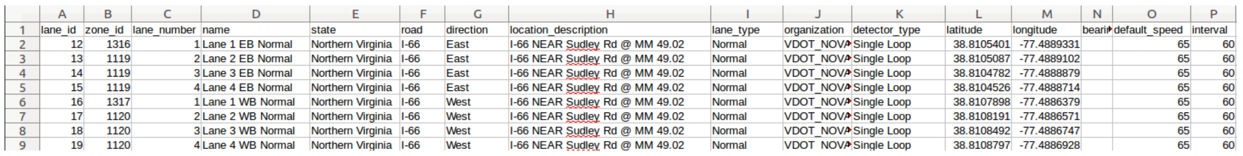
Pre-Pilot Evaluation Guideline

1. Cleaning Task (50%)

Description: participants are asked to clean traffic lane detector measurements containing incorrect flow values, providing correct traffic flow values for the erroneous traffic flow measurements.
Input: traffic lane detector measurements with erroneous traffic flow values and speeds (in number and average speed of vehicles since the last interval).
Output: cleaned traffic lane detector measurements with cleaned traffic flow values (in number of vehicles since the last interval).
Training data: N/A
Development data:

- core/lane_measurements/detector_inventory.csv
    - Description: providing detector information, including detectors' lane id and zone id.
    - Snapshot:

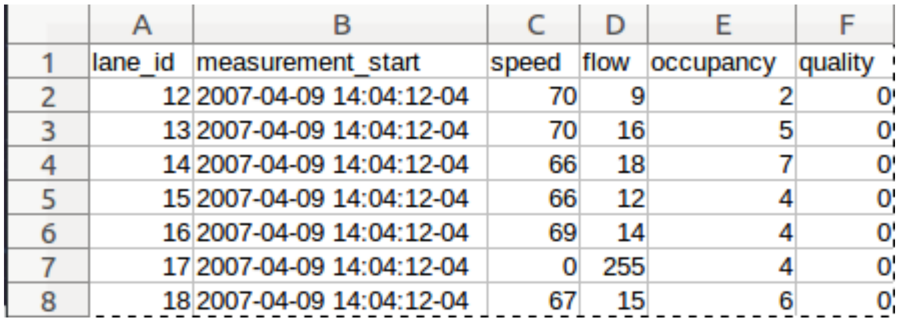


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	lane_id	zone_id	lane_number	name	state	road	direction	location_description	lane_type	organization	detector_type	latitude	longitude	bearir	default_speed	interval
2	12	1316	1	Lane 1 EB Normal	Northern Virginia	I-66	East	I-66 NEAR Sudley Rd @ MM 49.02	Normal	VDOT_NOVA	Single Loop	38.8105401	-77.4889331		65	60
3	13	1119	2	Lane 2 EB Normal	Northern Virginia	I-66	East	I-66 NEAR Sudley Rd @ MM 49.02	Normal	VDOT_NOVA	Single Loop	38.8105087	-77.4889102		65	60
4	14	1119	3	Lane 3 EB Normal	Northern Virginia	I-66	East	I-66 NEAR Sudley Rd @ MM 49.02	Normal	VDOT_NOVA	Single Loop	38.8104782	-77.4888879		65	60
5	15	1119	4	Lane 4 EB Normal	Northern Virginia	I-66	East	I-66 NEAR Sudley Rd @ MM 49.02	Normal	VDOT_NOVA	Single Loop	38.8104526	-77.4888714		65	60
6	16	1317	1	Lane 1 WB Normal	Northern Virginia	I-66	West	I-66 NEAR Sudley Rd @ MM 49.02	Normal	VDOT_NOVA	Single Loop	38.8107898	-77.4886379		65	60
7	17	1120	2	Lane 2 WB Normal	Northern Virginia	I-66	West	I-66 NEAR Sudley Rd @ MM 49.02	Normal	VDOT_NOVA	Single Loop	38.8108191	-77.4886571		65	60
8	18	1120	3	Lane 3 WB Normal	Northern Virginia	I-66	West	I-66 NEAR Sudley Rd @ MM 49.02	Normal	VDOT_NOVA	Single Loop	38.8108492	-77.4886747		65	60
9	19	1120	4	Lane 4 WB Normal	Northern Virginia	I-66	West	I-66 NEAR Sudley Rd @ MM 49.02	Normal	VDOT_NOVA	Single Loop	38.8108797	-77.4886928		65	60

