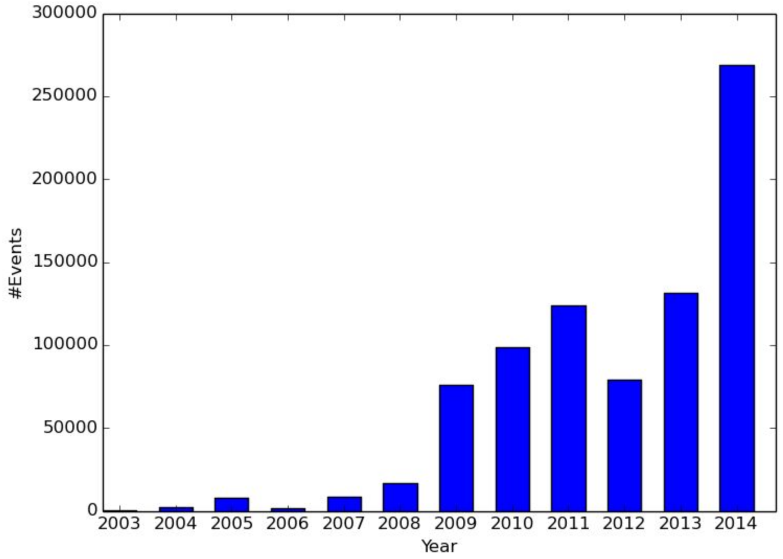
NIST 2015 Data Science Pre-pilot Evaluation Study

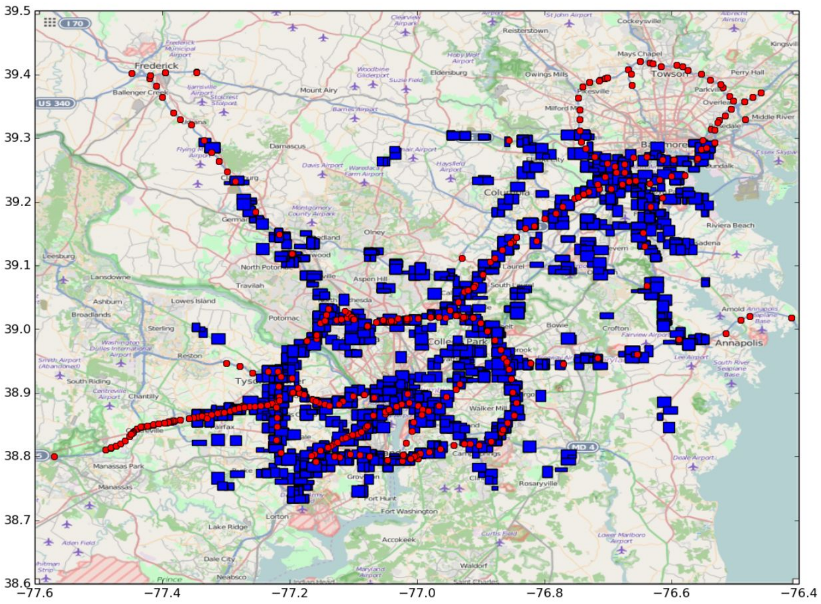
Dihong Gong, UF Data Science Research Lab
Advisor: Dr. Daisy Zhe Wang

