DSE 260B Final Report
California Highway Traffic Analysis

Kevin Dyer
John Gill III
Conway Wong

Advisor: Yoav Freund

June 10, 2016

Table of Contents

0. Abstract

Traffic modeling and prediction is a field that has been researched and studied for many years. With the introduction of large data sets taken from sensor stations throughout California, and a myriad of other data sources available contributing to traffic metrics collection, opportunities for analysis into traffic modeling and the factors causing traffic are ever expanding.

In this Capstone overview, we describe how traffic volume can be modeled utilizing Principal Component Analysis (PCA), additionally how easily and effectively it can be visualized to assist in the detection of traffic patterns and volume for further analysis. In addition, our work experiments with identifying different traffic behaviors exhibited throughout the week utilizing the KMeans++ clustering algorithm. A cursory examination into determining the factors that contribute to traffic volume is explored using Elastic Net Regression. The purpose of this work is to inform business decision makers and the general public of traffic patterns that exist in California, with the hope being that additional insight can guide solutions to address high traffic volume.

1. Introduction and Question Formulation

Traffic in California cities is ranked among America's worst. California certainly tops the list of American cities with the worst traffic congestion. According to the TomTom Traffic Index, 4 out of the top 20 congested traffic cities in America are found in the Golden State: #1 Los Angeles, #2 San Francisco, #5 San Jose, and #14 San Diego. Our project is an exploration into this phenomenon by studying California's highways in the hope that a greater understanding of traffic patterns can provide insight into addressing the problem. This work attempts to answer the following questions:

- Can the flow of traffic on a highway be modeled?
- Can the variation in traffic patterns be expressed by a small number of variables?
- What is the overall pattern along a particular busy highway such as Hwy 5?
- What factors have the most influence on traffic?

2. Team Roles and Responsibilities

Group Member	Roles
Conway Wong	Data Acquisition/Preparation, Programmer, Viz (GIS, Javascript)
John Gill III	Data Acquisition/Preparation, Programmer, Machine Learning
Kevin Dyer	Programmer, Machine Learning, Viz (Matplotlib, Bokeh)

3. Data Acquisition

The following data sources were identified and utilized for the Capstone project:

- Caltrans Performance Measurement System (PeMS)
- California Highway Patrol
- U.S. Census
- Zillow

All data sources are freely available for public use.

Data Sources

Caltrans PeMS

The primary data source chosen originates from the California Department of Transportation (CalTrans) Performance Measurement System (PeMS). The PeMS system is an Archived Data User Service (ADUS) site providing more than 10 years of historical traffic data collected from more than 39,000 sensors across California highways and major roadways. The site archives CalTrans loop detector data from traffic management centers (TMCs) throughout the state. The TMC's main purpose is to provide a portal which exposes analytical capabilities for various use cases such as supporting freeway operations, travel, and research. Additionally, it hosts a publically available data warehouse called the Data Clearinghouse. The Data Clearinghouse provides freely available historical data collected from sensors all across the state. The Data Clearinghouse was the source of the majority and most important data sets used for this Capstone project.

Station 5min Readings

As cars pass over loop detectors buried in California roadways, summary data is collected every 30 seconds and sent to the PeMS system for analysis and archiving. These 30 second readings are then aggregated into 5-minute data files and made available on the PeMS Clearinghouse site. These 5-minute files are the source of the traffic data that was the core of our datasets. Data was from the timeframe of January 1st, 2008 to the end of 2015 was collected for every online station in California. Below is the summary of the fields of the station data.

Name	Comment
Timestamp	The date and time of the beginning of the summary interval. For example, a time of 08:00:00 indicates that the aggregate(s) contain measurements collected between 08:00:00 and 08:04:59. Note that second values are always 0 for five-minute aggregations. The format is MM/DD/YYYY HH24:MI:SS.
Station	Unique station identifier. Use this value to cross-reference with Metadata files.
District	District #
Freeway #	Freeway #
Direction of Travel	N | S | E | W
Lane Type	A string indicating the type of lane. Possible values (and their meaning are: CD (Coll/Dist) CH (Conventional Highway) FF (Fwy-Fwy connector) FR (Off Ramp) HV (HOV) ML (Mainline) OR (On Ramp)

Station Length	Segment length covered by the station in miles/km.
Samples	Total number of samples received for all lanes.
% Observed	Percentage of individual lane points at this location that were observed (e.g. not imputed).
Total Flow	Sum of flows over the 5-minute period across all lanes. Note that the basic 5-minute rollup normalizes flow by the number of good samples received from the controller.
Avg Occupancy	Average occupancy across all lanes over the 5-minute period expressed as a decimal number between 0 and 1.
Avg Speed	Flow-weighted average speed over the 5-minute period across all lanes. If flow is 0, mathematical average of 5-minute station speeds.
Lane N Samples	Number of good samples received for lane N. N ranges from 1 to the number of lanes at the location.
Lane N Flow	Total flow for lane N over the 5-minute period normalized by the number of good samples.
Lane N Avg Occ	Average occupancy for lane N expressed as a decimal number between 0 and 1. N ranges from 1 to the number of lanes at the location.
Lane N Avg Speed	Flow-weighted average of lane N speeds. If flow is 0, mathematical average of 5-minute lane speeds. N ranges from 1 to the number of lanes
Lane N Observed	1 indicates observed data, 0 indicates imputed.

Station Metadata

Each 5-minute traffic reading is identified by a station ID number. The information about each station is found in another dataset, the station metadata files. Station metadata contains descriptive information for the recording traffic stations along California highways. The station metadata is reported as a tab-delimited format with the fields defined in the table below.

Name	Comment
ID	An integer value that uniquely identifies the Station Metadata. Use this value to 'join' other clearinghouse files that contain Station Metadata.
Freeway	Freeway Number
Freeway Direction	A string indicating the freeway direction.
County Identifier	The unique number that identifies the county that contains this census station within PeMS.
City	City
State Postmile	State Postmile
Absolute	Absolute Postmile

Postmile	
Latitude	Latitude
Longitude	Longitude
Length	Length
Type	Type
Lanes	Total number of lanes
Name	Name
User IDs[1-4]	User-entered string identifier

CHP Incidents Day

This dataset contains California Highway Patrol (CHP) Incidents from all Caltrans Districts. Each downloadable file contains all incidents that occurred in one day across the state.

Name	Comment
Incident ID	An integer value that uniquely identifies this incident within PeMS.
CC Code	CC Code
Incident Number	An integer incident number
Timestamp	Date and time of the incident with a format of MM/DD/YYYY HH24:MI:SS. For example 9/3/2013 13:58, indicating 9/3/2013 1:58 PM.
Description	A textual description of the incident.
Location	A textual description of the location.
Area	A textual description of the Area. For example, East Sac.
Zoom Map	Zoom Map
TB xy	Lat/lon in state plane. Available from 4/9/2009
Latitude	Latitude
Longitude	Longitude
District	The District number
County FIPS ID	The FIPS County identifier.
City FIPS ID	The FIPS City identifier.

Freeway Number	Freeway Number
Freeway Direction	A string indicating the freeway direction.
State Postmile	State Postmile
Absolute Postmile	Absolute Postmile
Severity	Severity
Duration	Duration
incident_id	Incident ID
detail_id	Detail ID
Timestamp	Date and time of the incident with a format of MM/DD/YYYY HH24:MI:SS. For example 9/3/2013 13:58, indicating 9/3/2013 1:58 PM.

Urban Areas

In order to perform regression on our data sets and determine feature importances, we wanted to include additional features - more than just raw total flow data and CHP incidents. One such feature was whether an area is urban or rural. The information for this feature is found from the CalTrans web site at http://www.dot.ca.gov/hq/tsip/gis/datalibrary/Metadata/UrbanArea_Adjusted.html.

Housing Prices

Another potential key feature characterizing traffic patterns we included was housing prices. Although there are numerous sources of home prices, we decided to use the median home values provided by the Zillow Home Value Index (ZHVI) found at http://www.zillow.com/research/data/.

