

THE DOWNLOAD

TECH TALKS BY HPCC SYSTEMS




The Download: Community Tech Talks Episode 15

June 28, 2018



Welcome!

- Please share: Let others know you are here with #HPCCTechTalks 
- Ask questions! We will answer as many questions as we can following each speaker.
- Look for polls at the bottom of your screen. Exit full-screen mode or refresh your screen if you don't see them.
- We welcome your feedback - please rate us before you leave today and visit our [blog](#) for information after the event.
- Want to be one of our featured speakers? Let us know! techtalks@hpccsystems.com

Community announcements

- HPCC Systems® Platform updates
 - 6.4.20-1 Gold available
 - 7.0.0-beta2 available
 - Better performance and usability
 - New ECL language and library features
 - WsSQL now integrated into the platform
 - Spark-HPCC Systems Connector to read Thor files natively
 - Download today – we need feedback!
- Latest Blogs
 - [Systemd – Easier management of your HPCC Systems components](#)
 - [First Look - HPCC Systems log visualizations using ELK](#)
 - [HPCC Systems 7.0.0 beta release - Try it now!](#)
- **Reminder:** 2018 HPCC Systems Community Day, Atlanta
 - We need speakers! CFP deadline on July 20.
 - Sponsor packages still available
 - Workshop & Poster Competition on October 8
 - Main event on October 9
 - Visit hpccsystems.com/hpccsummit2018



Dr. Flavio Villanustre

VP Technology

RELX Distinguished Technologist

LexisNexis® Risk Solutions

Flavio.Villanustre@lexisnexisrisk.com

2018 HPCC Systems
Community Day

Call for Presentations
& Poster Abstracts
Now Open



Today's speakers



Jingqing Zhang

PhD Candidate

Data Science Institute

Imperial College London

jingqing.zhang15@imperial.ac.uk

Imperial College
London

Jingqing Zhang is a 1st-year PhD (HiPEDS) at Department of Computing, Imperial College London under supervision of Prof. Yi-Ke Guo. His research interest includes Text Mining, Data Mining, Deep Learning and their applications. He received his MRes degree in Computing from Imperial College with Distinction in 2017 and BEng in Computer Science and Technology from Tsinghua University in 2016.



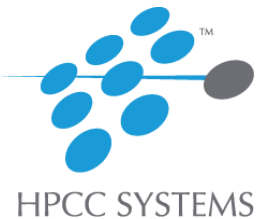
Bob Foreman

Senior Software Engineer

LexisNexis Risk Solutions

Robert.Foreman@lexisnexisrisk.com

Bob Foreman has worked with the HPCC Systems technology platform and the ECL programming language for over 5 years, and has been a technical trainer for over 25 years. He is the developer and designer of the HPCC Systems Online Training Courses, and is the Senior Instructor for all classroom and Webex/Lync based training.



THE DOWNLOAD

TECH TALKS BY HPCC SYSTEMS

Deep Sequence Learning in Traffic Prediction and Text Classification



Jingqing Zhang
PhD Candidate
Data Science Institute
Imperial College London



Imperial College
London

Background

- **Sequential data** is everywhere in our daily life: e.g. text, video, stock, etc.
- **Sequence learning** is a fundamental human ability. Infants may use such ability to learn skills. It can be explicit and implicit.
- The sequence itself can be very informative. Early research has exploited temporal pattern of sequence, such as ARMA family, SVR, Gaussian Process.
- However, in practical scenarios, the information of sequence itself is usually not enough and **background knowledge** can be important.

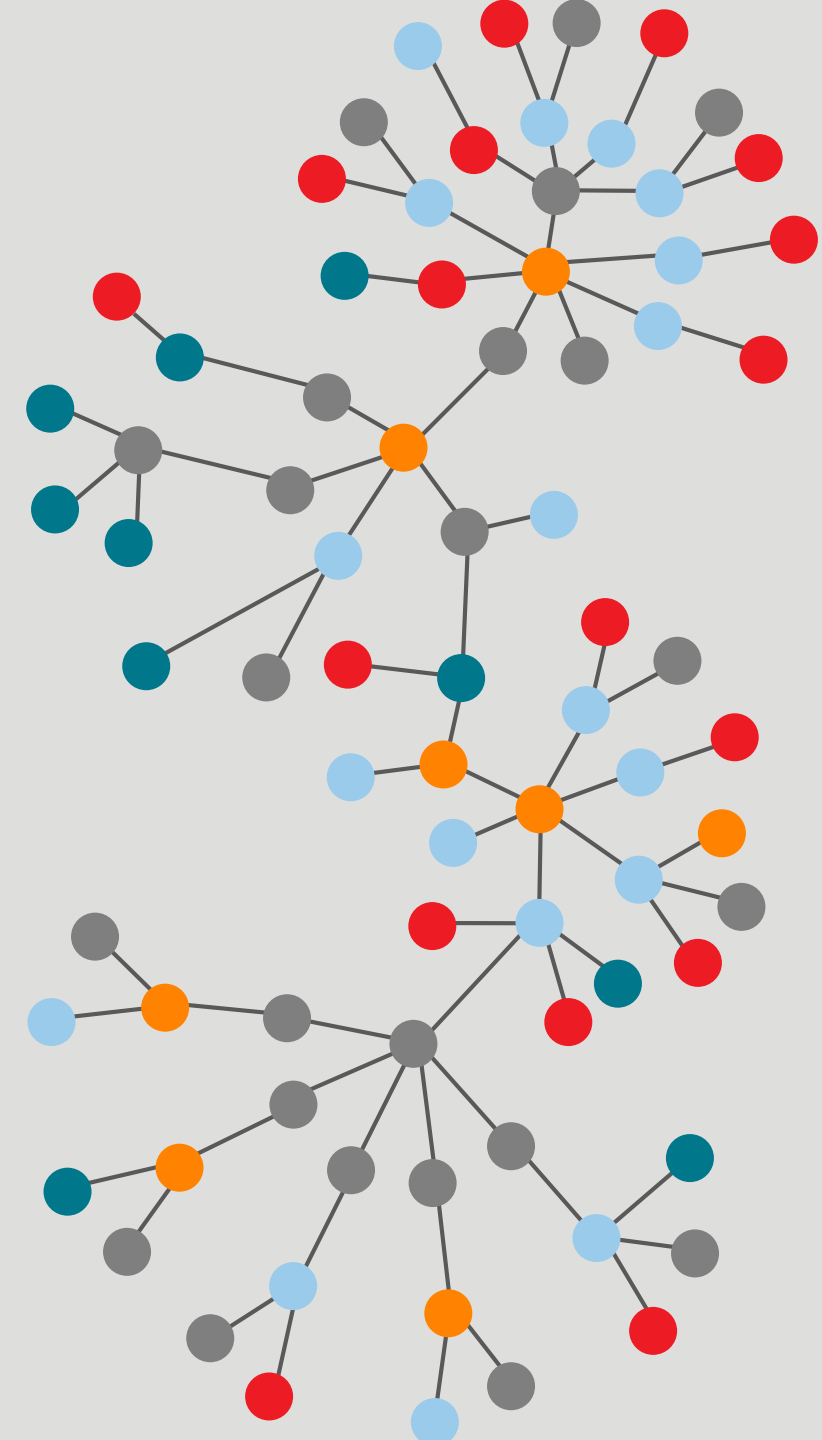
Outline

- Background
- Traffic Prediction with Auxiliary Information
- Zero-shot Text Classification with Knowledge Graph
- TensorLayer and HPCC Systems

Quick poll:

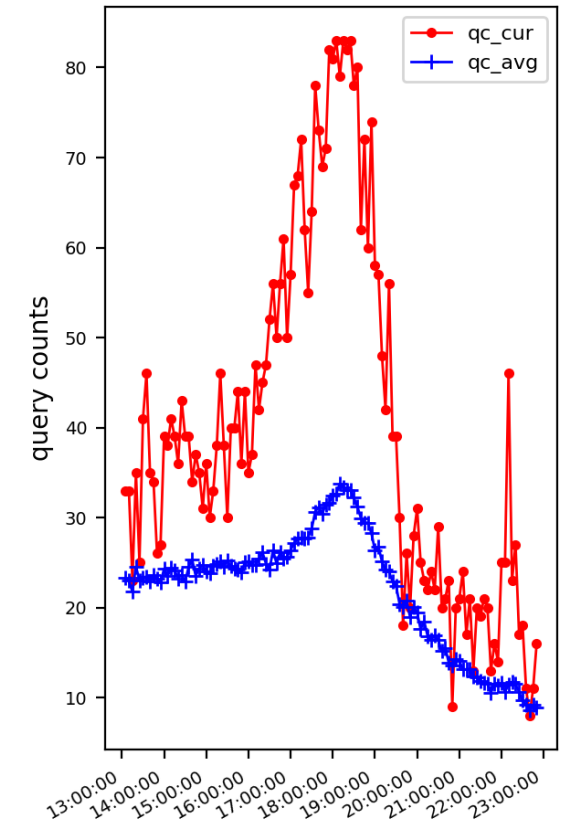
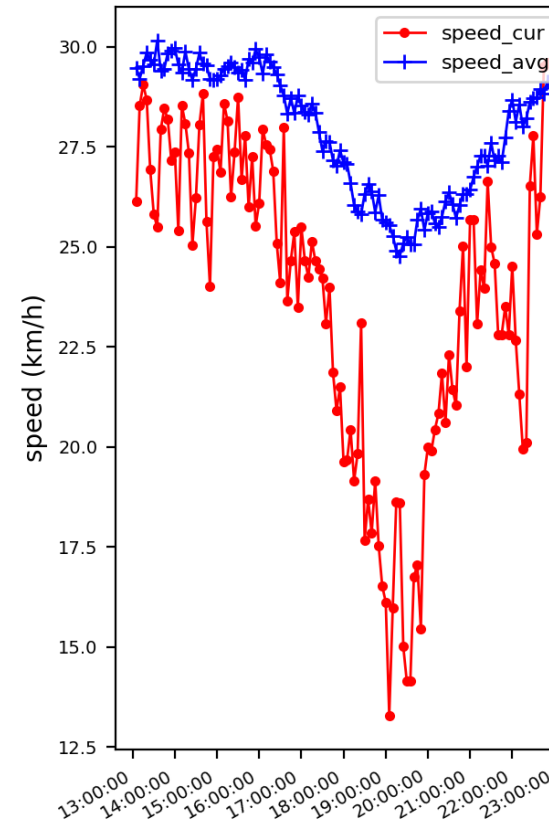
Would you use Google Maps (or any other map app) to search the destination you will go before departure?