- OpenStreetMap

Testing data:

- Location: core/lane_measurements/test/cleaning_test_yy_mm.csv
- Number of files: 108
- Total size: approx. 100GB.
- Snapshot:



	A	B	C	D	E	F
1	lane_id	measurement_start	speed	flow	occupancy	quality
2	12	2007-04-09 14:04:12-04	70	9	2	0
3	13	2007-04-09 14:04:12-04	70	16	5	0
4	14	2007-04-09 14:04:12-04	66	18	7	0
5	15	2007-04-09 14:04:12-04	66	12	4	0
6	16	2007-04-09 14:04:12-04	69	14	4	0
7	17	2007-04-09 14:04:12-04	0	255	4	0
8	18	2007-04-09 14:04:12-04	67	15	6	0

Extra comments:

- The only data available to you for this task include both *development data* and *testing data*.
- Both "speed" and "flow" (please ignore "occupancy" or "quality" fields) values can have errors.
- You make use of the given testing data (lane_id, measurement_start, speed and flow) and optionally development data, then output the correct "flow" value for each row of testing data.

Hints:

- Extraneous flow values usually violate some constraints.
    - In a period of time, flows in the same zone of different lanes should have similar values.

    - In a period of time, flows of nearby zones should be similar.
    - Flow values must be nonnegative numbers.
    - Measurements of (flow, speed, occupancy) should be consistent.
    - Your own constraints.
- How to correct extraneous flow values.
    - Correct the values as values of nearby location or time.
    - Correct the values as output of a regression model F(location,time) -> flows. The F is learned from the dirty data.
- Suggestions:
    - Walk through part of the testing data manually and see where errors can occur, and try to conclude useful patterns that can detect errors.

How we evaluate your work:

- Your work will be evaluated based on the following criteria:
    - Accuracy of your system:
        - accuracy is defined as: $p = \frac{\sum_{f \in S} |true(f) - predicted(f)|}{|S|}$, where S is a set of all flow entries in the test data.
    - Your implementation framework:
        - how you designed your mapreduce framework.
        - how you store the bulky data for efficient access (e.g. distributed access, minimum data transfer, in-memory caching, indexing, etc).
        - how the system pipeline facilitates solving the problem (e.g. minimize the number of passes of the data, design end-to-end system without much manual work to redirect input/output in between).
    - Quality of the source codes:
        - easy read and maintain.
        - specify/include necessary dependencies.
        - specify necessary system configurations in order to run the codes.

2. Prediction Task (50%)

Description: participants will develop a system that can predict the number and types of traffic events by type for a given (geographical bounding, interval of time) pair.

Input: geographical bounding boxes and time intervals.

Output: predicted counts for each specified type of traffic event.

Training data:

- All the common core data described in Section II (see the evaluation plan) will be available for training and development purposes.

Testing data:

- The test data consists of a series of trials, where each trial is a (location, time interval) pair. The location is given as decimal latitude and longitude coordinates that define a geographical bounding box, and time interval is given as start and end of the time window.
- See the evaluation plan for the required system output format, and how performance is measured.

Hints:

1. Framework.
    - For a given geographical bounding box bb, identify all major road segments contained withinbb, using OpenStreetMap. Denote set of road segments as S.
    - For the *k-th* segment $s_k$in S, predict the total number of events for each event type will occur within the given time interval t_i (interval is always 1 month). Denote predicted number of events will occur for the *i-th* event type ei asn_{ik}.
    - Predict the total number of occurrences for events of the *i-th* event type within given bb and ti as: $y_i = \sum_k \quad n_{ik}$
2. Predict #events of type event type e for a road segment s within time interval t.
    - Extract set of road features from segment s. The road features can be:
        - length of the road segment.
        - #lanes of the road segment.
        - month when the measurement starts (e.g. Jan, Feb, …, Dec).
    - Predict #events as output of $H_e(set\ of\ road\ features) \rightarrow \#events.$
        - The $H_e$is a regression model learnt from training data.
        - The subscript e denotes “event”, which implies we should train one model for per event.
3. Train a regression model.
    - Construct training data.
        - Identify a set of reasonable road features that your can obtain during both training and testing.
        - Extract a series of (road features, #events) as training data, where #events is regression target.
    - Select a regression model.
        - Polynomial regression
        - Gaussian Process for regression