Zip Codes

In order to join the station data information with other features such as urban/rural and housing prices, we needed a common feature among the data sets. We decided to use zip code as the common feature, but the traffic station metadata only reports latitude and longitude, not zip code. Using TIGER/Line shapefiles available from the U.S. Census, we are able to find a zipcode given a specified latitude/longitude. The Census shapefiles used are described at https://www.census.gov/geo/maps-data/data/tiger-line.html.

Data Collection

Caltrans PeMS

Although the PeMS website is very user friendly, the sheer number and volume of the PeMS traffic data set is impossible to download manually. Knowing that manual download of the 5-minute traffic data would be insurmountable, we spent some time looking analyzing the PeMS Clearinghouse web portal source code and discovered we could automate the discovery and download of all PeMS datasets. Collection of the data from PeMS was automated using a scraper written in Python using the BeautifulSoup and Mechanize frameworks. It was run on John's home server over the course of several days. After the PeMS files were downloaded onto John's server, the files were subsequently uploaded to S3 at s3://dse-team2-2014/dse_traffic. The source code for the scraper is found at https://github.com/conwaywong/dse_capstone/blob/master/traffic/src/scraper.py.

Although the volume of the station metadata and CHP incidents was fairly small, the number of files also made it unreasonable to manually download either data set as well. The collection of the station metadata and CHP incidents was performed via the same Python script by specifying additional type arguements.

Urban Areas

The CalTrans Adjusted Urban Areas data set is contained within a single file, so it was a manual download of the file located at http://www.dot.ca.gov/hq/tsip/gis/datalibrary/zip/Boundaries/2010_adjusted_urban_area.zip.

Housing Prices

The ZHVI data set is a single file, and the file was manually downloaded from http://files.zillowstatic.com/research/public/Zip/Zip_Zhvi_Summary_AllHomes.csv.

Zip Codes

The TIGER/Line California shapefile bundle is a single file and was manually downloaded from https://www2.census.gov/geo/tiger/TIGER2010/ZCTA5/2010/tl_2010_06_zcta510.zip.

Data Sizes

Below is a summary of the data sets and their sizes:

Data Set	Size
Station 5-minute Readings	1.40 TB (260 GB compressed)
Station Metadata	0.12GB
CHP Incidents Day	0.76 GB
Urban/Rural	2.1 MB (Shapefile)
Zillow Home Values	20.3 MB (CSV)
Census TIGER/Line	29 MB (Shapefile)

4. Data Preparation

Data Cleaning

CalTrans PeMS

The station 5-minute dataset was extremely clean and did not require any additional processing before we began to use it. Although we did not clean the data, we did observe missing data from a small percentage of 5-minute station observations. In particular, we observed a small number of values such as speed or flow missing from the data. Instead of cleaning the values using a method such as imputing the value with the mean or mode of that field, we decided to remove the entire observation when we encountered the missing data.

The station metadata was also clean and already in a fixed-width format and required no cleaning.

Urban Geography, Home Prices, Census

Because the data sets for the urban/rural areas, Zillow home prices, and Census shapefiles are curated datasets, no cleaning was necessary before data transformation and loading.

Data Transformation and Load

CalTrans PeMS

Before starting the modeling portion of the project, the 5-minute station data needed to be converted from a dataset where each row represented a freeway station's 5-minute interval readings into a dataset where each row represented a freeway station's daily readings. We grouped a station's 5-minute readings by day, sorted selected features (flow, occupancy, speed), and persisted the resulting row. Each station reported 288 5-minute readings each day (5 minutes X 24 hours = 288). We then used the resulting data rows to create a daily station output file with 869 features as shown below:

Pivoted Row

Station Id	District Id	Year	Day of Year	Day of Week	Total Flow (288)	Avg. Occupancy (288)	Avg. Speed (288)

The resulting pivoted dataset is a bzip2 CSV file with one file per year. Each file contains the total flow, average occupancy, and average speed features, each represented by 288 features (one per 5-minute reading). The key for each row is a combination of Station ID, Year, and Day of Year, but we added District Id and Day of Week for filtering purposes knowing that they would be useful for data partitioning during the modelling phase; calculating them during the pivoting phase saved us having to re-calculating the values during the modeling phase. We pivoted the entire state of California for each year between 2008 to 2015. The following table shows the sizes of the pivoted data files.

Year	Size (Compressed, bzip2)
2008	1.56 GB
2009	1.71 GB
2010	2.0 GB
2011	2.0 GB
2012	2.0 GB
2013	2.0 GB
2014	2.0 GB
2015	2.0 GB

Due to the sheer size of 5-minute traffic data required to to pivot, processing and pivoting the dataset was too much for our local machines to handle. Therefore, we uploaded the compressed, 5-minute dataset into Amazon S3 and wrote Scala algorithms to perform the pivoting. This pivoting was executed using Scala code in Databricks. The source code for the pivoting is found at https://github.com/conwaywong/dse_capstone/blob/master/ml/scala/traffic/src/main/scala/org/ucsd/dse/capstone/traffic/PivotExecutor.scala. Each resulting pivot dataset is found at s3://dse-team2-2014/pivot_output_<year>, where <year> is between 2008 and 2015. For reference purposes, the resulting pivot file for 2010 contains 2,441,966 rows representing 6997 unique traffic stations.

Station Meta, Urban Areas, and Zip Code

Because the total size of the PeMS station metadata, CalTrans Adjusted Urban Areas shapefile, and Census TIGER/Line shapefile are relatively small, the transformation and combination of the three sources into a single master station metadata file was performed on our local development machines using Python. The result of the transformation was a single JSON file where each PeMS mainline traffic station is represented by the following attributes:

- Station: PeMS station identifier
- latitude: station latitude in decimal degrees
- longitude: station longitude in decimal degrees
- freeway: highway number associated with the station
- district: CalTrans district identifier
- name: label assigned to the station
- zip: zipcode of the station given by the Census shapefile
- direction: 1 = North, 2 = South, 3 = East, 4 = West
- Urban: 0 = rural. 1 = urban as defined by the CalTrans Adjusted Urban Areas shapefile

Transforming and joining the three data sets into a single JSON file was significantly beneficial for our data analysis and visualization portions of the Capstone. The result of merging the three data source into an index table by traffic station and including all features (zipcode, geolocation, freeway, direction) into a single JSON file made the visualization and filtering of the station data became much more manageable. Additionally, the zipcode attribute is a feature that allowed us to join the station data with other features such as housing prices. The source code for creating the station JSON file is found at https://github.com/conwaywong/dse_capstone/blob/master/final/Traffic_Station_Meta_to_JSON.ipynb. The JSON file can be found at s3://dse-team2-2014/station_meta_ML2.json. Overall, the data transformation of the three sources into a single file was straight-forward, but was a little tricky to correctly parse the two shapefiles involved because not all stations report a latitude and longitude.

Housing Prices

Determining the best way for the Zillow Home Value Index (ZHVI) values to be integrated with the rest of our traffic data was a bit of a challenge. The initial challenge was determining how to join data where each row represented a day of the year (station readings) against the home price data that was reported monthly. Ultimately what we, deiced was to use the value for the home price for all readings that fell within that month.
Each row in the ZHVI was reported for a given City, State, Metro, and County with a column for each month ranging from April of 1996 to February of 2016. For the join to work a temporal index would need to be created based upon the home value date. This meant that each row would need to be broken out into the key fields of location, the date (month) of reporting and then ultimately the Value Index. The code for this script can be found at the following: https://github.com/conwaywong/dse_capstone/blob/master/final/density_pivot.py.

It should be noted that this script script was used to store the density information aggregated above into the RDBMS.

PostgreSQL

After all the data had been acquired and some of the preprocessing had been completed, we decided that storing it into an RDBMS for ease of joining and further analysis would be the best idea. This would in theory allow us to pick apart the various pieces of information and use them in any regression analysis for feature importance or predicting traffic flow.

One of the first decisions was to store the projected vector values of the station data to reduce the amount of data that would be stored and have to be processed. Instead of having to store 288 values for each station observation

only the top 10 coefficients would be required. Even with this reduced set of data it still required the use of Spark to be able to efficiently process the data and load it into a postgres database using the jdbc driver. Additional tricks in terms of setting the number and size of data partitions to chunk the data into postgres via the `COPY` command instead of individual `INSERT` statements to be able to load the roughly 18 million data rows for just observations.