See poll on bottom of presentation screen



Traffic Prediction with Auxiliary Information

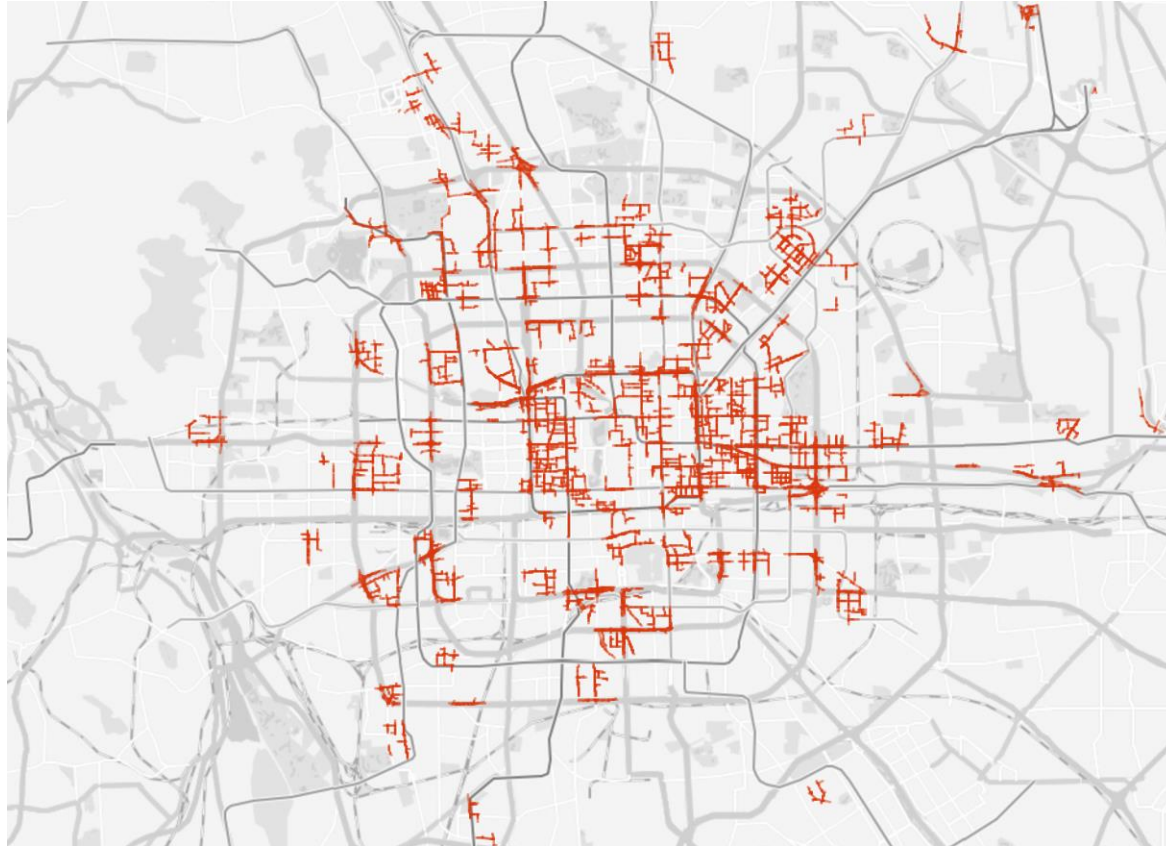
- Accurate traffic prediction is essential for a smart city especially huge cities
- Motivations and Challenges:
 - Limited dataset
 - Offline factors:
 - road network, public holidays
 - Online factors:
 - map app search queries



Traffic Prediction with Auxiliary Information (Cont.)

- Our solutions
 - Released a large-scale traffic dataset: **Q-Traffic**
 - Used Wide and Deep Network to model **offline attributes**
 - Used Graph CNN to model **spatial dependencies**
 - Devised an algorithm to calculate impact of **online search queries**
 - Proposed a hybrid model to incorporate **all auxiliary information**
- Our recent work on KDD'18: Deep Sequence Learning with Auxiliary Information for Traffic Prediction
- <https://github.com/JingqingZ/BaiduTraffic>

Q-Traffic Dataset



- Distribution of road segments, Beijing, Baidu Map App

Table 5: Comparison of different datasets for traffic speed prediction.

Datasets	Scale	Road info.	Road net.	Auxiliary info.	Highway	Urban	Available
Subset of PeMS							
State Route 22, Garden Grove [37]	9						
PeMSD7 (S) [38]	228		√		√		√
San Francisco Bay area [16]	943						
PeMSD7 (L) [38]	1,026						
Subset of Beijing							
Ring road around Beijing [26]	2					√	
Beijing 4th ring road [33]	3					√	
Beijing 2nd/3rd ring road [36]	80		√		√		
Beijing 2nd/3rd ring road [36]	122				√		
Beijing taxi dataset [25]	236				√	√	
Beijing taxi dataset [25]	352				√	√	
I-80 in California [11]	6		√		√		√
Busan Metropolitan City [20]	10		√			√	
California PATH [4]	12				√		
Corridor in Orlando [30]	71				√		
Rome dataset [12]	120		√			√	
D100 [14]	122			weather		√	
Bedok area [8]	226		√		√	√	
Los Angeles [9]	1,642		√		√	√	
Los Angeles [9]	4,048		√		√	√	
Dallas-Forth Worth area [15]	4,764				√		
Subnetwork in Singapore [3]	5,024		√		√	√	
Q-Traffic Dataset	15,073	√	√	map query	√	√	√

Discovery of Events by Query Records

- It is assumed that query can help to predict traffic in the region affected by public events.
- A quick boost of search query in a short period of time is a key indicator to discover events.
- Several kinds of events are discovered, including concerts, forums, places of interest and anniversaries.

Definition 2.1 (Moment). A tuple $m = (x, y, t)$ is a moment if

$$d_{x,y,t-\Delta t} > 0 \quad (2)$$

$$d_{x,y,t} - d_{x,y,t-\Delta t} > \zeta \quad (3)$$

$$\frac{d_{x,y,t} - d_{x,y,t-\Delta t}}{d_{x,y,t-\Delta t}} > \eta \quad (4)$$

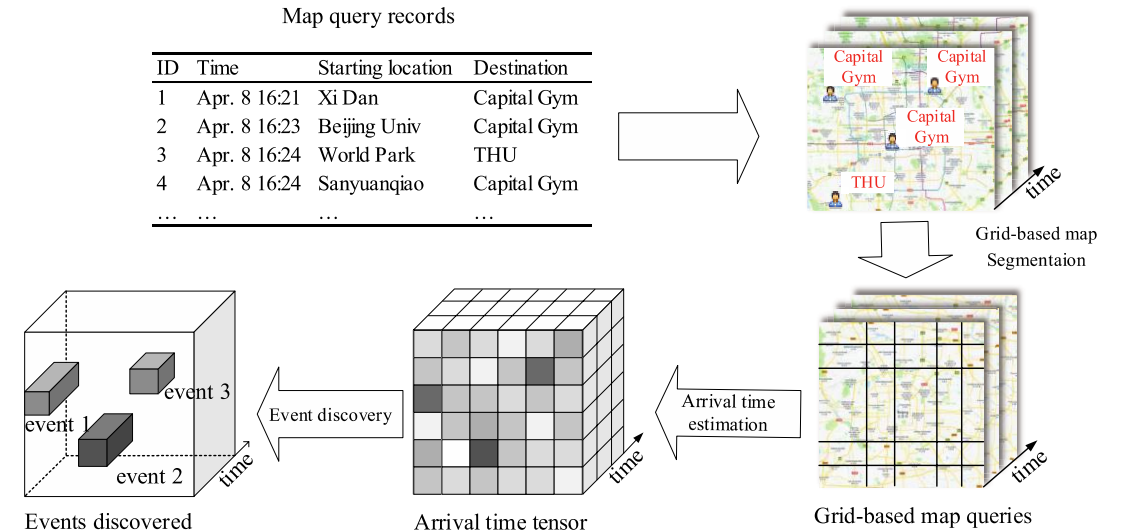
We denote \mathcal{M} a set of all moments.

Definition 2.2 (Event). A tuple $E = (x, y, t_s, t_d)$ is an event if

$$t_d - t_s > \epsilon \quad (5)$$

$$\forall t \in [t_s, t_d], \quad m = (x, y, t) \in \mathcal{M} \quad (6)$$

$$m = (x, y, t_s - 1) \notin \mathcal{M} \wedge m = (x, y, t_d + 1) \notin \mathcal{M} \quad (7)$$



Modelling of Query Impact

- The query counts have a clear correlation with the traffic speed.
- The query impact QI is defined to measure the influence of queries on road segments.
- It is calculated based on the query counts and the spatial region that the query will influence.

Algorithm 1: QUERYIMPACT Calculate the query impact

Input: A set $Q = \{q^1, q^2, \dots, q^n\}$ of queries, where $q^i = (t_s^i, t_d^i, s^i, d^i, x_s^i, y_s^i, x_d^i, y_d^i)$, a set

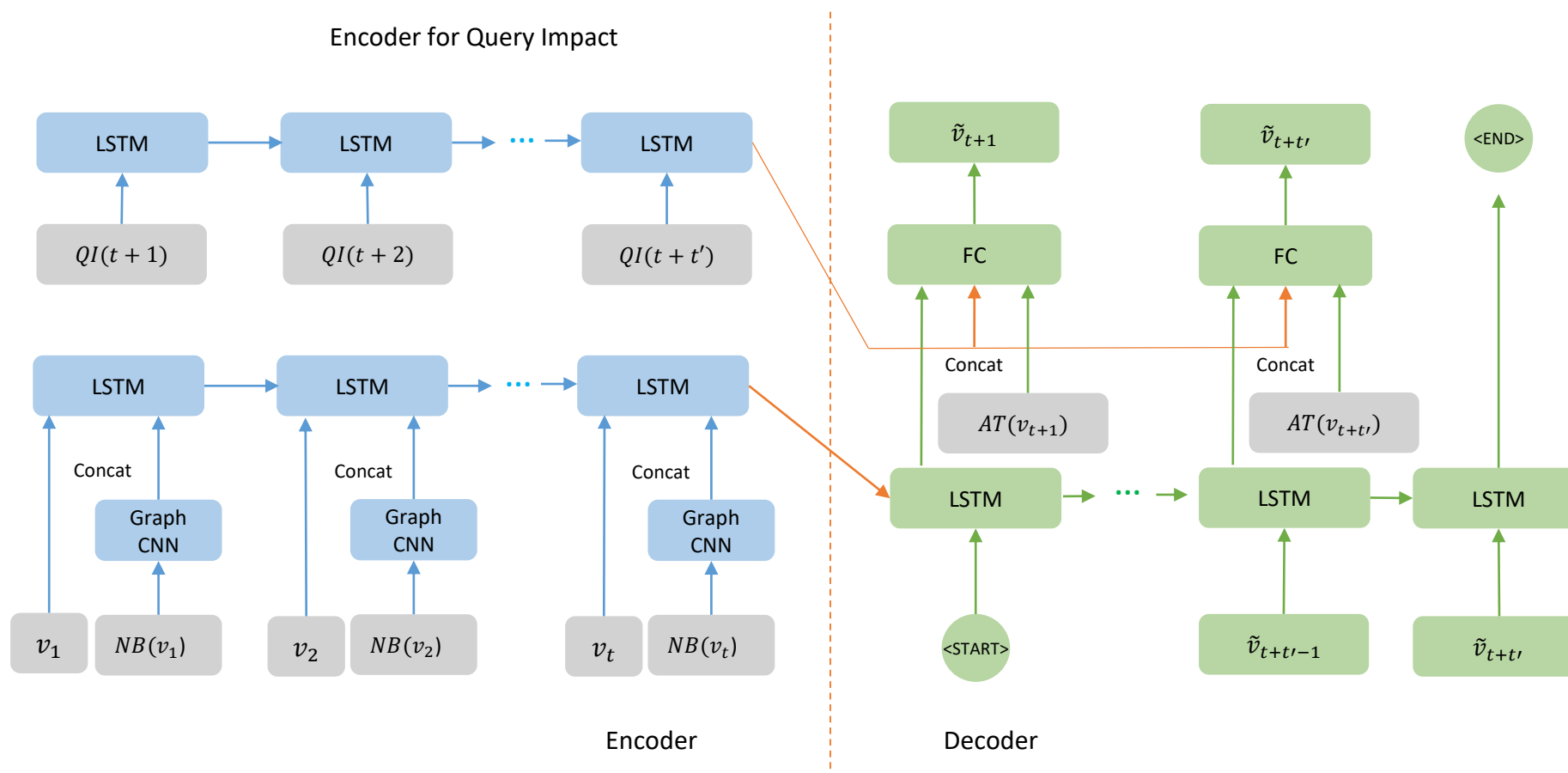
$RS = \{l^1, l^2, \dots, l^k\}$ of links, total time stamp T