How we evaluate your work:

- Your work will be evaluated based on the following criteria:
    - Accuracy of your system:
        - accuracy of an event prediction is defined as: $p = \frac{\sum_{b \in S} \quad |true(b) - predicted(b)|}{|S|}$, where S is a set of testing cases (bounding boxes and time intervals).
    - Your implementation framework:
        - how you designed your mapreduce framework.
        - how you store the bulky data for efficient access (e.g. distributed access, minimum data transfer, in-memory caching, indexing, etc).
        - how the system pipeline facilitates solving the problem (e.g. minimize the number of passes of the data, design end-to-end system without much manual work to redirect input/output in between).
    - Quality of the source codes:
        - easy read and maintain.
        - specify/include necessary dependencies.

- specify necessary system configurations in order to run the codes.

3. Logistics

Where to find data: data will be available from an AWS S3 bucket at https://s3.amazonaws.com/nist-intro-data-science to synchronize with your own AWS account you can use the following command. `aws s3 sync <source> <target> [--options]` This will create a copy of the NIST lane measurement data with the folder structure described before. (you will need to install AWS CLI)

Exploratory data analysis task (Bonus 5%): as a first task to get you started (and hopefully excited), we want you to perform some exploratory data analysis on the given data (or a subset of it). After you are able to perform some computation of the downloaded files, we want you to eyeball it, run some code scripts and come up with some interesting insights/statistics about the data. Good ideas for this task can be to generate some sort of visualization. Your findings must be shared via Post on ***piazza*** and will be taken into account in chronological order. We expect to see no repetitions during this phase, and posts will be public so every group can benefit from everybody's analysis.

First Submission (5%): For the first deadline, you will need to submit your preliminary cleaning results. You will need to submit one **README** file, an informal description of the methods, tools, and pipelines used to develop your system. System output results must be provided in multiple files, one per test data file, using standard ASCII encoding. For every test file ***cleaning_test_yy_mm.csv***, the corresponding submission file shall be named ***cleaning_subm_yy_mm_{team}.txt*** where *team* is your team ID (nist1, nist2, etc) file names should be all lower case. Every submission file will have the same number of lines in the same order as the corresponding test file, and consist of three **tab** separated attributes per line: The first attribute is binary {0,1}. 1 represents that the corresponding measure in the test file is correct and 0 the corresponding measure in the test file is incorrect. The second attribute is a real number: cleaned flow, i.e., the corrected traffic flow for the corresponding measurement. Every line must have both attributes, if the first attribute is 1, the second attribute must match the corresponding measure in the test file. The third attribute will only be present in lines with first attribute value 0 and consists of a short description of why this attribute was marked incorrect. This should be done programmatically by assigning error codes. If you use error codes please attach a file with error code meanings.

Second Submission (5%): For the second deadline you are required to submit your preliminary prediction results. You will need to submit one **README** file, an informal description of the methods, tools, and pipelines used to develop your system. System output results must be provided in a single file using standard ASCII encoding and named ***prediction_submissions_{team}.txt.*** The output for each trial, i.e., (location, time interval) pair, must consist of a single line and be given in the same order as in the test data file. Each line must consist of a list of tab separated real values corresponding to the estimated number of events per type and ordered as follows: Accidents and Incidents, Roadwork, Precipitation, Device Status, Obstruction, Traffic Conditions.

Third Submission (5%): Cleaned lane measurement data will be distributed, and for this deadline you are asked to re-run you prediction pipeline using the new dataset. The output format is the same as second submission.

Final presentation (10%): You need to present the improvements you worked on after your preliminary results and how you achieved them. Additional details about final presentation and write up will be announced later.

Note: The size of the output files might get lengthy to submit using canvas. If this is the case please point us to an AWS bucket where we can download your results by the deadline.

Deadlines

Exploratory task: Monday, October 19th. 3:00 p.m. (Before Class)
First Submission: Sunday, October 25th.
Second Submission: Wednesday, October 28th.
Third Submission: Friday, November 6th.
End-of-Semester Presentation: TBD