To do some preliminary filtering joins were done in spark against the Traffic Station metadata to ensure that rows for only stations existing in the database were being stored. This also allowed for the station metadata and CHP incidents to be processed via the same partitioning and bulk load operations. Also by using the parallelism of Spark it was easy to do the transform required to split off data from the traffic station row into the individual fact tables.

The only information that was not processed via Spark was the housing value index data, the size of the data was small enough that serializing it via a single python script was adequate to load it into the postgres database. Similarly the update statements to populate the urban and density information were done via a python script (see the density_pivot.py uder Housing Prices).

Queries

Once all the resulting data had been stored into the relational model designing the queries for what features about the data was a challenge unto itself. We had data that was both temporally based on various intervals (home value) as well as data that was geolocated near traffic stations (CHP incidents). The ability to join all this information together into a cohesive row that could be used for performing regression against was definitely challenging.

One of the first things was at what distance away from a traffic station would a CHP incident be included as a registered event for a day's readings? Looking at the frequency of how often stations occur on a highway and some random trial and error it was decided to use a half mile. Any interesting exploration could be done in varying the distance at which a CHP incident is from a station to determine what effect it has upon the flow of that station.

What follow are sample queries for pulling the requisite features from the various fact and feature tables to join them into a single row. The YearDOYToDate function is a simple macro that takes a Year and the DOY value and returns a Date value.

Some interesting notes are the distinction between INNER and LEFT OUTER joins for preventing the loss of the observation rows and making sure that null values are returned. Additionally the temporal range checks used for joining in data from CHP and ZHVI features. By converting the Latitude and Longitude values into a Geography type it allows for the use of the POSTGIS Geospatial libraries to easily calculate the distance between to two points. The ST_Distance function returns the distance in meters so 804.672 meters is equivalent to half a mile.

```
SELECT t.ID, t.Num_Lanes, t.Length, t.Urban, t.Density,
       f.Num,
       CASE f.Direction
       WHEN 'N' THEN 1 WHEN 'E' THEN 3 WHEN 'S' THEN 2 WHEN 'W' THEN 4 ELSE -1
       END,
       z.Avg_Value,
       CASE WHEN chp.ID IS NULL THEN 'F' ELSE 'T' END,
       CAST(chp.CC_CODE AS CHAR(4)), CAST(chp.Description AS CHAR(78)), chp.Duration,
       YearDOYToDate(o.year, o.DOY),
       o.Flow_Coef[1], o.Flow_Coef[2], o.Flow_Coef[3], o.Flow_Coef[4], o.Flow_Coef[5],
       o.Flow_Coef[6], o.Flow_Coef[7], o.Flow_Coef[8], o.Flow_Coef[9], o.Flow_Coef[10]
FROM Observations o
        INNER JOIN Traffic_Station t ON (o.Station_ID=t.ID)
```

```
        INNER JOIN ST_Type st ON (t.Type_ID=st.ID AND st.type='ML')
        INNER JOIN Freeways f ON (f.ID=t.Fwy_ID)
        LEFT OUTER JOIN Zillo_Home_Value z ON
            ((EXTRACT(YEAR FROM z.month)=o.year) AND
             EXTRACT(MONTH FROM z.month)=EXTRACT(MONTH FROM YearDOYToDate(o.year, o.DOY)))
        INNER JOIN County_Zip cz ON (t.ZIPCODE=cz.ZIPCODE AND cz.ZIPCODE=z.ZIPCODE)
        LEFT OUTER JOIN CHP_INC chp ON (
            CAST(chp.time AS DATE)=YearDOYToDate(o.year, o.DOY)
            AND chp.Fwy_ID=t.Fwy_ID
            AND ST_Distance(chp.Location, t.Location) < 804.672 -- Half-Mile away
        )
WHERE o.year={y}
ORDER BY 1, 13;

SELECT t.ID, t.Num_Lanes, t.Length, t.Urban, t.Density,
       f.Num,
       CASE f.Direction
       WHEN 'N' THEN 1 WHEN 'E' THEN 3 WHEN 'S' THEN 2 WHEN 'W' THEN 4 ELSE -1
       END,
       z.Avg_Value,
       CASE WHEN chp.ID IS NULL THEN 'F' ELSE 'T' END,
       CAST(chp.CC_CODE AS CHAR(4)), CAST(chp.Description AS CHAR(78)), chp.Duration,
       YearDOYToDate(o.year, o.DOY),
       o.weekend_coef[1], o.weekend_coef[2], o.weekend_coef[3],
       o.weekend_coef[4], o.weekend_coef[5]
FROM Observations o
        INNER JOIN Traffic_Station t ON (o.Station_ID=t.ID)
        INNER JOIN ST_Type st ON (t.Type_ID=st.ID AND st.type='ML')
        INNER JOIN Freeways f ON (f.ID=t.Fwy_ID)
        LEFT OUTER JOIN Zillo_Home_Value z ON
            ((EXTRACT(YEAR FROM z.month)=o.year) AND
             EXTRACT(MONTH FROM z.month)=EXTRACT(MONTH FROM YearDOYToDate(o.year, o.DOY)))
        INNER JOIN County_Zip cz ON (t.ZIPCODE=cz.ZIPCODE AND cz.ZIPCODE=z.ZIPCODE)
        LEFT OUTER JOIN CHP_INC chp ON (
            CAST(chp.time AS DATE)=YearDOYToDate(o.year, o.DOY)
            AND chp.Fwy_ID=t.Fwy_ID
            AND ST_Distance(chp.Location, t.Location) < 804.672 -- Half-Mile away
        )
WHERE o.year={y}
AND EXTRACT(DOW FROM YearDOYToDate(o.year, o.DOY)) IN (1,7)
ORDER BY 1, 13;

SELECT t.ID, t.Num_Lanes, t.Length, t.Urban, t.Density,
       f.Num,
       CASE f.Direction
       WHEN 'N' THEN 1 WHEN 'E' THEN 3 WHEN 'S' THEN 2 WHEN 'W' THEN 4 ELSE -1
       END,
       z.Avg_Value,
       CASE WHEN chp.ID IS NULL THEN 'F' ELSE 'T' END, CAST(chp.CC_CODE AS CHAR(4)),
       CAST(chp.Description AS CHAR(78)), chp.Duration,
       YearDOYToDate(o.year, o.DOY),
       o.weekday_coef[1], o.weekday_coef[2], o.weekday_coef[3],
       o.weekday_coef[4], o.weekday_coef[5]
```

```
FROM Observations o
        INNER JOIN Traffic_Station t ON (o.Station_ID=t.ID)
        INNER JOIN ST_Type st ON (t.Type_ID=st.ID AND st.type='ML')
        INNER JOIN Freeways f ON (f.ID=t.Fwy_ID)
        LEFT OUTER JOIN Zillo_Home_Value z ON
            ((EXTRACT(YEAR FROM z.month)=o.year) AND
             EXTRACT(MONTH FROM z.month)=EXTRACT(MONTH FROM YearDOYToDate(o.year, o.DOY)))
        INNER JOIN County_Zip cz ON (t.ZIPCODE=cz.ZIPCODE AND cz.ZIPCODE=z.ZIPCODE)
        LEFT OUTER JOIN CHP_INC chp ON (
            CAST(chp.time AS DATE)=YearDOYToDate(o.year, o.DOY)
            AND chp.Fwy_ID=t.Fwy_ID
            AND ST_Distance(chp.Location, t.Location) < 804.672 -- Half-Mile away
        )
WHERE o.year={y}
AND EXTRACT(DOW FROM YearDOYToDate(o.year, o.DOY)) NOT IN (1,7)
ORDER BY 1, 13;
```

Once the queries were run the resulting CSV files were stored on s3 at s3://dse-team2-2014/regression/trim_<year>_<type> for the team to process using python. Year is in the range 2008 to 2015 and type is one of wkday for PCA results of just Monday-Friday station readings, wkend for PCA results of just Saturday and Sunday, and wkfull for PCA results of all days in the week.