Output: The query impact QI

```
1 Initialisation:  $QI(l, t) \leftarrow 0$ 
2 for  $i \leftarrow 1$  to  $n$  do
3   // return the longitude and latitude of a location
4    $lonlat_s \leftarrow lonlat(s^i)$ 
5    $lonlat_d \leftarrow lonlat(d^i)$ 
6    $seg_i \leftarrow segment(lonlat_s, lonlat_d)$ 
7   // return the set of road segments within 1km
8    $L \leftarrow nearroad(lonlat_d)$ 
9   for each  $l \in L$  do
10     $lonlat_l \leftarrow lonlat(l)$ 
11     $d_l \leftarrow dist(lonlat_l, seg_i)$ 
12     $QI(l, t_d^i) \leftarrow QI(l, t_s^i) + h(d_l)$ 
13 return  $QI$ 
```

- $h = \exp(-\frac{x}{\sigma})$

Hybrid Model



Results

Table 6: Err_T (%): MAPE on the whole testing set. The results with the best performance are marked in bold.

Prediction	15-min	30-min	45-min	60-min	75-min	90-min	105-min	120-min	Overall
RF	6.00	9.15	10.20	10.66	10.98	11.21	11.39	11.56	10.14
SVR	5.44	9.20	10.07	10.34	10.51	10.65	10.76	10.83	9.73
Seq2Seq	4.61	8.22	9.28	9.72	9.98	10.27	10.48	10.61	9.23
Seq2Seq+AT	4.53	8.06	9.09	9.48	9.70	9.84	9.93	10.01	8.83
Seq2Seq+NB	4.52	8.05	9.07	9.45	9.67	9.83	9.93	9.99	8.81
Seq2Seq+QI	4.58	8.01	8.95	9.31	9.51	9.66	9.80	9.94	8.72
Hybrid	4.52	7.93	8.89	9.24	9.43	9.56	9.69	9.78	8.63

Table 7: Err_E (%): MAPE during events on the testing set. The results with the best performance are marked in bold.

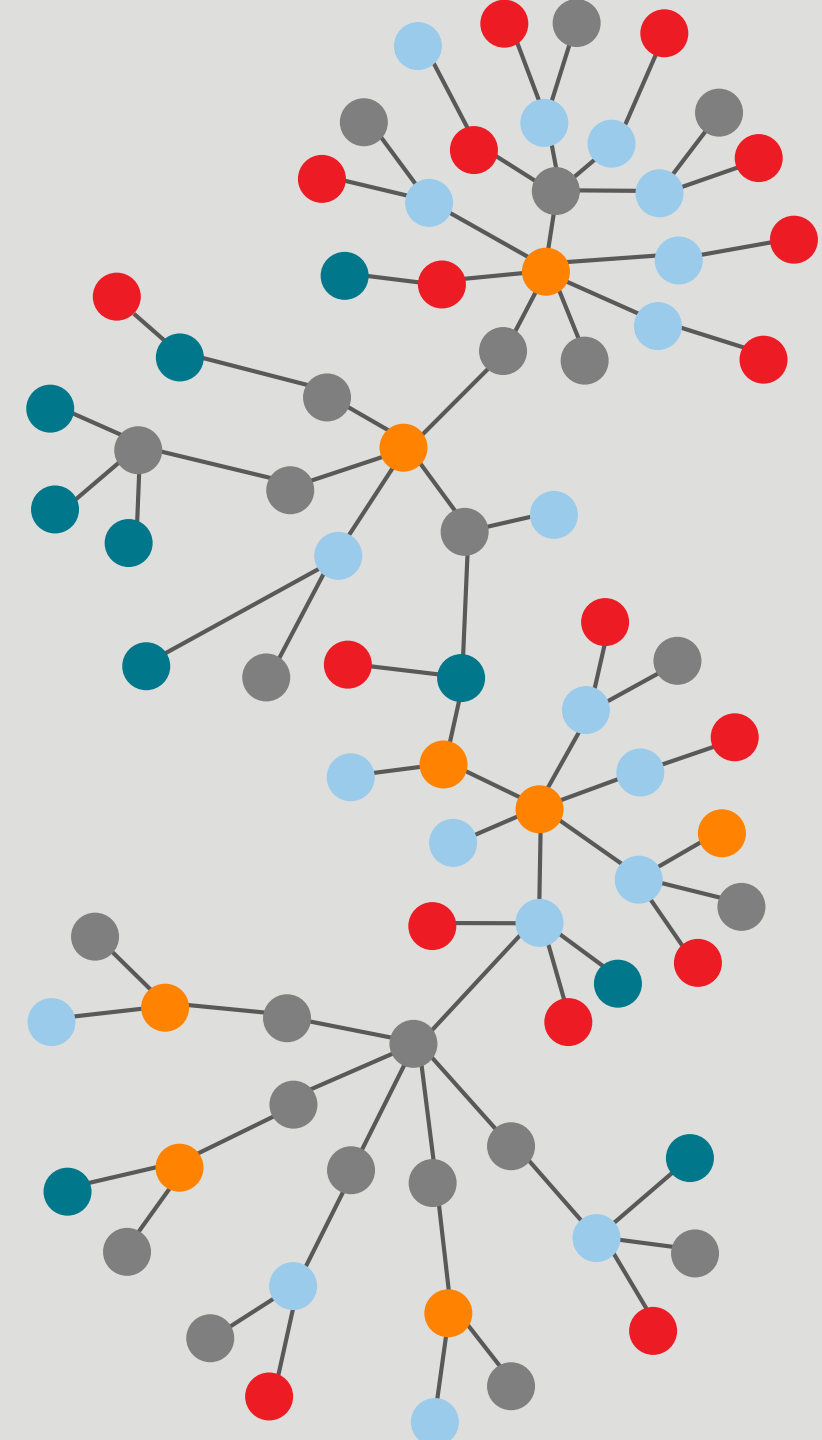
Prediction	15-min	30-min	45-min	60-min	75-min	90-min	105-min	120-min	Overall
RF	6.14	9.51	10.81	11.45	11.84	12.13	12.38	12.56	10.85
SVR	5.64	9.56	10.59	11.02	11.32	11.56	11.73	11.83	10.41
Seq2Seq	4.76	8.52	9.87	10.52	10.91	11.31	11.60	11.80	9.91
Seq2Seq+AT	4.65	8.32	9.63	10.23	10.58	10.81	10.98	11.13	9.54
Seq2Seq+NB	4.63	8.25	9.53	10.10	10.45	10.70	10.89	11.02	9.45
Seq2Seq+QI	4.69	8.18	9.37	9.93	10.28	10.55	10.77	10.98	9.34
Hybrid	4.61	8.09	9.30	9.84	10.16	10.39	10.60	10.76	9.22

Quick poll:

Which category does the following paragraph belong to?

“This removes bulky ions, such as magnesium and sulphate, as well as bacteria and other large particles.” – The Economist

See poll on bottom of presentation screen



Zero-shot Text Classification with Knowledge Graph

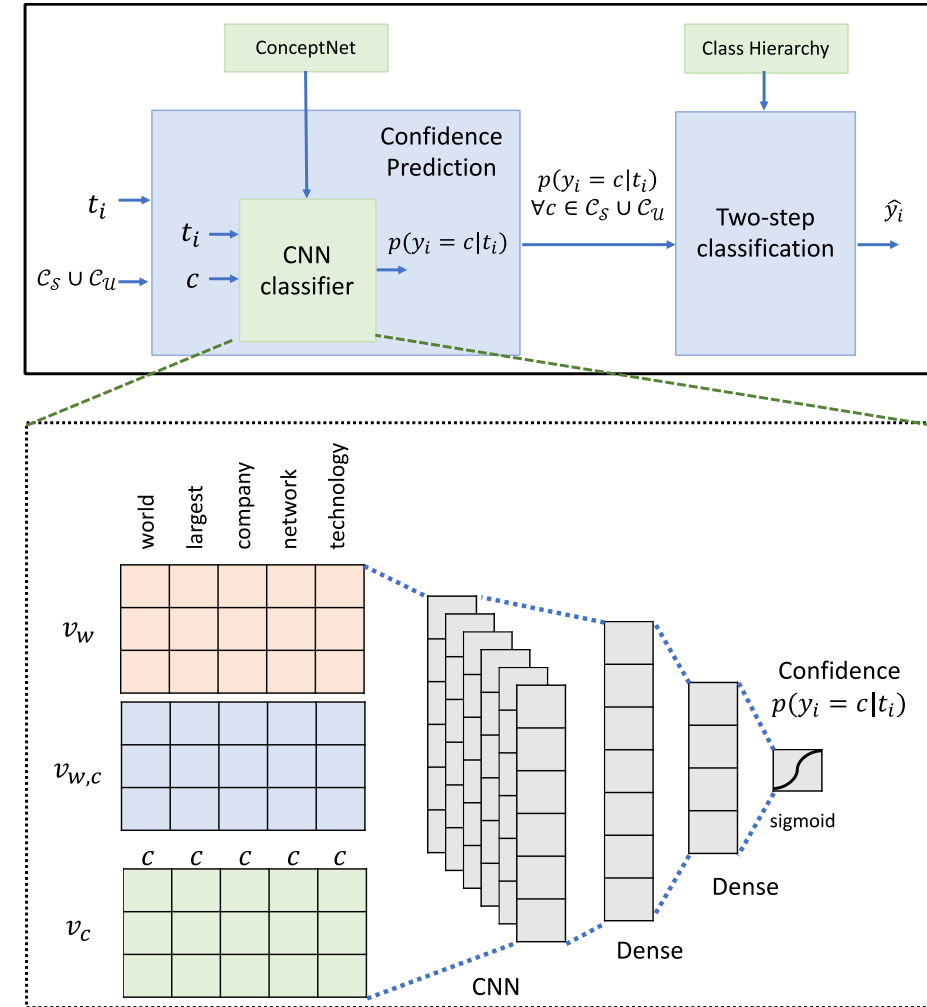
- Traditional assumption of closed-world classification – the classes in testing data must be seen during training – cannot adapt to dynamic open world.
- Insufficient training data of rare and/or emerging classes is a big challenge of many classification tasks.
- Human can identify which class a document belongs to because of background knowledge. For example, some keywords can clearly indicate a class.

Zero-shot Text Classification with Knowledge Graph (Cont.)

