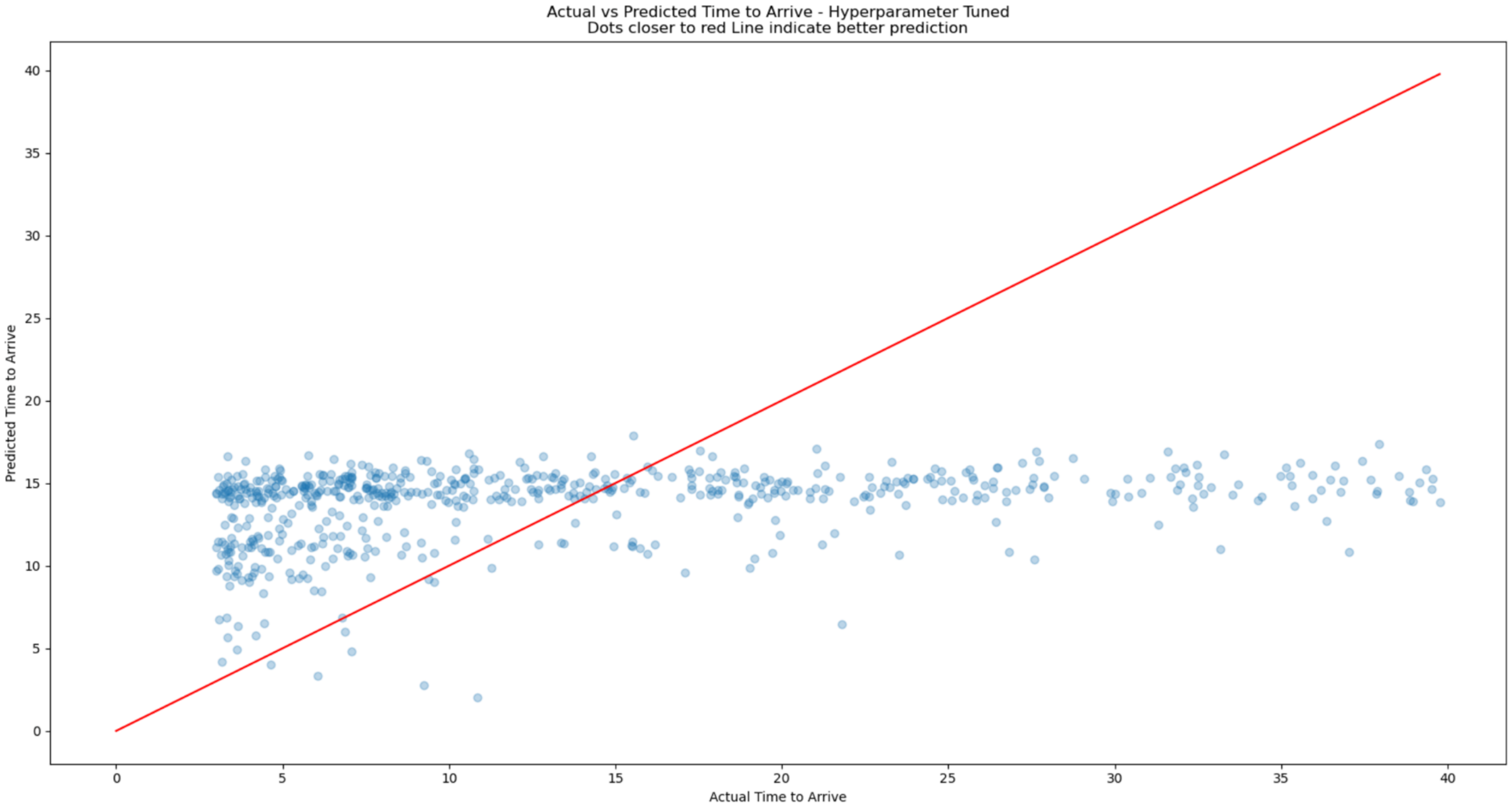
Machine Learning Models

Rerun model with tuning

- Parameter Grid
- Cross Validator with 5 folds

Hyperparameter Takeaways

- Hyperparameter tuning can lead to overfitting on the training data
- Chosen parameter grid might not be optimal
- Can sometimes make the model too restrictive or too flexible



RMSE: 9.31

Conclusion

Takeaways

- Can be incredibly difficult to clean and parse data from public sources
- Models often need lots of work to get them in shape
- Can be incredibly difficult to predict things that have many more inputs than what can be represented in data

Next Steps

- Further tune the model by removing ancillary features
- Run the model Borough by borough, or even neighborhood by neighborhood
- Run the model through Dask or other ML Models
- Bring in more historic data
- Bring in weather data to see if weather on a given day affects the arrival time