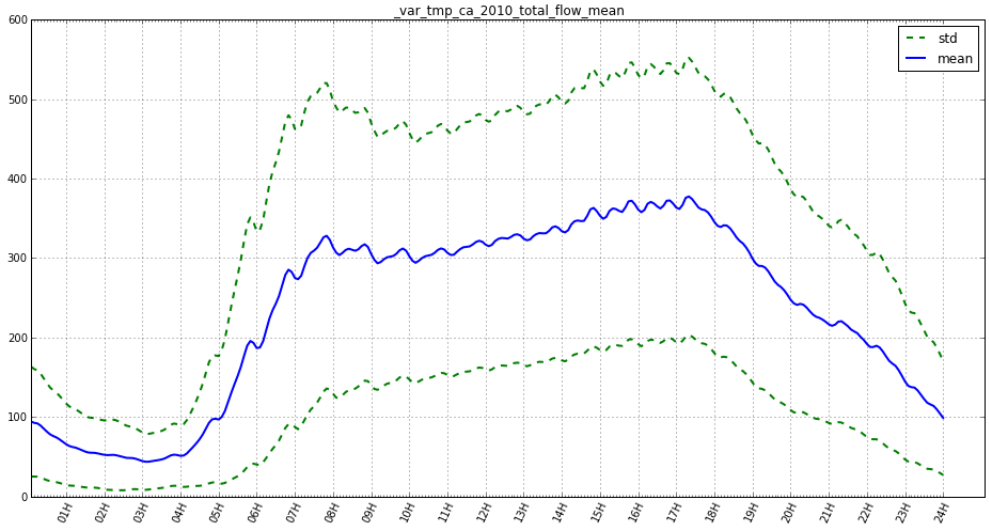
Data Exploration

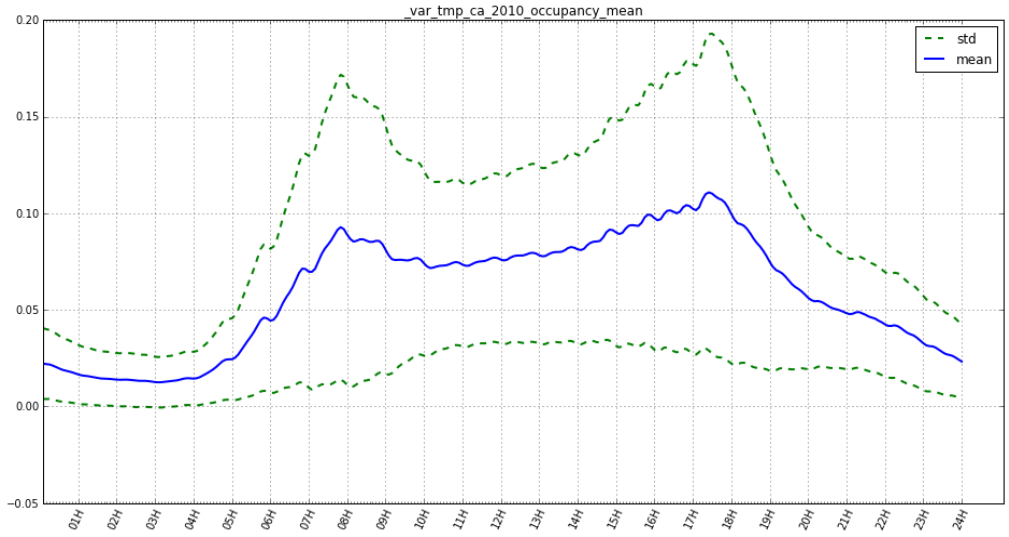
Method

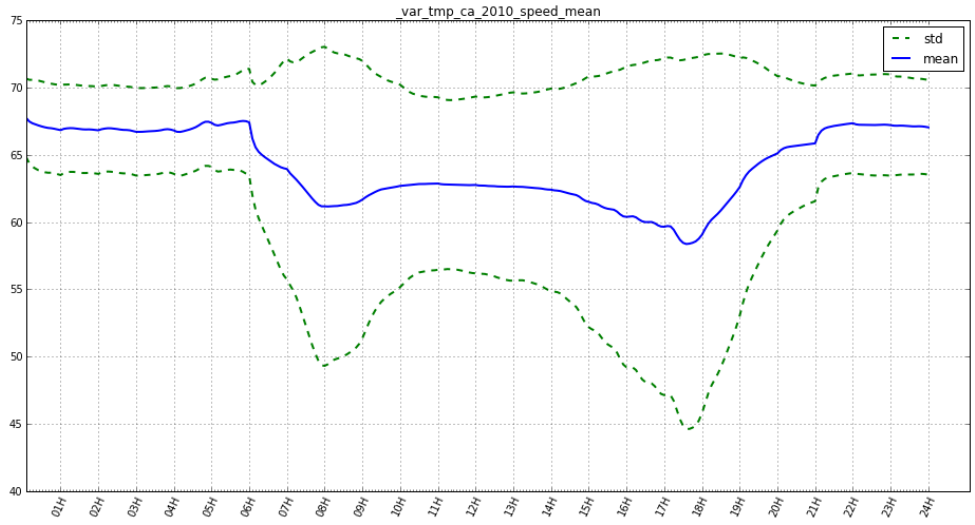
To gain more insight into the CalTrans dataset, the mean, standard deviation and top four eigenvectors were calculated from the pivoted traffic data for the total flow, speed, and occupancy in the year 2010. Additionally, a scatter density plot depicting the first two eigenvector coefficients for total flow was created (see below).

Results

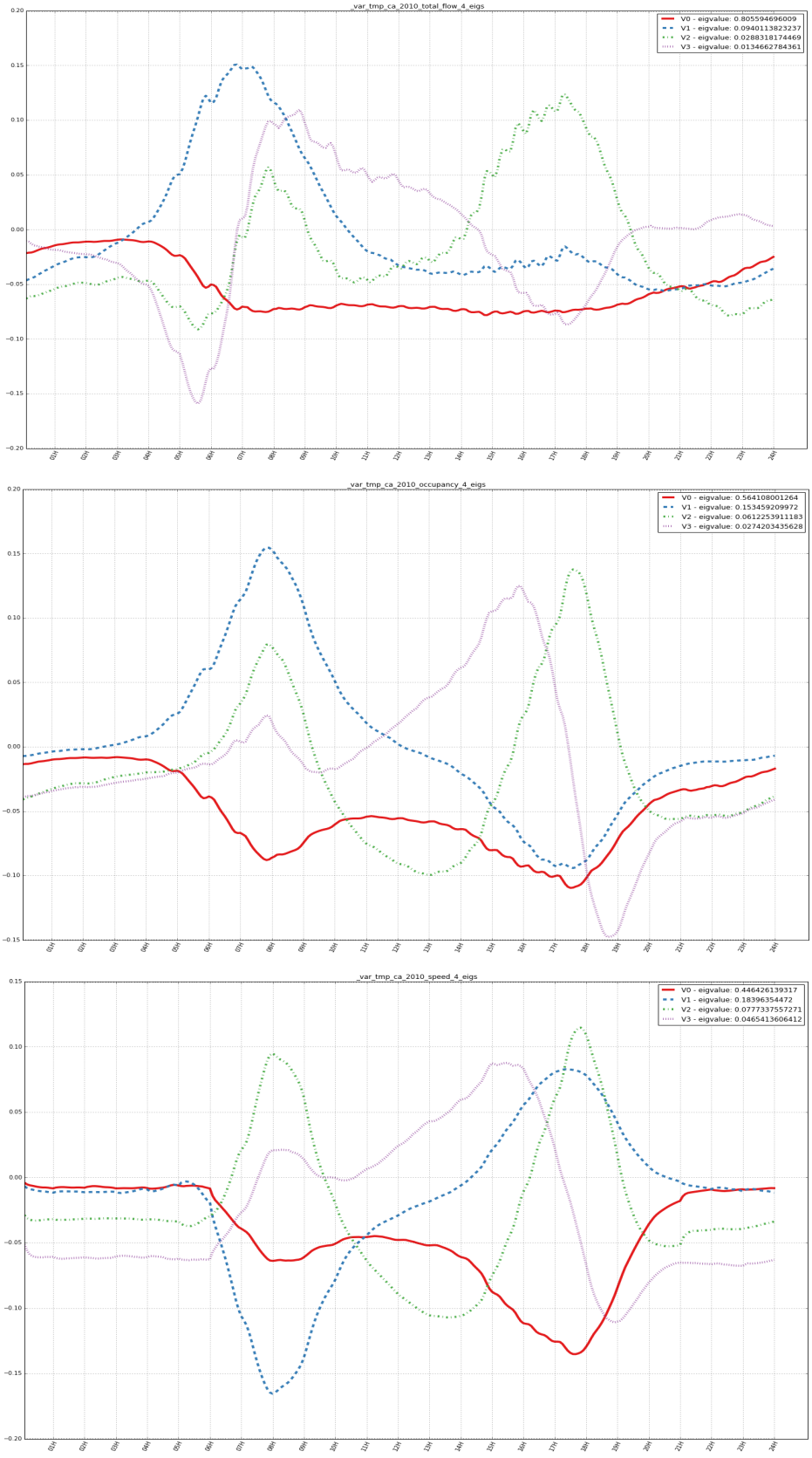
The following graphs depict the mean and std for total flow, occupancy, and speed for CA 2010, respectively:





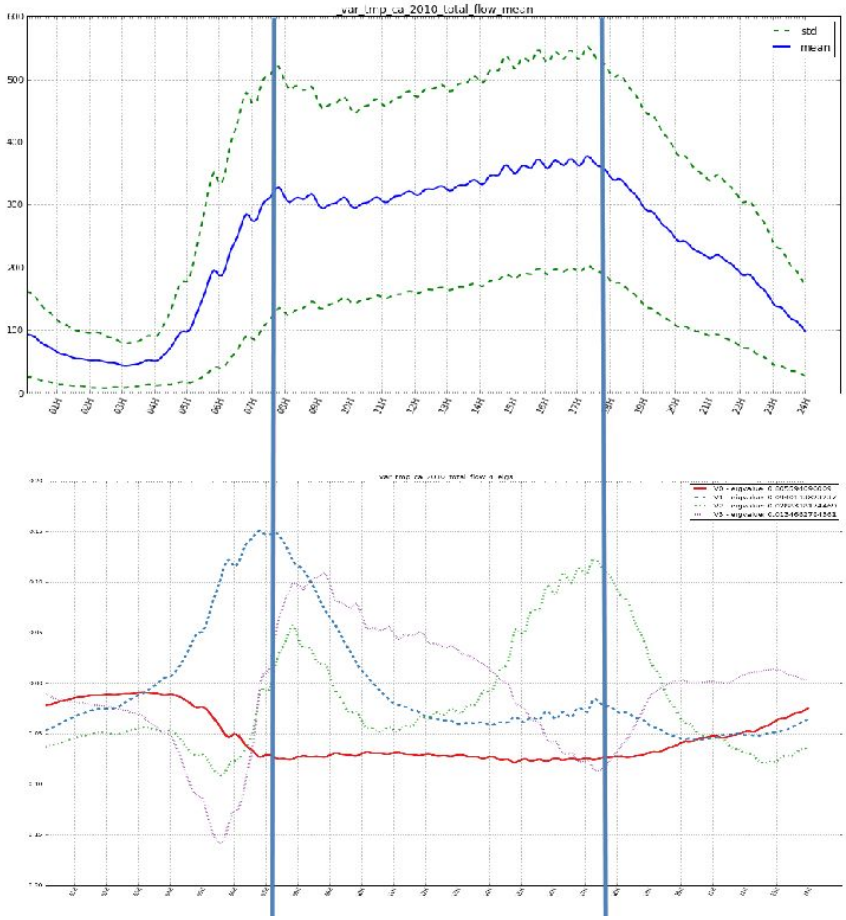


The following graphs depict the top 4 eigenvectors for total flow, occupancy, and speed for CA 2010 respectively:



Preliminary analysis of the mean for each group was executed. Total flow, which measures traffic volume, is directly proportional to occupancy, which measures vehicle occupancy across lanes, and is inversely proportional to speed. Traffic volume, is high between the hours of 6AM and 9PM, as can be seen from the mean of total flow.

From these observations it was decided that the focus of any further analysis should be done solely on the total flow.

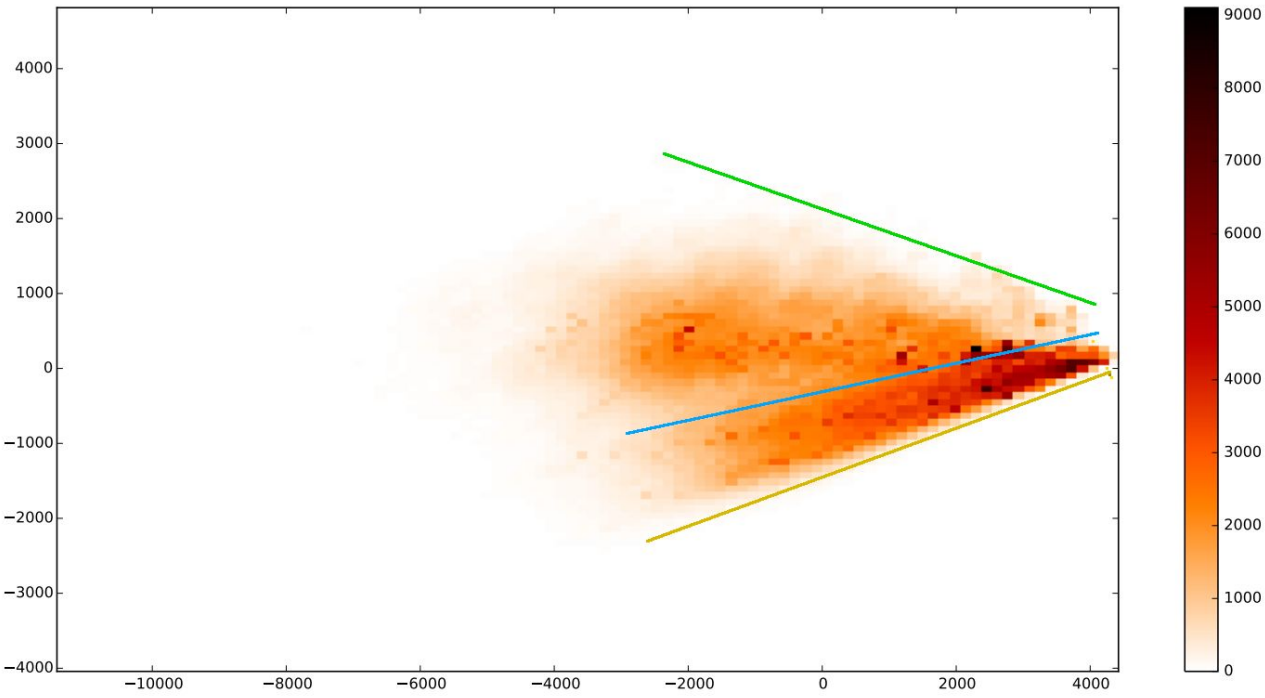


The blue lines on the figure above, outline the time of day corresponding to the total flow's highest values across the mean and eigenvector graphs. The eigenvectors within this range exhibit interesting behavior, and represent certain aspects of total flow. Using Bokeh and a small interactive visualization, the following can be deduced about what each eigenvector represents:

- V0 → Always negative and is the negation of the mean vector. Represents the rise and fall of total flow and overall traffic volume
- V1 → Is positive in the morning and negative in the evening. Represents the flow of traffic exhibited in the morning rush hour versus the evening rush hour.
- V2 → Is positive in both morning and night, with a greater magnitude in the evening. Represents shifts in peaks and valleys in the total flow of the morning rush hour and evening rush hour.
- V3 → Is positive in the morning and dips negative in the evening. Represents minor deviations in total flow of when the morning/afternoon rush hour starts and ends

A short demonstration of the Bokeh tool can be viewed at https://youtu.be/KeEEMWgGXC8, and the source for generating the interactive tool is found at https://github.com/conwaywong/dse_capstone/blob/master/final/eigenvector_analysis.py.