- Problem Definition:
 - Set of seen classes \mathcal{C}_S and unseen classes \mathcal{C}_U
 - Training set: $\{(t_1, y_1), (t_2, y_2), \dots\}$, $y_i \in \mathcal{C}_S$
 - Testing set: same format as the training set, except $y_i \in \mathcal{C}_S \cup \mathcal{C}_U$
- Our solutions:
 - Used ConceptNet, a knowledge graph of general human knowledge

Framework

- Phrase 1: Confidence Prediction
 - Word vector v_w v_c : GloVe
 - Relation vector $v_{w,c}$: path from w to c in ConceptNet
- Phrase 2: Two-step classification
 - Confidence on seen classes is more reliable
 - Check if t_i belongs to any of the seen classes or not
 - $\operatorname{argmax} p(y_i = c|t_i), c \in \mathcal{C}_S$
 - If not, predict an unseen class that t_i belongs to
 - using class hierarchy
 - $\operatorname{argmax} \sum_{c' \in \mathcal{C}_S \cup \{c\}} w(c, c') p(y_i = c|t_i), c \in \mathcal{C}_U$



Results

Table 1: The comparison between different settings. We use the accuracy $\frac{1}{n} \sum_{i=1}^n \mathbb{I}(\hat{y}_i = y_i)$ to evaluate.

Inputs & Classification Policy	$[v_w; v_c]$			$[v_w; v_{w,c}]$			$[v_w; v_c; v_{w,c}]$		
	seen	unseen	overall	seen	unseen	overall	seen	unseen	overall
1-step	98.2	11.1	79.5	94.5	18.8	79.3	97.0	17.6	80.0
2-step / $p(y_i = c_u t_i)$	97.6	22.0	81.4	93.1	36.8	81.1	93.9	39.8	82.3
2-step / $s(y_i = c_u)$	97.6	22.3	81.5	93.1	39.8	81.7	93.9	40.8	82.5

- Dataset: DBPedia ontology dataset ^[1]
- Using **knowledge graph** significantly helps the framework classify instances of unseen classes correctly. Using $[v_w; v_c; v_{w,c}]$ produces the best results.
- **Two-step classification** slightly undermines the accuracy of classifying seen classes but doubles the accuracy on unseen classes and results in the better accuracy overall.
- Using **the new confidence measure** which uses class hierarchy improves the accuracy of unseen and overall, but insignificantly.
- This project is still in progress.

TensorLayer and HPCC Systems

- **Deep Learning** has enabled many AI systems to exhibit unprecedented performance even beyond human, such as natural language processing, computer vision, medical imaging, gaming, etc.
- The efficient **development tools** are essential for the success of deep learning besides datasets, algorithms and hardware.
- **TensorFlow** is one of the most popular and powerful tool to develop deep neural networks but also **hard to use**.
- **TensorLayer** is a high-level wrapper of TensorFlow and naturally supports low-level APIs of TensorFlow.
- <https://github.com/tensorlayer/tensorlayer>



TensorLayer and HPCC Systems

- Future work:
 - Text mining research:
 - Classification is just the beginning.
 - Background knowledge can help machine to understand natural language.
 - Summarisation of a scientific document and a collection of documents is a real challenge in the near future.
 - High performance platform:
 - HPCC Systems provide outstanding computation capability, which can be useful for parallel training of neural networks to find best parameters and hyper-parameters.
 - HPCC Systems provide efficient data delivery.
 - TensorLayer is a easy-to-use deep learning tool.
 - Our work should combine advantages from both.

Questions?



Jingqing Zhang

Data Science Institute
Imperial College London

<https://www.doc.ic.ac.uk/~jz9215/>

jingqing.zhang15@imperial.ac.uk

<http://www.imperial.ac.uk/>

**Imperial College
London**

THE DOWNLOAD

TECH TALKS BY HPCC SYSTEMS

ECL Summer Code Camp Review

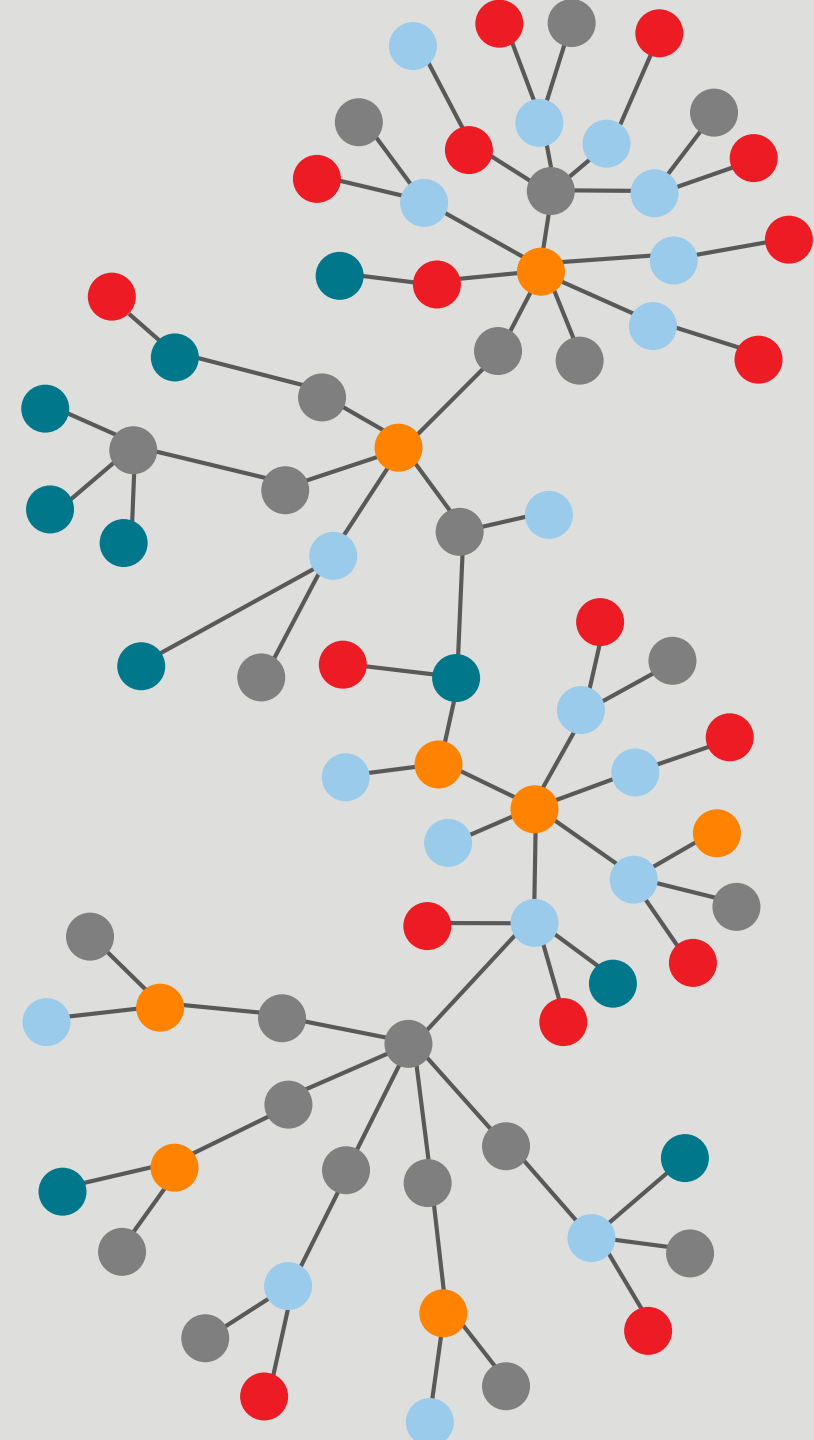


Bob Foreman
Senior Software Engineer
LexisNexis Risk Solutions



Quick poll:
Are you aware of a new ECL tutorial
located on GitHub?

See poll on bottom of presentation screen



History

- In January 2018, Arjuna Chala and Dan Camper conducted a three day workshop at New College of Florida in Sarasota. To use a fresh new look at real world Big Data, the NYC Yellow Taxi Fares dataset was introduced. The initial work with this dataset was archived in the HPCC Systems GitHub.
- In May 2018, members of the FAU iRise² class met with several ECL mentors at the LexisNexis Boca Raton campus for a one-day ECL introduction and workshop using the same Taxi data.
- Tomorrow, June 29th, 30 high school students from the Atlanta Metro area will meet in Alpharetta for the HPCC/ECL Code Lab once again using the NYC Taxi Dataset.
- Let's look at some of the cool ECL that was used for this project.



Initial Workflow – Phase 1

- **Spray** and **define** raw input data
- **Clean** the data – standardize dates and other numeric information
- **Validate** the data – what makes a valid record? Boolean rules.
- **Enrich** the data – time analysis is of great interest, so break down the dates and times into pieces that are easier to examine
- **Enhance** the data – link with weather conditions to see if it had any effect on the taxi trips and other metrics.
- **Analyze** the data – using filtering and cross-tab reports

Initial Workflow – Phase 1

Step 1: Move the data to the cluster, define raw RECORD and DATASET

The RECORD:

```
EXPORT Raw := MODULE

  EXPORT YellowLayout := RECORD
    STRING VendorID;
    STRING tpep_pickup_datetime;
    STRING tpep_dropoff_datetime;
    STRING passenger_count;
    STRING trip_distance;
    STRING pickup_longitude;
    STRING pickup_latitude;
    STRING rate_code_id;
    STRING store_and_fwd_flag;
    STRING dropoff_longitude;
    STRING dropoff_latitude;
    STRING payment_type;
    STRING fare_amount;
    STRING extra;
    STRING mta_tax;
    STRING tip_amount;
    STRING tolls_amount;
    STRING improvement_surcharge;
    STRING total_amount;
  END;
```

The DATASET:

```
EXPORT PATH := '~{'
  + PREFIX + '::raw::yellow_tripdata_2015-01.csv,'
  + PREFIX + '::raw::yellow_tripdata_2015-02.csv,'
  + PREFIX + '::raw::yellow_tripdata_2015-03.csv,'
  + PREFIX + '::raw::yellow_tripdata_2015-04.csv,'
  + PREFIX + '::raw::yellow_tripdata_2015-05.csv,'
  + PREFIX + '::raw::yellow_tripdata_2015-06.csv,'
  + PREFIX + '::raw::yellow_tripdata_2015-07.csv,'
  + PREFIX + '::raw::yellow_tripdata_2015-08.csv,'
  + PREFIX + '::raw::yellow_tripdata_2015-09.csv,'
  + PREFIX + '::raw::yellow_tripdata_2015-10.csv,'
  + PREFIX + '::raw::yellow_tripdata_2015-11.csv,'
  + PREFIX + '::raw::yellow_tripdata_2015-12.csv,'
  + PREFIX + '::raw::yellow_tripdata_2016-01.csv,'
  + PREFIX + '::raw::yellow_tripdata_2016-02.csv,'
  + PREFIX + '::raw::yellow_tripdata_2016-03.csv,'
  + PREFIX + '::raw::yellow_tripdata_2016-04.csv,'
  + PREFIX + '::raw::yellow_tripdata_2016-05.csv,'
  + PREFIX + '::raw::yellow_tripdata_2016-06.csv'
  + '}';

EXPORT inFile := DATASET(PATH, YellowLayout, CSV(HEADING(1)));
```


Initial Workflow – Phase 1

RAW DATA:

##	vendorid	tpcp_pickup_datetime	tpcp_dropoff_datetime	passenger_count	trip_distance	pickup_longitude	pickup_latitude	rate_code_id	store_and_fwd_flag	dropoff_longitude	dropoff_latitude
1	2	2015-01-15 19:05:39	2015-01-15 19:23:42	1	1.59	-73.993896484375	40.750110626220703	1	N	-73.974784851074219	40.75061
2	1	2015-01-10 20:33:38	2015-01-10 20:53:28	1	3.30	-74.00164794921875	40.7242431640625	1	N	-73.994415283203125	40.75910
3	1	2015-01-10 20:33:38	2015-01-10 20:43:41	1	1.80	-73.963340759277344	40.802787780761719	1	N	-73.951820373535156	40.82441
4	1	2015-01-10 20:33:39	2015-01-10 20:35:31	1	.50	-74.009086608886719	40.713817596435547	1	N	-74.004325866699219	40.71990
5	1	2015-01-10 20:33:39	2015-01-10 20:52:58	1	3.00	-73.971176147460938	40.762428283691406	1	N	-74.004180908203125	40.74265
6	1	2015-01-10 20:33:39	2015-01-10 20:53:52	1	9.00	-73.874374389648438	40.7740478515625	1	N	-73.986976623535156	40.75819
7	1	2015-01-10 20:33:39	2015-01-10 20:58:31	1	2.20	-73.9832763671875	40.726009368896484	1	N	-73.992469787597656	40.74963
8	1	2015-01-10 20:33:39	2015-01-10 20:42:20	3	.80	-74.002662658691406	40.734142303466797	1	N	-73.995010375976563	40.72632
9	1	2015-01-10 20:33:39	2015-01-10 21:11:35	3	18.20	-73.783042907714844	40.644355773925781	2	N	-73.987594604492187	40.75935
10	1	2015-01-10 20:33:40	2015-01-10 20:40:44	2	.90	-73.985588073730469	40.767948150634766	1	N	-73.985916137695313	40.75936
11	1	2015-01-10 20:33:40	2015-01-10 20:41:39	1	.90	-73.988616943359375	40.723102569580078	1	N	-74.00439453125	40.72858
12	1	2015-01-10 20:33:41	2015-01-10 20:43:26	1	1.10	-73.993782043457031	40.751419067382812	1	N	-73.9674072265625	40.75721
13	1	2015-01-10 20:33:41	2015-01-10 20:35:23	1	.30	-74.00836181640625	40.7043762220703125	1	N	-74.009773254394531	40.70772
14	1	2015-01-10 20:33:41	2015-01-10 21:03:04	1	3.10	-73.973945617675781	40.760448455810547	1	N	-73.997344970703125	40.73521
15	1	2015-01-10 20:33:41	2015-01-10 20:39:23	1	1.10	-74.006721496582031	40.731777191162109	1	N	-73.995216369628906	40.73989
16	2	2015-01-15 19:05:39	2015-01-15 19:32:00	1	2.38	-73.976425170898437	40.739810943603516	1	N	-73.983978271484375	40.75788
17	2	2015-01-15 19:05:40	2015-01-15 19:21:00	5	2.83	-73.968704223632812	40.754245758056641	1	N	-73.955123901367188	40.78689
18	2	2015-01-15 19:05:40	2015-01-15 19:28:18	5	8.33	-73.863059997558594	40.769580841064453	1	N	-73.952713012695312	40.78578
19	2	2015-01-15 19:05:41	2015-01-15 19:20:36	1	2.37	-73.945541381835938	40.779422760009766	1	N	-73.980850219726563	40.78608
20	2	2015-01-15 19:05:41	2015-01-15 19:20:22	2	7.13	-73.874458312988281	40.774009704589844	1	N	-73.952377319335938	40.71858

ECL Watch | Graphs | **RawTaxiData** | CleanedTaxidata | ValidatedTaxidata | EnrichedTaxidata | WeatherData

Initial Workflow – Phase 1

Step 2: Standardize (Clean) the data:

```
EXPORT CoercedYellowLayout := RECORD
  UNSIGNED1 VendorID;
  STRING19 tpep_pickup_datetime;
  STRING19 tpep_dropoff_datetime;
  UNSIGNED1 passenger_count;
  DECIMAL10_2 trip_distance;
  DECIMAL9_6 pickup_longitude;
  DECIMAL9_6 pickup_latitude;
  UNSIGNED1 rate_code_id;
  STRING1 store_and_fwd_flag;
  DECIMAL9_6 dropoff_longitude;
  DECIMAL9_6 dropoff_latitude;
  UNSIGNED1 payment_type;
  DECIMAL8_2 fare_amount;
  DECIMAL8_2 extra;
  DECIMAL8_2 mta_tax;
  DECIMAL8_2 tip_amount;
  DECIMAL8_2 tolls_amount;
  DECIMAL8_2 improvement_surcharge;
  DECIMAL8_2 total_amount;
END;
```

```
EXPORT YellowLayout := RECORD
  UNSIGNED4 record_id;
  CoercedYellowLayout;
END;
```

```
rawTaxiData := DATASET(Taxi.Files.Raw.PATH,
                        Taxi.Files.ETL.CoercedYellowLayout, CSV(HEADING(1)));

node := STD.System.Thorlib.node();
nodes := CLUSTER_SIZE;

etlTaxiData := PROJECT
(
  rawTaxiData,
  TRANSFORM
  (
    Taxi.Files.ETL.YellowLayout,
    SELF.record_id := ((COUNTER-1) * (nodes-1)) + node + COUNTER,
    SELF := LEFT
  )
, LOCAL);

OUTPUT(etlTaxiData, '~taxi_data::ETLData', COMPRESSED, OVERWRITE);
```

Initial Workflow – Phase 1

CLEAN DATA:

##	record_id	vendorid	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	pickup_longitude	pickup_latitude	rate_code_id	store_and_fwd_flag	dropoff_longitude	dropoff_latitude	payment_type
1	1	2	2015-01-15 19:05:39	2015-01-15 19:23:42	1	1.59	-73.993896	40.750111	1	N	-73.974785	40.750618	1
2	5	1	2015-01-10 20:33:38	2015-01-10 20:53:28	1	3.3	-74.001648	40.724243	1	N	-73.994415	40.759109	1
3	9	1	2015-01-10 20:33:38	2015-01-10 20:43:41	1	1.8	-73.963341	40.802788	1	N	-73.95182	40.824413	2
4	13	1	2015-01-10 20:33:39	2015-01-10 20:35:31	1	0.5	-74.009087	40.713818	1	N	-74.004326	40.719986	2
5	17	1	2015-01-10 20:33:39	2015-01-10 20:52:58	1	3	-73.971176	40.762428	1	N	-74.004181	40.742653	2
6	21	1	2015-01-10 20:33:39	2015-01-10 20:53:52	1	9	-73.874374	40.774048	1	N	-73.986977	40.758194	1
7	25	1	2015-01-10 20:33:39	2015-01-10 20:58:31	1	2.2	-73.983276	40.726009	1	N	-73.99247	40.749634	2
8	29	1	2015-01-10 20:33:39	2015-01-10 20:42:20	3	0.8	-74.002663	40.734142	1	N	-73.99501	40.726326	1
9	33	1	2015-01-10 20:33:39	2015-01-10 21:11:35	3	18.2	-73.783043	40.644356	2	N	-73.987595	40.759357	2
10	37	1	2015-01-10 20:33:40	2015-01-10 20:40:44	2	0.9	-73.985588	40.767948	1	N	-73.985916	40.759365	1
11	41	1	2015-01-10 20:33:40	2015-01-10 20:41:39	1	0.9	-73.988617	40.723103	1	N	-74.004395	40.728584	1
12	45	1	2015-01-10 20:33:41	2015-01-10 20:43:26	1	1.1	-73.993782	40.751419	1	N	-73.967407	40.757217	1
13	49	1	2015-01-10 20:33:41	2015-01-10 20:35:23	1	0.3	-74.008362	40.704376	1	N	-74.009773	40.707726	2
14	53	1	2015-01-10 20:33:41	2015-01-10 21:03:04	1	3.1	-73.973946	40.760448	1	N	-73.997345	40.73521	1
15	57	1	2015-01-10 20:33:41	2015-01-10 20:39:23	1	1.1	-74.006721	40.731777	1	N	-73.995216	40.739895	2
16	61	2	2015-01-15 19:05:39	2015-01-15 19:32:00	1	2.38	-73.976425	40.739811	1	N	-73.983978	40.757889	1
17	65	2	2015-01-15 19:05:40	2015-01-15 19:21:00	5	2.83	-73.968704	40.754246	1	N	-73.955124	40.786858	2
18	69	2	2015-01-15 19:05:40	2015-01-15 19:28:18	5	8.33	-73.86306	40.769581	1	N	-73.952713	40.785782	1
19	73	2	2015-01-15 19:05:41	2015-01-15 19:20:36	1	2.37	-73.945541	40.779423	1	N	-73.98085	40.786083	1
20	77	2	2015-01-15 19:05:41	2015-01-15 19:20:22	2	7.13	-73.874458	40.77401	1	N	-73.952377	40.71859	1

ECL Watch | Graphs | RawTaxiData | **CleanedTaxidata** | ValidatedTaxidata | EnrichedTaxidata | WeatherData

Initial Workflow – Phase 1

Step 3: Validate the data: What is a “good” record?

Validation Rules

bad_trip_distance

bad_passenger_count

bad_pickup_coordinates

bad_dropoff_coordinates

bad_fare_amount

bad_tip_amount

bad_tolls_amount

bad_improvement_surcharge

bad_total_amount

trip_distance ≤ 0

passenger_count < 1 OR passenger_count > 6

pickup_longitude = 0 OR pickup_latitude = 0,

dropoff_longitude = 0 OR dropoff_latitude = 0

fare_amount ≤ 0 OR fare_amount ≥ 1000

tip_amount < 0 ,

tolls_amount < 0 ,

improvement_surcharge < 0 ,

total_amount < 0 ,

Initial Workflow – Phase 1

Validate the data:

```
validatedData := PROJECT
(
  etlData,
  TRANSFORM
  (
    Taxi.Files.Validated.YellowLayout,
    SELF.bad_trip_distance      := LEFT.trip_distance <= 0,
    SELF.bad_passenger_count    := LEFT.passenger_count < 1 OR LEFT.passenger_count > 6,
    SELF.bad_pickup_coordinates := LEFT.pickup_longitude = 0 OR LEFT.pickup_latitude = 0,
    SELF.bad_dropoff_coordinates := LEFT.dropoff_longitude = 0 OR LEFT.dropoff_latitude = 0,
    SELF.bad_fare_amount        := LEFT.fare_amount <= 0 OR LEFT.fare_amount >= 1000,
    SELF.bad_tip_amount         := LEFT.tip_amount < 0,
    SELF.bad_tolls_amount       := LEFT.tolls_amount < 0,
    SELF.bad_improvement_surcharge := LEFT.improvement_surcharge < 0,
    SELF.bad_total_amount       := LEFT.total_amount < 0,
    SELF.is_valid_record        := NOT (SELF.bad_trip_distance OR
                                        SELF.bad_passenger_count OR
                                        SELF.bad_pickup_coordinates OR
                                        SELF.bad_dropoff_coordinates OR
                                        SELF.bad_fare_amount OR
                                        SELF.bad_tip_amount OR
                                        SELF.bad_tolls_amount OR
                                        SELF.BAD_improvement_surcharge OR SELF.bad_total_amount),
    SELF := LEFT
  )
):PERSIST('~taxi_data_validated_PERSIST');
OUTPUT(validatedData,, '~taxi_data::data_allvalidated', COMPRESSED, OVERWRITE);
```