Prediction Task

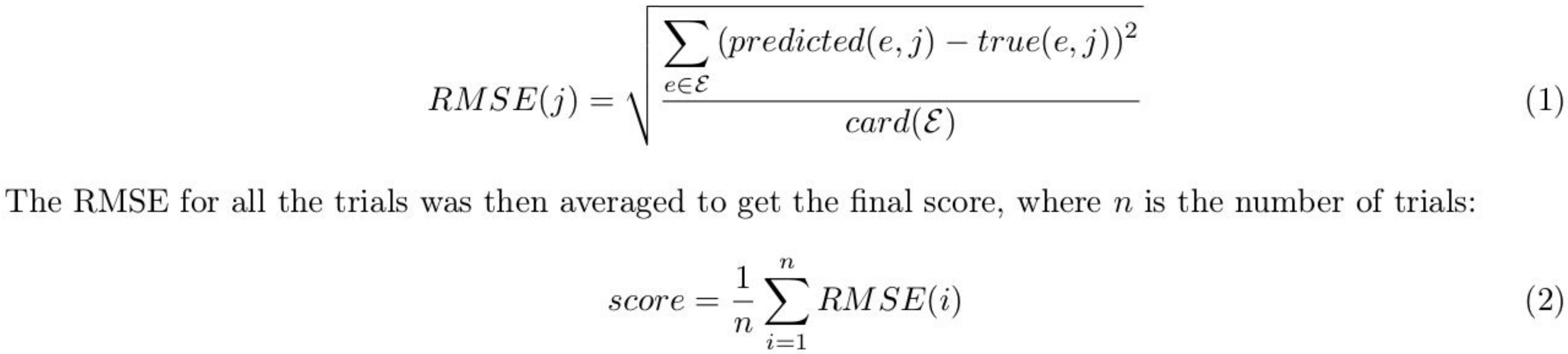
- **Description:** develop a system that can predict the number and types of traffic events by type for a given (geographical bounding, interval of time) pair.
- **Data Information:**
    - Traffic Events
    - OpenStreetMaps
    - Weather Data
- **Challenge:**
    - The missing measurements for some event types and years.
    - The measurements are inconsistent (e.g. due to different measurement techniques/standards used over years) -- makes it more difficult to learn regression models over years.
    - The data is relatively sparse -- lead to potential curse of dimensionality problem.
- **Participation:**
    - we have 7 teams (students coming from our data science class) participated in this challenge, with each team having one submission.

Event Data Information





How we Evaluated?



$$RMSE(j) = \sqrt{\frac{\sum_{e \in \mathcal{E}} (predicted(e, j) - true(e, j))^2}{card(\mathcal{E})}} \tag{1}$$

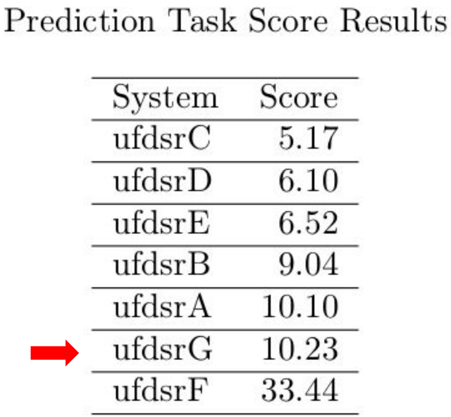
The RMSE for all the trials was then averaged to get the final score, where n is the number of trials:

$$score = \frac{1}{n} \sum_{i=1}^{n} RMSE(i) \tag{2}$$

Prediction Task - System ufdsrG

Authors: Babak Alipour, Roozbeh Ketabi, Alex Coleman

- Weather data by NOAA between 2003-2015, group by stations. (altitude, wind speed, visibility, temperature, dew point) taken from 3 stations closest to the bounding box, in addition to traffic event data.
- Road segment model, with OSM data.
- Traffic event data only (x,y,month).
- They reported that the (x,y,month) model works the best.
- Comments: the result of this group suggests that, the effective integration of diverse data source is challenging and we may end up with worse performance if they are not handled properly (due to noises or inconsistence).



Prediction Task Score Results

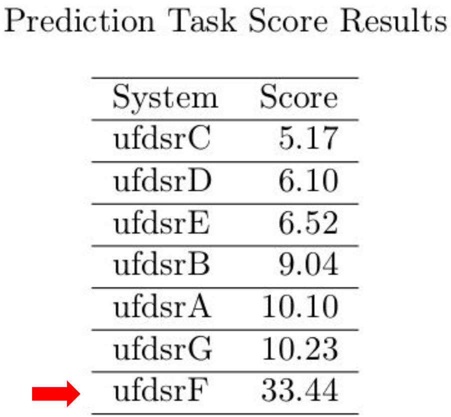
System	Score
ufdsrC	5.17
ufdsrD	6.10
ufdsrE	6.52
ufdsrB	9.04
ufdsrA	10.10
ufdsrG	10.23
ufdsrF	33.44

(smaller is better)