A scatter density plot was created to understand these dynamics of total flow further:



The x axis corresponds to the first eigenvector (negation of the mean) and the y axis corresponds to the second eigenvector (rush hour of morning vs night). The majority of station day readings throughout CA in 2010 falls into the right side of the density plot (as per the color map legend). This area represents low traffic volume - ie. when the V0 coefficient is large, the total flow approaches zero. This would imply that there are less cars on the road. The y axis in this situation (depicting morning and evening rush hours) is near zero, and implies that there is little to no variation between the morning and evening hours with respect to total flow. To the left of the density plot, there is a smaller distribution of station day readings. This area represents high volume traffic. The y axis here has a wide variation, and implies that morning and evening hour rush hours exist, and total flow readings are high. There also appears to be two modes in the overall distribution seen in the lower section (between the blue and yellow line) being more concentrated and the higher section (between the green and blue line) more diffuse. This bimodal distribution leads to possibly different partitioning strategies in the data, such as segregation between weekday and weekend traffic; a strategy that was taken moving forward in the analysis.

5. Analysis Methods

To address the following questions:

- How can the flow of traffic on a highway be modeled?
- Can the variation in traffic patterns be expressed by a small number of variables?
- What is the overall pattern along a particular busy highway such as Hwy 5?

Principal Component Analysis (PCA) was selected. PCA inherently provides a compressed way to model traffic patterns using a small number of dimensions. In addition, the resulting principal components can be utilized to reconstruct traffic patterns for any particular PeMS traffic station in CA, thus identifying traffic patterns and explaining variation with a small number of variables. To assess traffic pattern reconstruction, the normalized root mean square error (NRMSE) is calculated between the reconstruction and the actual sensor reading values.

Expanding further in the aforementioned analysis, segregation of the data into weekday and weekend partitions was done. PCA and KMeans clustering was executed against the two partitions to identify any potentially different traffic behaviors. As the model is unsupervised, no assessment criteria was applied. See the results section for more details.

To address the following question:

- What factors have the most influence on traffic?

An Elastic Net Regressor was selected. The Elastic Net Regressor combines L1 and L2 Regularization which intrinsically executes feature selection (L1 Regularization forces weaker features to have zero coefficients) while simultaneously maintaining correlated features (L2 Regularization keeps grouping effect, correlated features tend to have similar coefficients). The magnitude and sign of each feature coefficient provides an indication of how total flow increases (positive sign) or decreases (negative sign) with respect to a given feature.

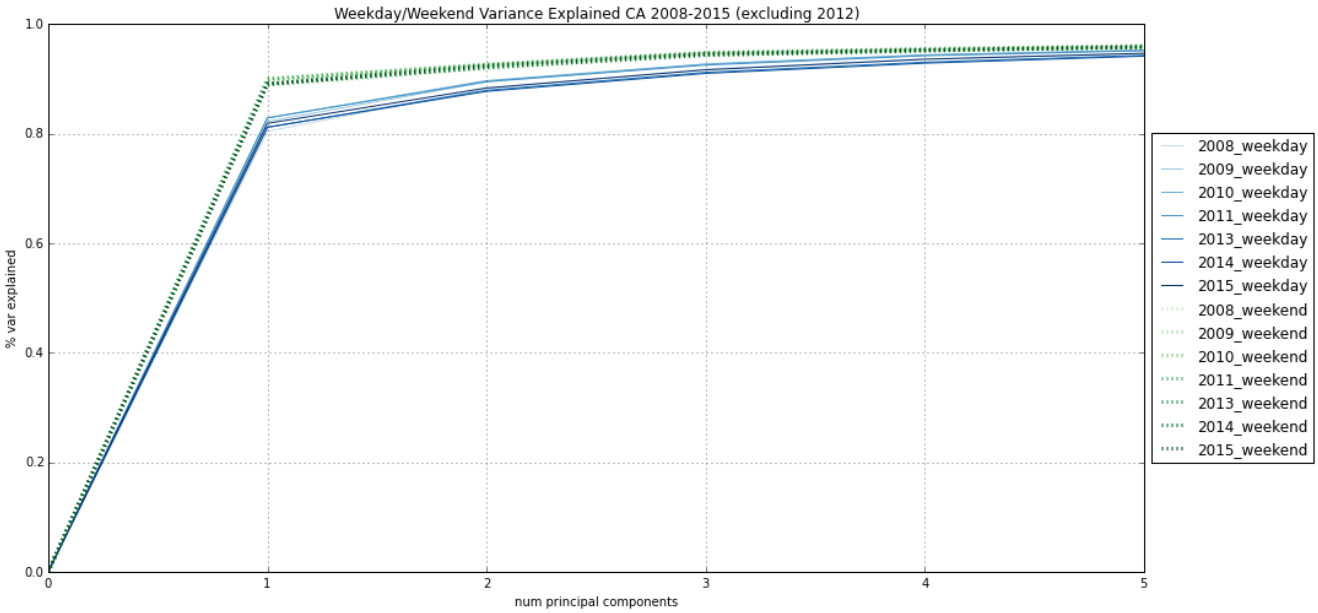
A note on execution; the following techniques/technologies were used:

- PCA
    - Execution: Scala, SparkMLib, SparkML, Spark SQL (Dataframes), Amazon S3
    - Visualization: Python, matplotlib, numpy, pandas, Jupyter Notebook
- KMeans++
    - Execution: Scala, SparkMLib, SparkML, Spark SQL (Dataframes), Amazon S3
    - Visualization: Python, matplotlib, numpy, pandas, Jupyter Notebook
- Elastic Net Regression
    - Preprocessing: Postgres, SQL, Python, numpy, pandas
    - Execution: Scala, SparkML, Spark SQL (Dataframes), Amazon S3
    - Visualization: Python, matplotlib, numpy, pandas, Jupyter Notebook

6. Findings

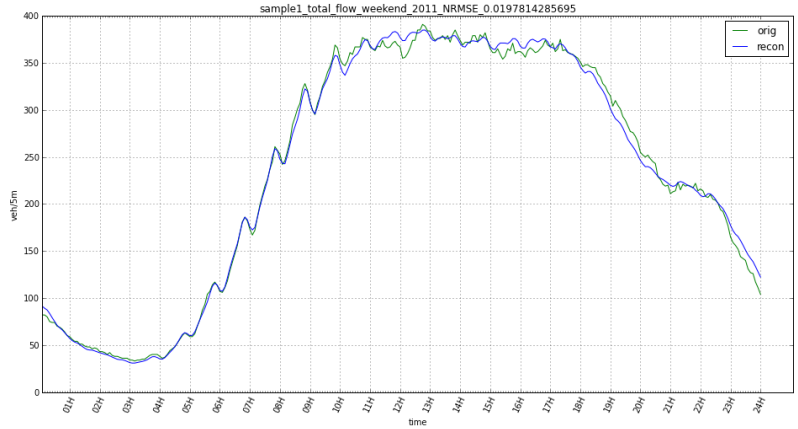
PCA

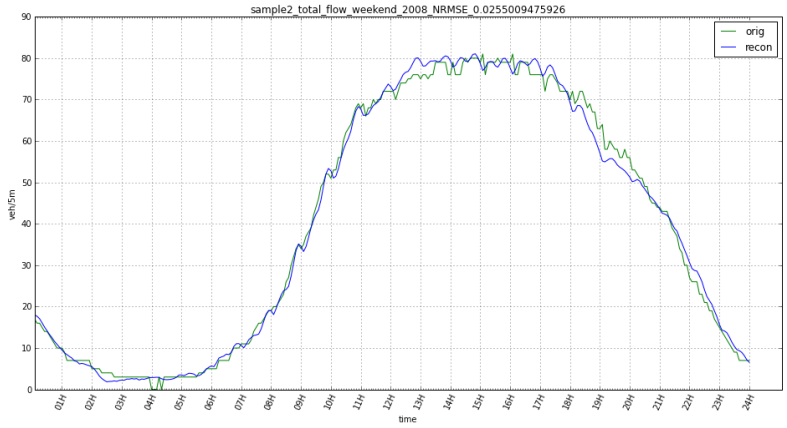
The explained variance of the top five principal components was obtained and plotted by executing PCA against the pivoted total flow data (n X 288):