Initial Workflow – Phase 1

VALIDATED DATA:

##	urcharge	total_amount	bad_trip_distance	bad_passenger_count	bad_pickup_coordinates	bad_dropoff_coordinates	bad_fare_amount	bad_tip_amount	bad_tolls_amount	bad_improvement_surcharge	bad_total_amount	is_valid_record
57		12.09	false	false	false	false	false	false	false	false	false	true
58		11.76	false	false	false	false	false	false	false	false	false	true
59		40.3	false	false	false	false	false	false	false	false	false	true
60		9.8	false	false	false	false	false	false	false	false	false	true
61		3.3	false	false	false	false	false	false	false	false	false	true
62		14.15	false	false	true	true	false	false	false	false	false	false
63		4.3	false	false	false	false	false	false	false	false	false	true
64		15.8	false	false	false	false	false	false	false	false	false	true
65		9.2	false	false	false	false	false	false	false	false	false	true
66		33.13	false	false	false	false	false	false	false	false	false	true
67		6.3	false	false	true	true	false	false	false	false	false	false
68		5.8	false	false	false	false	false	false	false	false	false	true
69		8.3	false	false	false	false	false	false	false	false	false	true
70		9.3	false	false	false	false	false	false	false	false	false	true
71		8.8	false	false	false	false	false	false	false	false	false	true
72		10.55	false	false	false	false	false	false	false	false	false	true
73		9.3	false	false	false	false	false	false	false	false	false	true
74		5.8	false	false	false	false	false	false	false	false	false	true
75		36.13	false	false	false	false	false	false	false	false	false	true
76		8.3	false	false	false	false	false	false	false	false	false	true

ECL Watch | Graphs | RawTaxiData | CleanedTaxidata | **ValidatedTaxidata** | EnrichedTaxidata | WeatherData

Initial Workflow – Phase 1

Step 4: Enrich the data:

```
EXPORT Enriched := MODULE
```

```
    EXPORT YellowLayout := RECORD
```

```
        Validated.YellowLayout;
```

```
        Std.Date.Date_t      pickup_date;
```

```
        Std.Date.Time_t      pickup_time;
```

```
        UNSIGNED2            pickup_minutes_after_midnight;
```

```
        UNSIGNED2            pickup_time_window;
```

```
        UNSIGNED1            pickup_time_hour;
```

```
        UNSIGNED1            pickup_day_of_week;
```

```
        Std.Date.Date_t      dropoff_date;
```

```
        Std.Date.Time_t      dropoff_time;
```

```
        UNSIGNED2            dropoff_minutes_after_midnight;
```

```
        UNSIGNED1            dropoff_time_window;
```

```
        UNSIGNED1            dropoff_time_hour;
```

```
        UNSIGNED1            dropoff_day_of_week;
```

```
        UNSIGNED2            trip_duration_minutes;
```

```
        UNSIGNED2            trip_distance_bucket;
```

```
    END;
```


Initial Workflow – Phase 1

```
withTimeEnrichment := PROJECT
(
  validatedData,
  TRANSFORM
  (
    Taxi.Files.Enriched.YellowLayout,
    SELF.pickup_date := Std.Date.FromStringToDate(LEFT.tpep_pickup_datetime[..10], '%Y-%m-%d'),
    SELF.pickup_time := Std.Date.FromStringToTime(LEFT.tpep_pickup_datetime[12..], '%H:%M:%S'),
    SELF.pickup_minutes_after_midnight := Std.Date.Hour(SELF.pickup_time) * 60 + Std.Date.Minute(SELF.pickup_time),
    SELF.pickup_time_window := SELF.pickup_minutes_after_midnight DIV Taxi.Constants.TIME_WINDOW_MINUTES + 1,
    SELF.pickup_time_hour := Std.Date.Hour(SELF.pickup_time),
    SELF.pickup_day_of_week := Std.Date.DayOfWeek(SELF.pickup_date),
    SELF.dropoff_date := Std.Date.FromStringToDate(LEFT.tpep_dropoff_datetime[..10], '%Y-%m-%d'),
    SELF.dropoff_time := Std.Date.FromStringToTime(LEFT.tpep_dropoff_datetime[12..], '%H:%M:%S'),
    SELF.dropoff_minutes_after_midnight := Std.Date.Hour(SELF.dropoff_time) * 60 + Std.Date.Minute(SELF.dropoff_time),
    SELF.dropoff_time_window := SELF.dropoff_minutes_after_midnight DIV Taxi.Constants.TIME_WINDOW_MINUTES + 1,
    SELF.dropoff_time_hour := Std.Date.Hour(SELF.dropoff_time),
    SELF.dropoff_day_of_week := Std.Date.DayOfWeek(SELF.dropoff_date),
    SELF.trip_duration_minutes := MAP
    (
      SELF.dropoff_date = SELF.pickup_date      => SELF.dropoff_minutes_after_midnight - SELF.pickup_minutes_after_midnight + 1,
      SELF.dropoff_date = SELF.pickup_date + 1  => SELF.dropoff_minutes_after_midnight + ((24 * 60) - SELF.pickup_minutes_after_midnight) + 1,
      SELF.dropoff_date > SELF.pickup_date + 1  => ((Std.Date.DaysBetween(SELF.pickup_date, SELF.dropoff_date) - 1) * (60 * 24)) +
        SELF.dropoff_minutes_after_midnight + ((24 * 60) -
        SELF.pickup_minutes_after_midnight) + 1,
      0
    ),
    SELF.trip_distance_bucket := LEFT.trip_distance DIV Taxi.Constants.TRIP_DISTANCE_BUCKET_SIZE + 1,
    SELF := LEFT
  )
)
```

Initial Workflow – Phase 1

ENRICHED DATA:

#	is_valid_record	pickup_date	pickup_time	pickup_minutes_after_midnight	pickup_time_window	pickup_time_hour	pickup_day_of_week	dropoff_date	dropoff_time	dropoff_minutes_after_midnight	dropoff_time_window	dropoff_time
1	true	20150115	190539	1145	230	19	5	20150115	192342	1163	233	19
2	true	20150110	203338	1233	247	20	7	20150110	205328	1253	251	20
3	true	20150110	203338	1233	247	20	7	20150110	204341	1243	249	20
4	true	20150110	203339	1233	247	20	7	20150110	203531	1235	248	20
5	true	20150110	203339	1233	247	20	7	20150110	205258	1252	251	20
6	true	20150110	203339	1233	247	20	7	20150110	205352	1253	251	20
7	true	20150110	203339	1233	247	20	7	20150110	205831	1258	252	20
8	true	20150110	203339	1233	247	20	7	20150110	204220	1242	249	20
9	true	20150110	203339	1233	247	20	7	20150110	211135	1271	255	21
10	true	20150110	203340	1233	247	20	7	20150110	204044	1240	249	20
11	true	20150110	203340	1233	247	20	7	20150110	204139	1241	249	20
12	true	20150110	203341	1233	247	20	7	20150110	204326	1243	249	20
13	true	20150110	203341	1233	247	20	7	20150110	203523	1235	248	20
14	true	20150110	203341	1233	247	20	7	20150110	210304	1263	253	21
15	true	20150110	203341	1233	247	20	7	20150110	203923	1239	248	20
16	true	20150115	190539	1145	230	19	5	20150115	193200	1172	235	19
17	true	20150115	190540	1145	230	19	5	20150115	192100	1161	233	19
18	true	20150115	190540	1145	230	19	5	20150115	192818	1168	234	19
19	true	20150115	190541	1145	230	19	5	20150115	192036	1160	233	19
20	true	20150115	190541	1145	230	19	5	20150115	192022	1160	233	19

Initial Workflow – Phase 1

Finally, Step 5: Analyze the data:

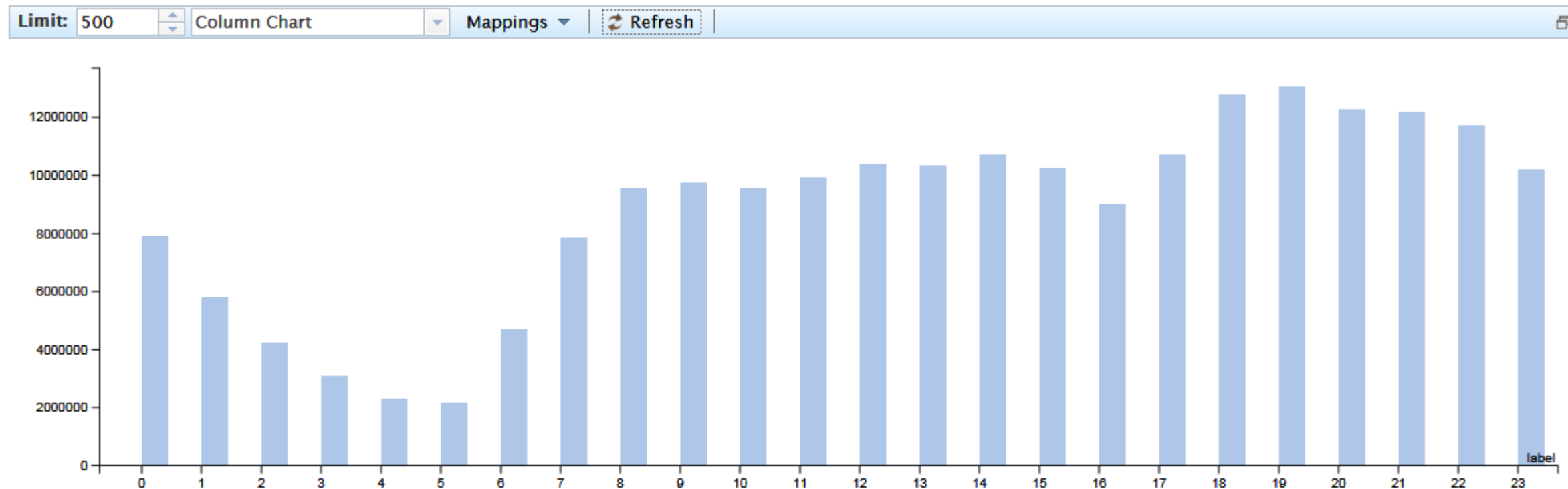
```
theData := CodeCampLab.File_TaxiEnrich.File(is_valid_record);

// Base aggregation
baseAggregation := TABLE
(
    theData,
    {
        pickup_day_of_week,
        pickup_time_hour,
        UNSIGNED4 cnt := COUNT(GROUP),
        DECIMAL10_2 total_trip_distance := SUM(GROUP, trip_distance)
    },
    pickup_day_of_week, pickup_time_hour
);

OUTPUT(SORT(baseAggregation, pickup_day_of_week, pickup_time_hour), NAMED('baseAggregation'), ALL);
```

Initial Workflow – Phase 1

Cumulative Pickups by Hour:



Initial Workflow – Phase 1

Pickups per hour by day of week

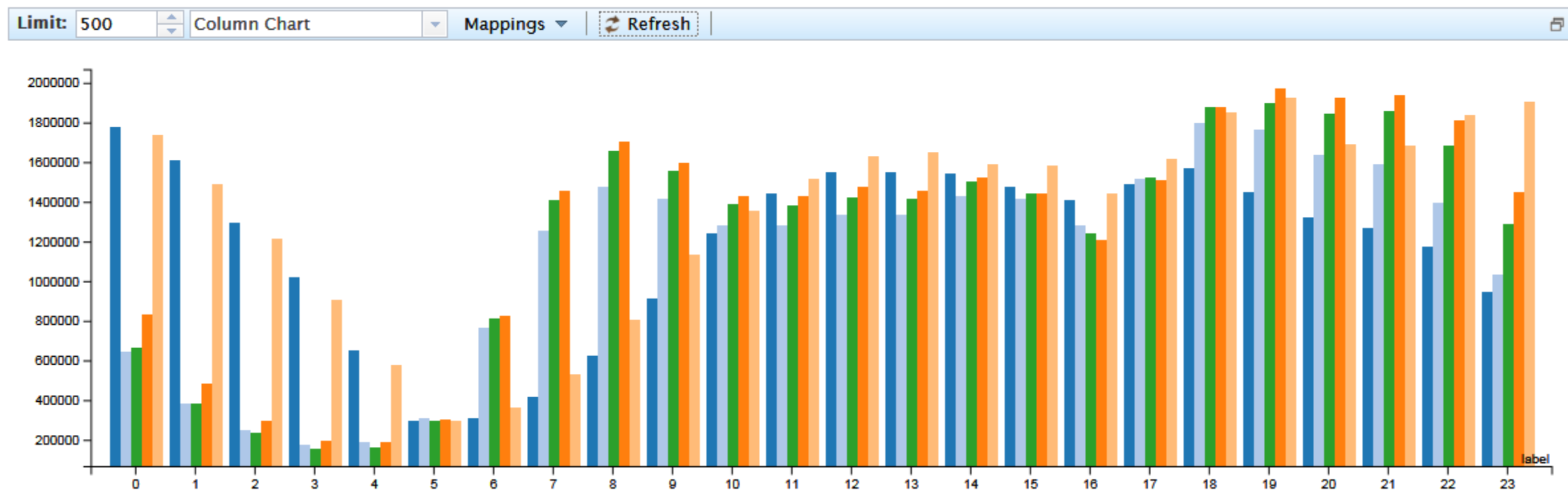
```
//=====

// Pickups per hour by day of week
perHourPerDayCount := TABLE
(
    baseAggregation,
    {
        pickup_time_hour,
        UNSIGNED4 cnt_total := SUM(GROUP, cnt),
        UNSIGNED4 cnt_sunday := SUM(GROUP, IF(pickup_day_of_week = 1, cnt, 0)),
        UNSIGNED4 cnt_monday := SUM(GROUP, IF(pickup_day_of_week = 2, cnt, 0)),
        UNSIGNED4 cnt_tuesday := SUM(GROUP, IF(pickup_day_of_week = 3, cnt, 0)),
        UNSIGNED4 cnt_wednesday := SUM(GROUP, IF(pickup_day_of_week = 4, cnt, 0)),
        UNSIGNED4 cnt_thursday := SUM(GROUP, IF(pickup_day_of_week = 5, cnt, 0)),
        UNSIGNED4 cnt_friday := SUM(GROUP, IF(pickup_day_of_week = 6, cnt, 0)),
        UNSIGNED4 cnt_saturday := SUM(GROUP, IF(pickup_day_of_week = 7, cnt, 0)),
    },
    pickup_time_hour
);

OUTPUT(SORT(perHourPerDayCount, pickup_time_hour), NAMED('perHourPerDayCount'), ALL);
```

Initial Workflow – Phase 1

Pickups by Hour by Day of Week:



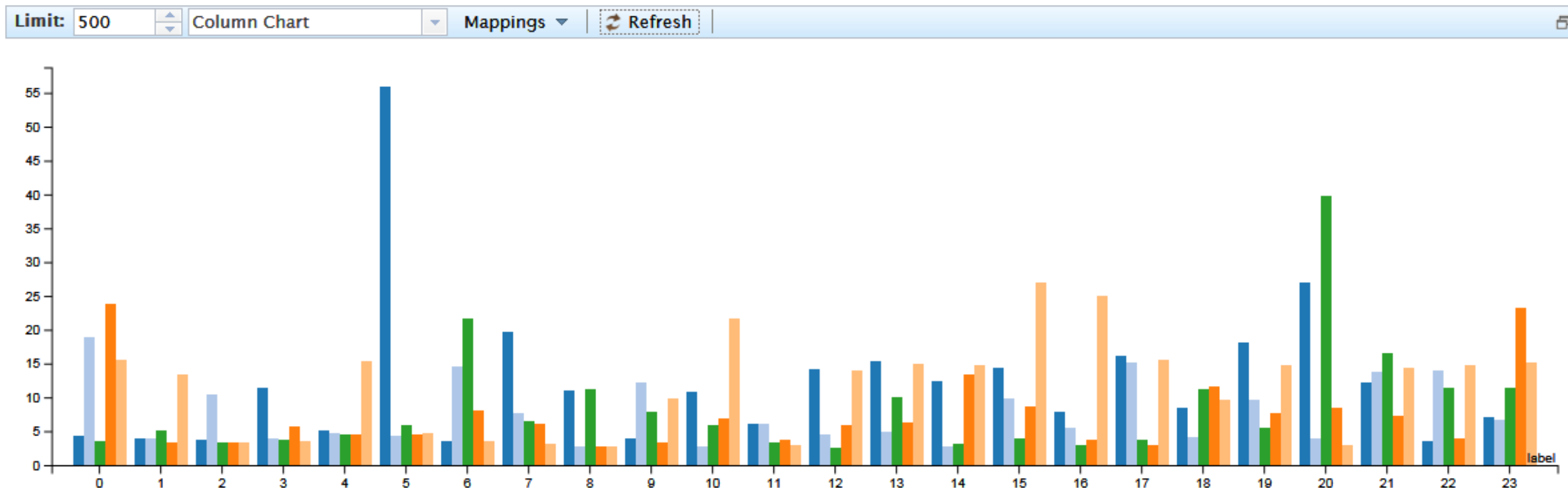
Initial Workflow – Phase 1

Average trip distance per hour by day of week:

```
// Average trip distance per hour by day of week
perHourPerDayAveDistance := TABLE
(
    baseAggregation,
    {
        pickup_time_hour,
        DECIMAL10_2 dist_total := SUM(GROUP, total_trip_distance) / SUM(GROUP, cnt),
        DECIMAL10_2 dist_sunday := SUM(GROUP, IF(pickup_day_of_week = 1, total_trip_distance / cnt, 0)),
        DECIMAL10_2 dist_monday := SUM(GROUP, IF(pickup_day_of_week = 2, total_trip_distance / cnt, 0)),
        DECIMAL10_2 dist_tuesday := SUM(GROUP, IF(pickup_day_of_week = 3, total_trip_distance / cnt, 0)),
        DECIMAL10_2 dist_wednesday := SUM(GROUP, IF(pickup_day_of_week = 4, total_trip_distance / cnt, 0)),
        DECIMAL10_2 dist_thursday := SUM(GROUP, IF(pickup_day_of_week = 5, total_trip_distance / cnt, 0)),
        DECIMAL10_2 dist_friday := SUM(GROUP, IF(pickup_day_of_week = 6, total_trip_distance / cnt, 0)),
        DECIMAL10_2 cnt_saturday := SUM(GROUP, IF(pickup_day_of_week = 7, total_trip_distance / cnt, 0)),
    },
    pickup_time_hour
);
OUTPUT(SORT(perHourPerDayAveDistance, pickup_time_hour), NAMED('perHourPerDayAveDistance'), ALL);
```

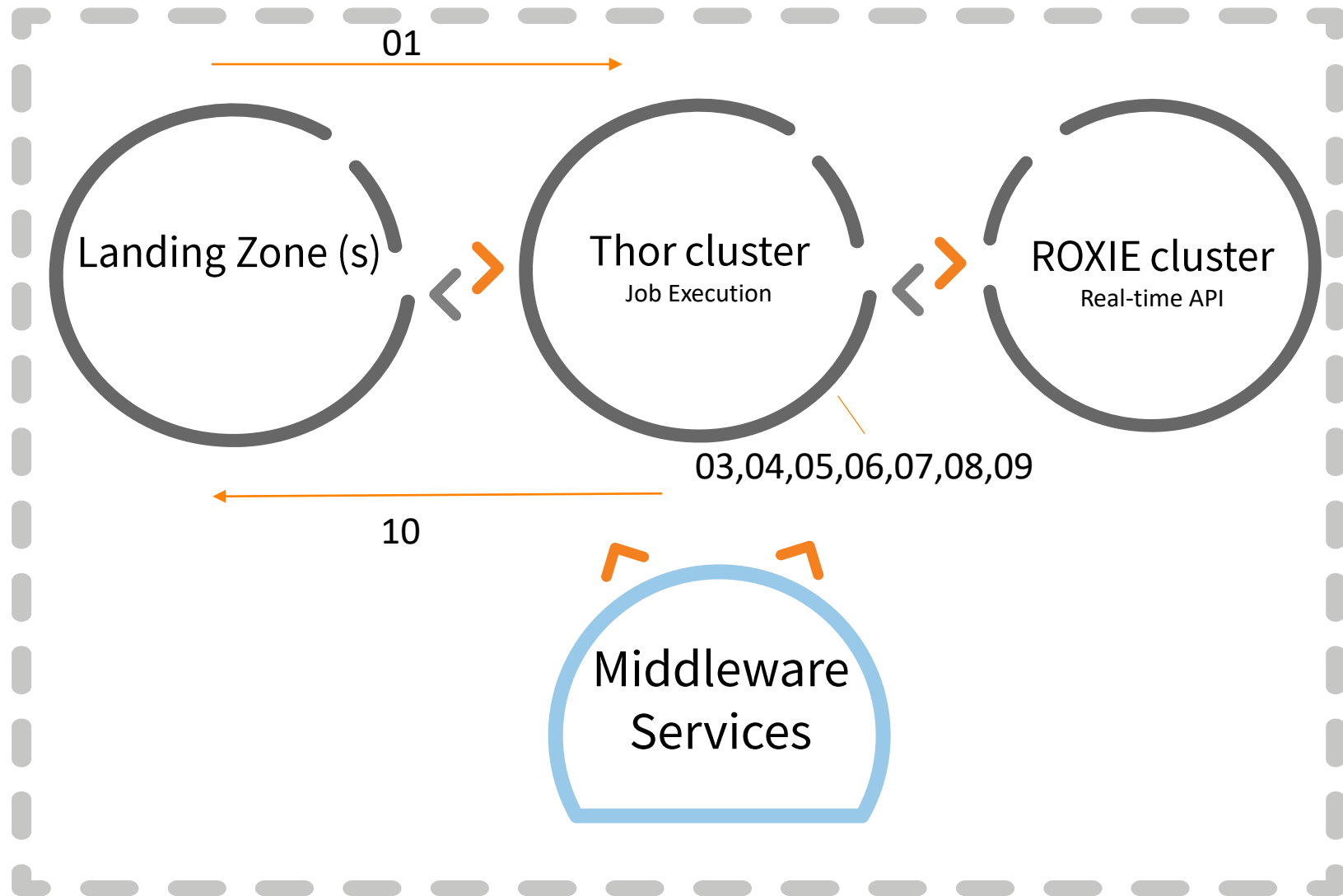
Initial Workflow – Phase 1

Average trip distance per hour by day of week:



Latest workflow (Phase 2)

- 01 – Data Import
- 02 – Data Import Validate
- 03 – Data Profile
- 04 – Clean
- 05 – Enrich
- 06 – Analysis
- 07 - Visualize
- 08 – Train
- 09 – Model Build
- 10 – Export Data



Acknowledgements

Arjuna Chala

Dan Camper – admitted thaumaturge ☺

Dinesh Shetye

Get the code!