Prediction Task - System ufdsrF

Authors: Mebin Jacob, Pranav Achanta, Rui Zhang, Sandeep Dommaraju, Sudeep Reddy, Vipul Mittal

❏ Features: (year,month,x,y)
❏ Train one model per event type
❏ Accidents: linear regression
❏ Traffic condition, device status, obstruction: SVR
❏ Roadwork, precipitation: KNN



Prediction Task Score Results

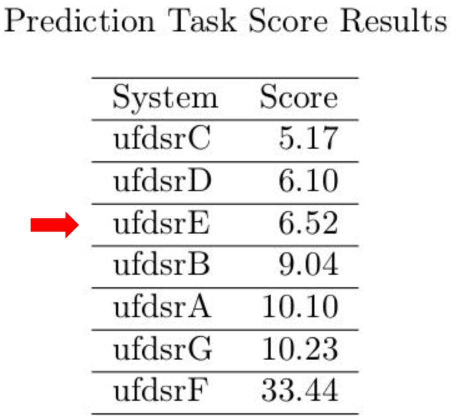
System	Score
ufdsrC	5.17
ufdsrD	6.10
ufdsrE	6.52
ufdsrB	9.04
ufdsrA	10.10
ufdsrG	10.23
ufdsrF	33.44

(smaller is better)

Prediction Task - System ufdsrE

Authors: Sergio Puleri, Max Fresonke, Kevin Wu, Craig Lu, Kevin Neumann

- ❑ Features: (month,x,y)
- ❑ Train one model per event type
- ❑ Linear regression



Prediction Task Score Results

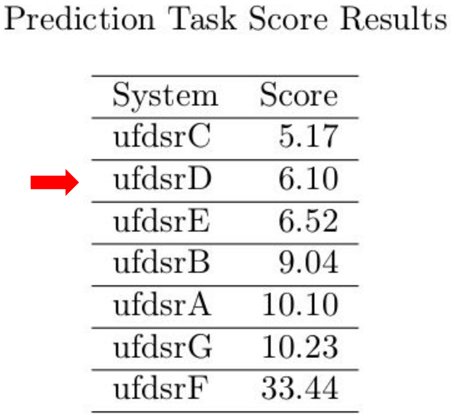
System	Score
ufdsrC	5.17
ufdsrD	6.10
ufdsrE	6.52
ufdsrB	9.04
ufdsrA	10.10
ufdsrG	10.23
ufdsrF	33.44

(smaller is better)

Prediction Task - System ufdsrD

Authors: Gokul Palwe, Rashmi Paliwal, Revathi Kadari

- ❑ Features: (month,x,y)
- ❑ Train one model per event type
- ❑ Polynomial regression (order 1,2,3,4)



Prediction Task Score Results

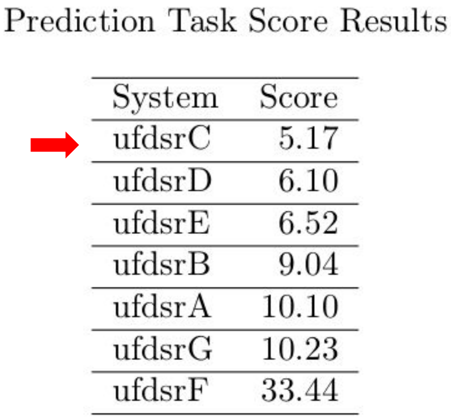
System	Score
ufdsrC	5.17
ufdsrD	6.10
ufdsrE	6.52
ufdsrB	9.04
ufdsrA	10.10
ufdsrG	10.23
ufdsrF	33.44

(smaller is better)

Prediction Task - System ufdsrC

Authors: Dihong and Miguel

- Features: (year,x,y)
- Train one model per event type
- Removed “zero-count” entries before training models.
- Linear regression