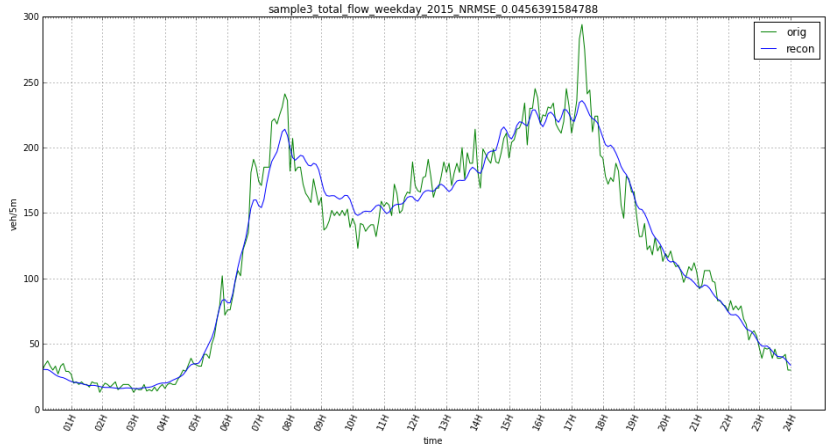


The explained variance is well over 90% when using five principal components for each year and partition combination. This implies that five principal components are ideal in modeling and explaining the variation in total flow using a small number of variables. To confirm this hypothesis, three original readings from each year and partition were selected at random, and vector reconstruction using the top five principal components was executed.

NRMSE was calculated between the reconstruction and the original vector to determine goodness of fit. The graphs of the three reconstructions are shown below:



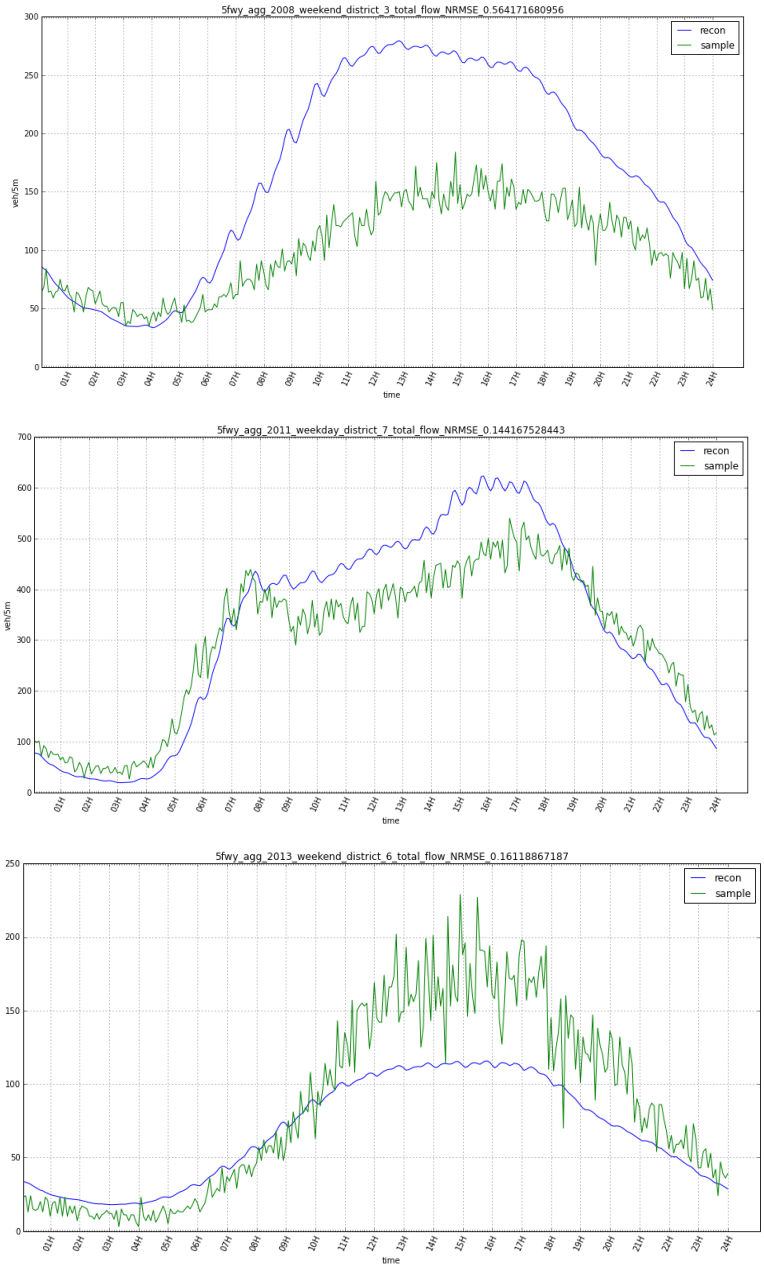




From the results, the NRMSE between the reconstruction and the original vector is low for each year and partition (station+day combination). It can be concluded that PCA and utilizing the resulting principal components for reconstruction is accurate in modeling total flow and explaining the variation in total flow using a small number of variables (addressing questions 1 and 2).

To determine the overall traffic pattern along a busy highway, such as Highway 5, station+day combinations (containing the resulting five eigenvector coefficients after projection) were grouped by district and freeway (Highway 5), and aggregated, extracting the mean vector of the coefficients. Reconstruction was executed against

the mean vector, and compared against sample readings taken at random from partition+year+district combinations:



In the above reconstructions, the following sample combinations were used (all combinations along Highway 5):

- year=2008, partition='wkend', district=3, station=316096
- year=2011, partition='wkday', district=7, station=716985
- year=2013, partition='wkend', district=6, station=601310

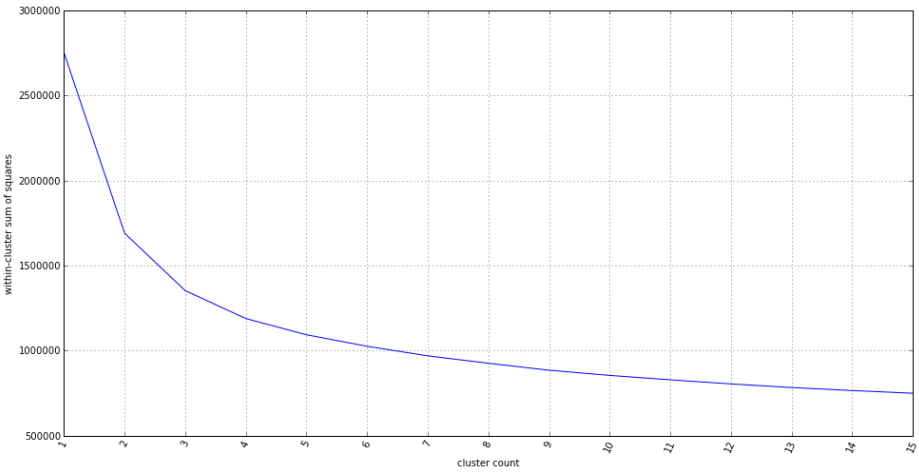
The NRMSE tends to be higher in these reconstructions, compared to the previous reconstructions. This is expected, due to the following:

- Aggregation is being done for Highway 5 reconstruction. Prior reconstruction utilizes the corresponding eigenvector coefficients for a given station+day combination without aggregation; the Highway 5 reconstruction executes aggregation across station+district+day combinations, and then uses resulting mean vector for reconstruction. This aggregation results in data loss.
- The sampling process is random and can potentially select an outlier, a combination that can deviate heavily from the mean and lie on the outskirt of the distribution.