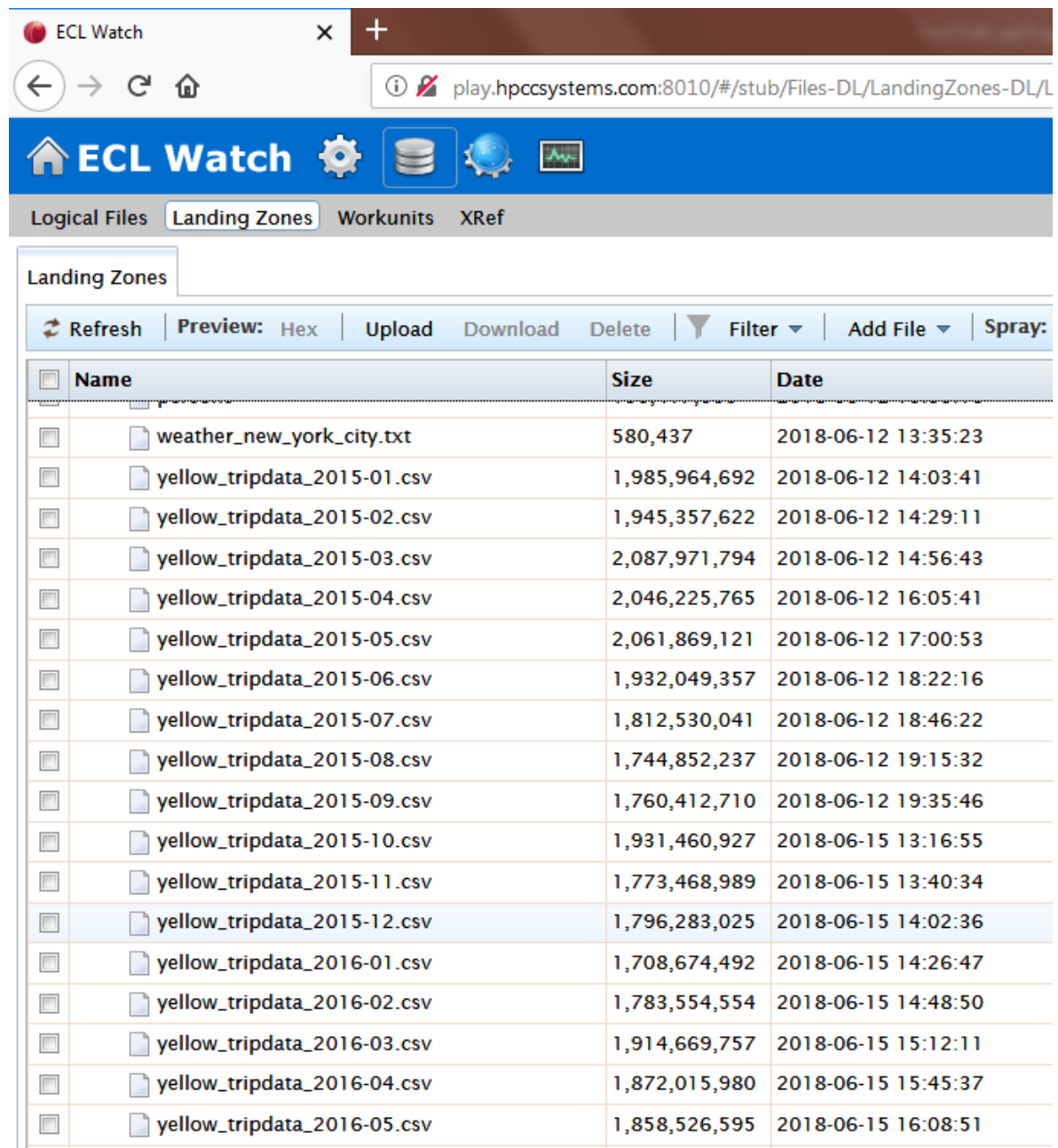
<https://github.com/hpcc-systems/Solutions-ECL-Training>

Play with the data!

<http://play.hpccsystems.com:8010>

Play.hpccsystems.com

– Landing Zone



ECL Watch

play.hpccsystems.com:8010/#/stub/Files-DL/LandingZones-DL/L

ECL Watch

Logical Files Landing Zones Workunits XRef

Landing Zones

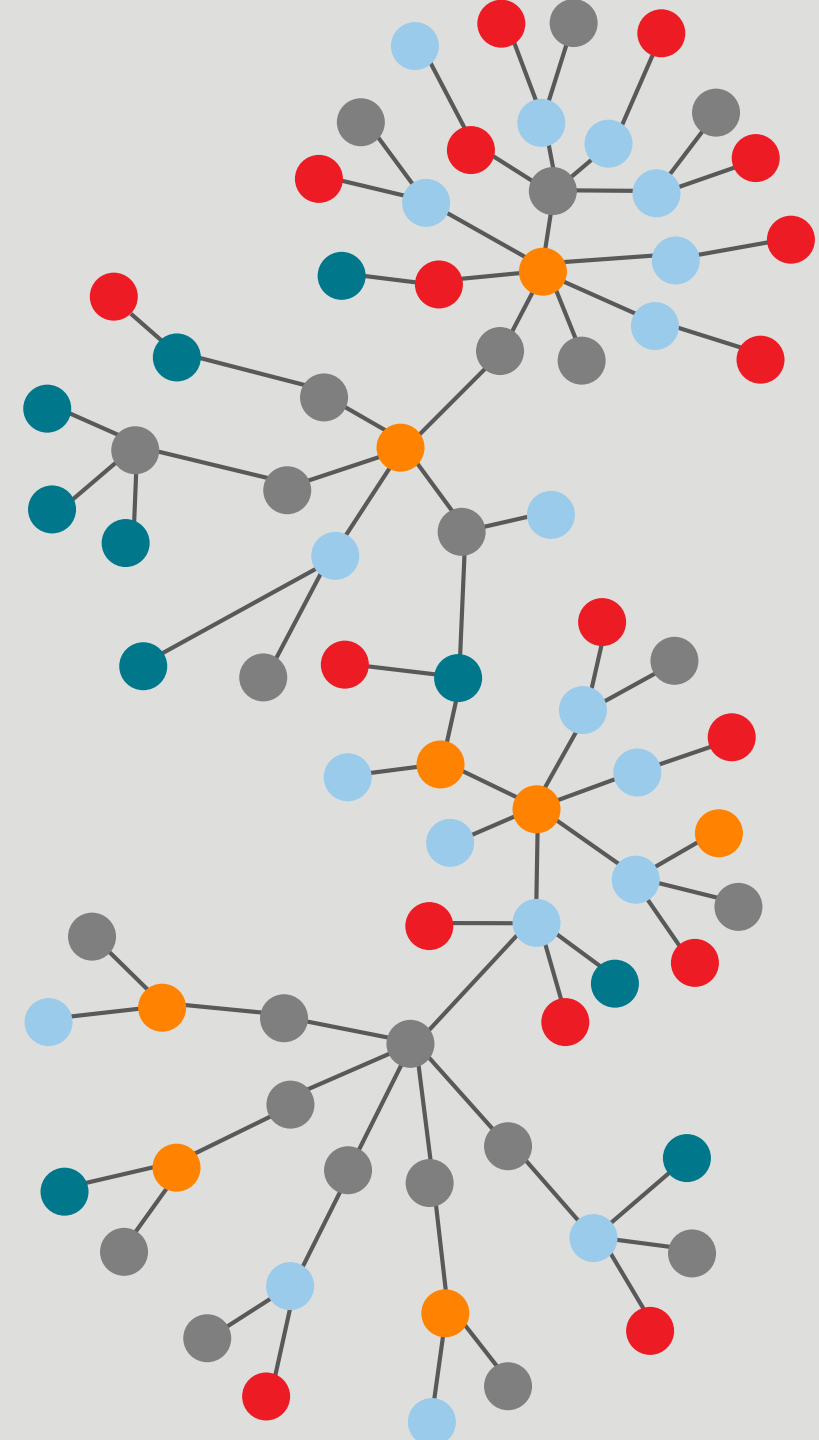
Refresh Preview: Hex Upload Download Delete Filter Add File Spray:

Name	Size	Date
weather_new_york_city.txt	580,437	2018-06-12 13:35:23
yellow_tripdata_2015-01.csv	1,985,964,692	2018-06-12 14:03:41
yellow_tripdata_2015-02.csv	1,945,357,622	2018-06-12 14:29:11
yellow_tripdata_2015-03.csv	2,087,971,794	2018-06-12 14:56:43
yellow_tripdata_2015-04.csv	2,046,225,765	2018-06-12 16:05:41
yellow_tripdata_2015-05.csv	2,061,869,121	2018-06-12 17:00:53
yellow_tripdata_2015-06.csv	1,932,049,357	2018-06-12 18:22:16
yellow_tripdata_2015-07.csv	1,812,530,041	2018-06-12 18:46:22
yellow_tripdata_2015-08.csv	1,744,852,237	2018-06-12 19:15:32
yellow_tripdata_2015-09.csv	1,760,412,710	2018-06-12 19:35:46
yellow_tripdata_2015-10.csv	1,931,460,927	2018-06-15 13:16:55
yellow_tripdata_2015-11.csv	1,773,468,989	2018-06-15 13:40:34
yellow_tripdata_2015-12.csv	1,796,283,025	2018-06-15 14:02:36
yellow_tripdata_2016-01.csv	1,708,674,492	2018-06-15 14:26:47
yellow_tripdata_2016-02.csv	1,783,554,554	2018-06-15 14:48:50
yellow_tripdata_2016-03.csv	1,914,669,757	2018-06-15 15:12:11
yellow_tripdata_2016-04.csv	1,872,015,980	2018-06-15 15:45:37
yellow_tripdata_2016-05.csv	1,858,526,595	2018-06-15 16:08:51

Quick poll:

Will you come by and visit our new
public cluster (play.hpccsystems.com)
after today?

See poll on bottom of presentation screen



Questions?



Bob Foreman

Senior Software Engineer

LexisNexis Risk Solutions

Robert.Foreman@lexisnexisrisk.com

Submit a talk for an upcoming episode!

- Have a new success story to share?
- Want to pitch a new use case?
- Have a new HPCC Systems application you want to demo?
- Want to share some helpful ECL tips and sample code?
- Have a new suggestion for the roadmap?
- Be a featured speaker for an upcoming episode! Email your idea to Techtalks@hpccsystems.com
- Visit The Download Tech Talks wiki for more information: <https://wiki.hpccsystems.com/display/hpcc/HPCC+Systems+Tech+Talks>

Stay tuned for our next Tech Talk!
Watch our [Events](#) page for details.

Thank You!



 **RELX** Group

A copy of this presentation will be made available soon on our blog:
hpccsystems.com/blog