Prediction Task Score Results

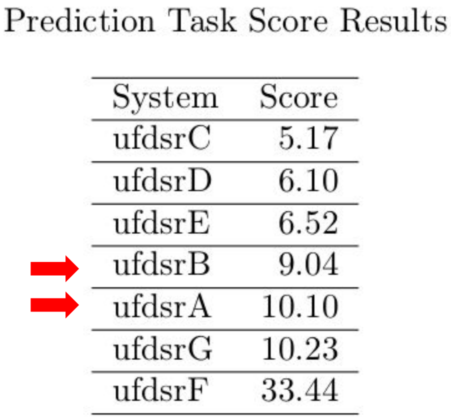
System	Score
ufdsrC	5.17
ufdsrD	6.10
ufdsrE	6.52
ufdsrB	9.04
ufdsrA	10.10
ufdsrG	10.23
ufdsrF	33.44

(smaller is better)

Prediction Task - System ufdsr[A|B]

❑ ufdsrA: merge of [C], [D], [E], [F].

❑ ufdsrB: merge of [C], [D], [E], [F] and [G].



Prediction Task Score Results

System	Score
ufdsrC	5.17
ufdsrD	6.10
ufdsrE	6.52
ufdsrB	9.04
ufdsrA	10.10
ufdsrG	10.23
ufdsrF	33.44

(smaller is better)

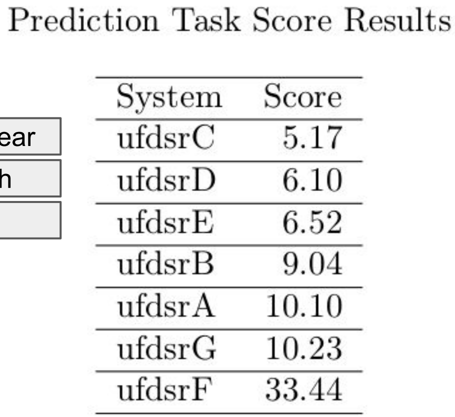
What we have learned

- Simple system works better, when we are unable to handle the excessive information properly.
- Cleaning the data before feeding into regression models is effective (reduced 1.4% errors compared to other systems).
- Higher order regression model can capture the exponential growth of #events better than the linear models.

Cleaning, Linear, Year

Polynomial, Month

Linear, Month



Prediction Task Score Results

System	Score
ufdsrC	5.17
ufdsrD	6.10
ufdsrE	6.52
ufdsrB	9.04
ufdsrA	10.10
ufdsrG	10.23
ufdsrF	33.44

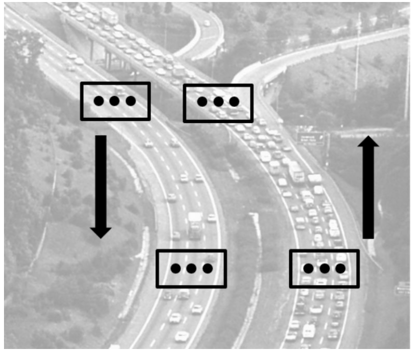
(smaller is better)

Cleaning Task

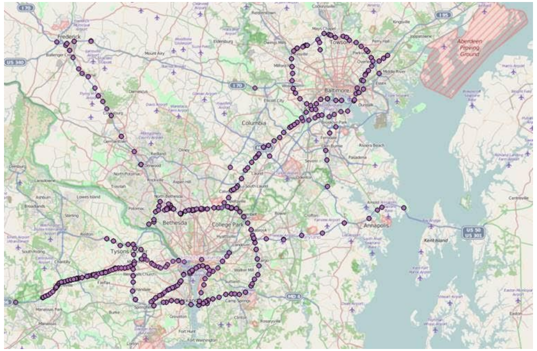
- **Description:** clean traffic lane detector measurements containing incorrect flow values, providing correct traffic flow values for the erroneous traffic flow measurements.
- **Data Information:**
  - Lane Detector Measurements (108 files each file containing 1.4 million measures on average)
  - Traffic Events (not used)
  - Traffic Camera Video (not used)
- **Challenges:**
  - Big data. The text data is around 150GM (compressed), with totally 1.46 billion records.
  - Detecting erroneous flow values. How do you efficiently detect the incorrect flow values based on dirty flow measurements?
  - Correcting erroneous flow values. How do you correct the erroneous flow values?
- **Participation:** we have 7 teams (students coming from our data science class) participated in this challenge, but only have two valid submissions due to the high difficulty level of this problem.

Lane Measurement Data

1. Lane_measurements
   a. detector_lane_inventory.csv
      i. lane_id: uniquely identify a detector (totally 2,135).
      ii. zone_id: identifier of a zone in a road.
      iii. road: on which road, e.g. I-66.
      iv. location_description: e.g. I-66 NEAR Sudley Rd @ MM 49.02
      v. Geographical coordinate: (latitude, longitude)
      vi. There are 11 other fields that may not be of our interest in the original data.



lane and zone illustration (courtesy by NIST)



Detector distribution (courtesy by Sreten Cvetojevic)

How we Evaluated?

The score is the **average absolute error** found in the cleaned data:

$$score = \frac{\sum_{i=1}^{card(r)} |cleaned(fl_i) - true(fl_i)|}{card(r)}$$

- $card(r)$ is the total number of lane detector measurements in a test data set.

For each lane detector measurement i in $card(r)$:

- $cleaned(fl_i)$ is the estimated traffic flow for measurement i.
- $true(fl_i)$ is the correct traffic flow for measurement i.

Cleaning Task, System ufdsrA

Score (average absolute error): 0.4007463

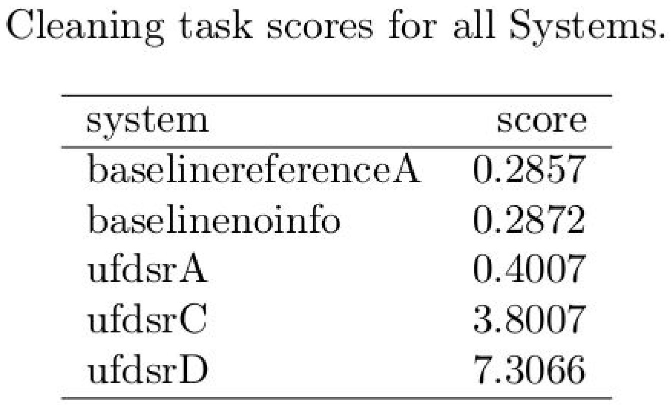
Total absolute error: $1.4581635*10^9$

Number of measurements scored on: $3.6386198*10^9$

Cleaning Task

a. clean traffic lane detector measurements containing incorrect flow values, providing correct traffic flow values for the erroneous traffic flow measurements.

b. We have submitted three systems for this task, namely ufdsrA, ufdsrC, and ufdsrD.



Cleaning task scores for all Systems.

system	score
baselinereferenceA	0.2857
baselinenoinfo	0.2872
ufdsrA	0.4007
ufdsrC	3.8007
ufdsrD	7.3066

Cleaning Task Scores of Systems

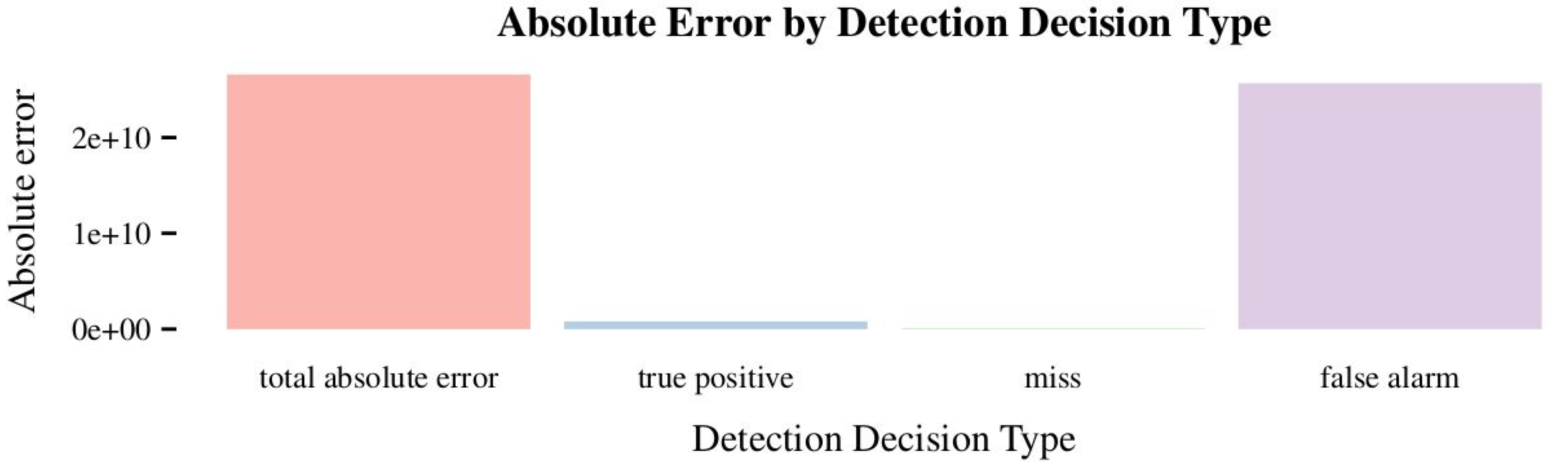
Error scores (smaller is better)

- **baselinereferenceA**: return the median of neighboring flow values if the current flow value differs too more from the median.
- **baselinenoinfo**: not changing anything.
- ufdsrA: linear regression model, no weather or street map data was used.
- ufdsrD: the output format was incorrect.
- ufdsrC: the merge (mean) of A and D.

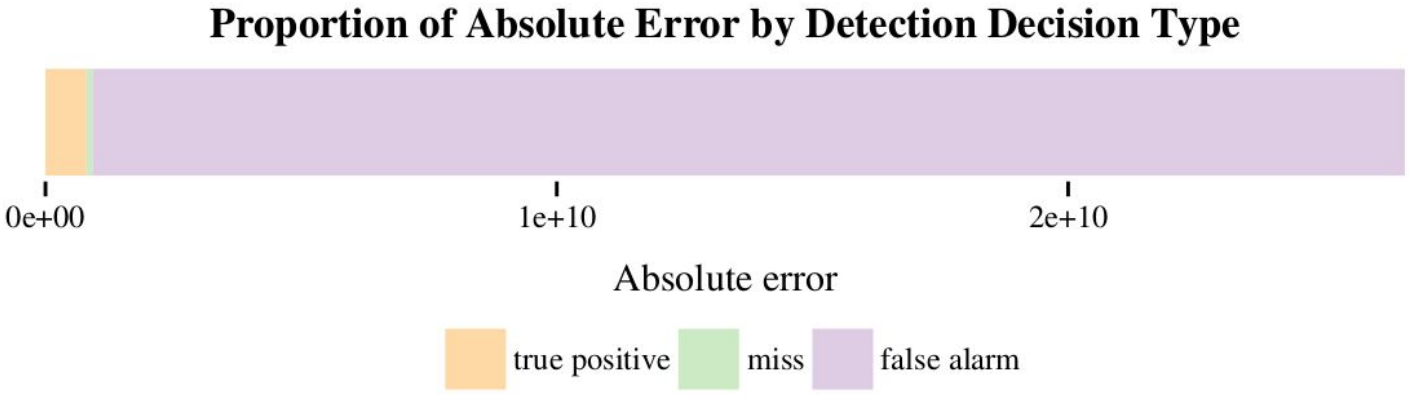
Cleaning Task - System ufdsrA

Authors: Mebin Jacob, Pranav Achanta, Rui Zhang, Sandeep Dommaraju, Sudeep Reddy, Vipul Mittal

- ❏ Detecting incorrect flow values.
  - ❏ Negative flow/speed values.
  - ❏ Spatial or temporal smoothness constraints: values > 2*std of a sliding window.
- ❏ Predicting the values for invalid data: mean value of k nearest neighbours.



Absolute Error by Detection Decision Type



Proportion of Absolute Error by Detection Decision Type

What we have learned

- Data set
  - Around 5% of the flow values were incorrect.
  - The correct flow values was added noises with mean = 8.8 and std = 6.6.
- We need to improve how to determine when a flow value should be changed. We have made around 21% changes (too aggresive), but actually there are only 5% values need to be changed. That’s why our system cannot beat the baseline (keep everything unchanged).
- “keeping everything unchanged” performs equally with “smoothing” (reduced 0.5% errors). It reveals nearby values do not provide much useful information in this task!