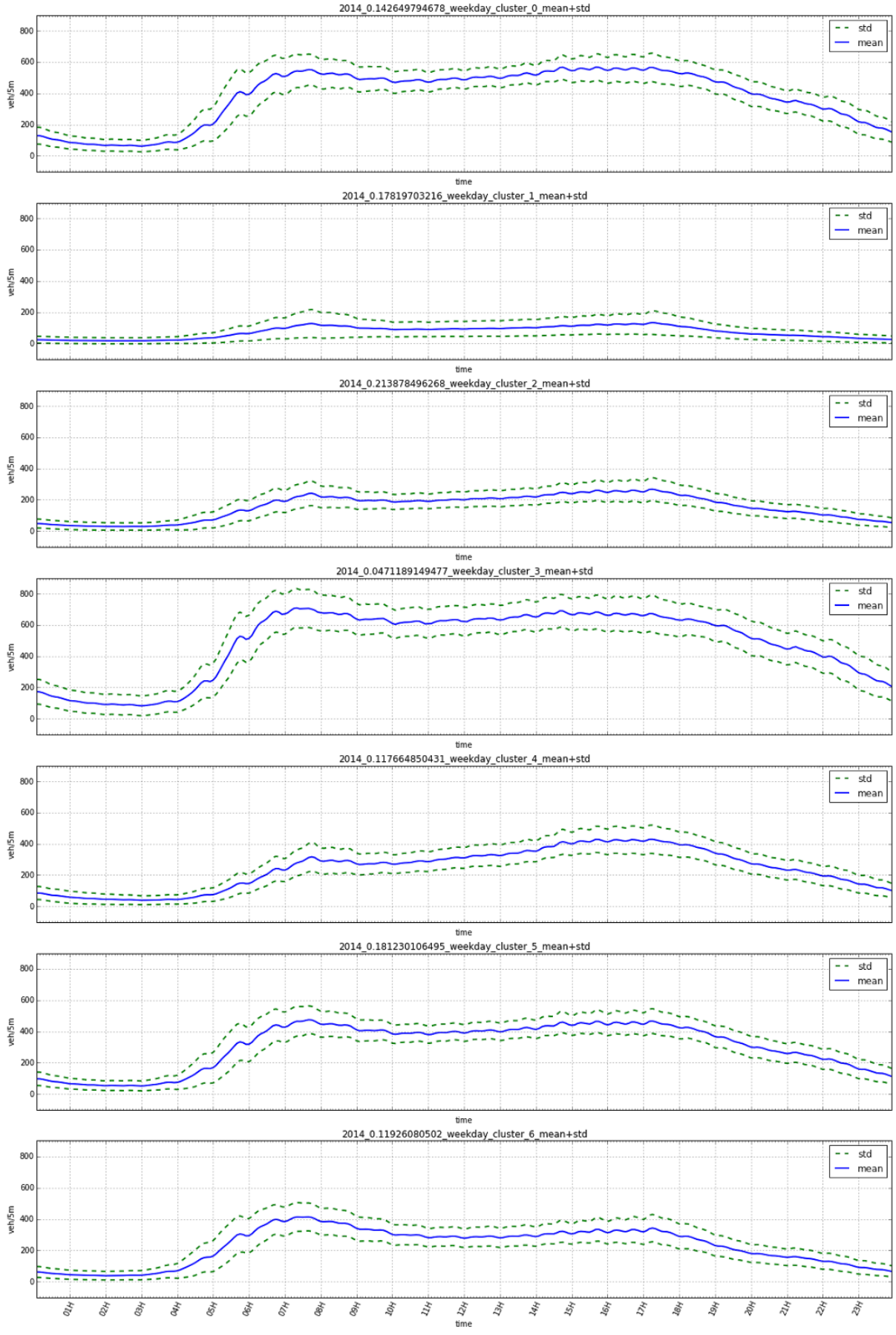
Regardless of these conditions, Highway 5 reconstructions in the second and third plot exhibit sufficiently low NRMSE. It can be concluded that using this model, the overall pattern for a particular highway, such as Highway 5, can be determined with reasonable accuracy (addressing question 3).

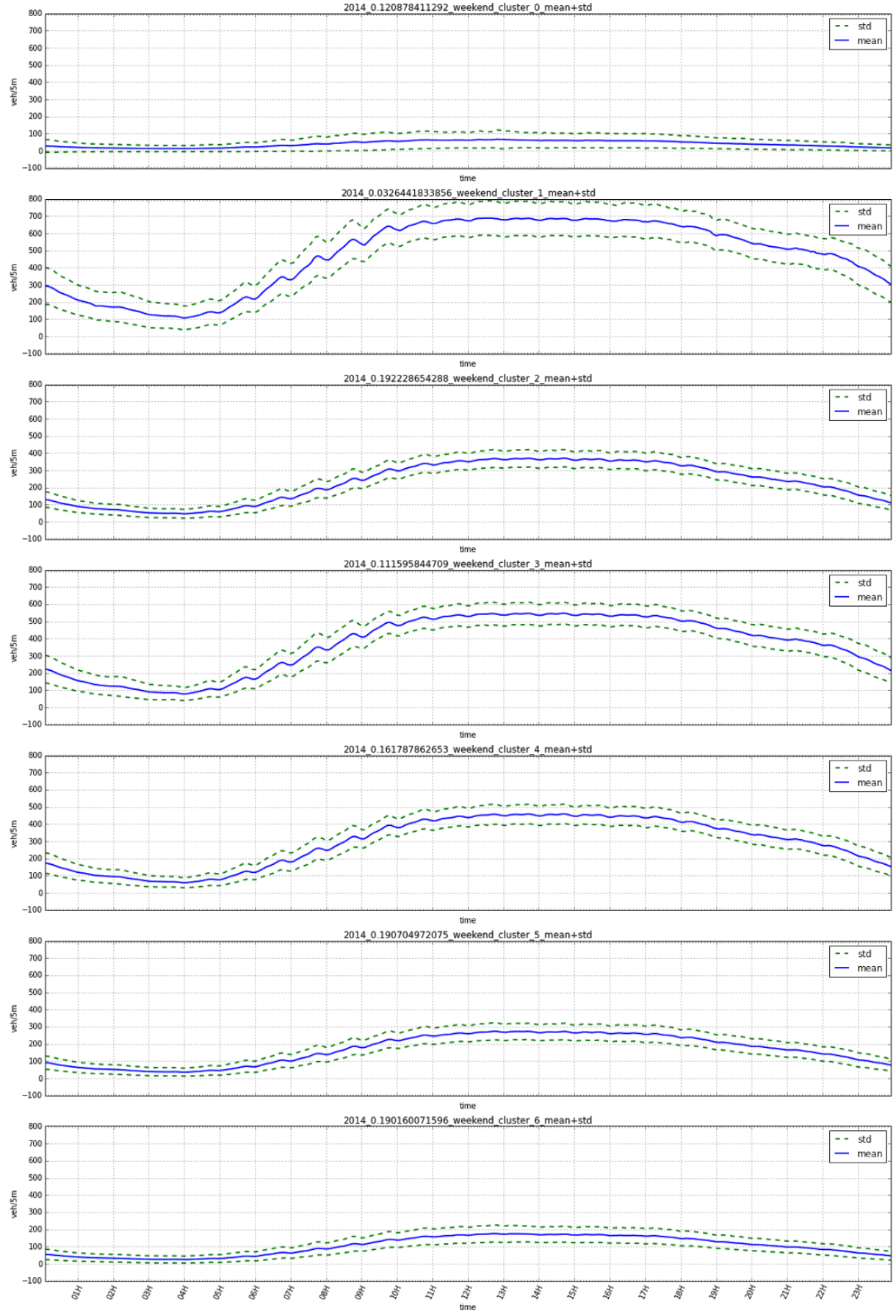
K-Means++

A supplementary analysis using KMeans++ was executed to identify potential differences in traffic behavior between the weekday and weekend data partitions, using 2014 as a sample. The optimal cluster count of 7 was selected by identifying the "elbow" in the within-cluster sum of squares plot.

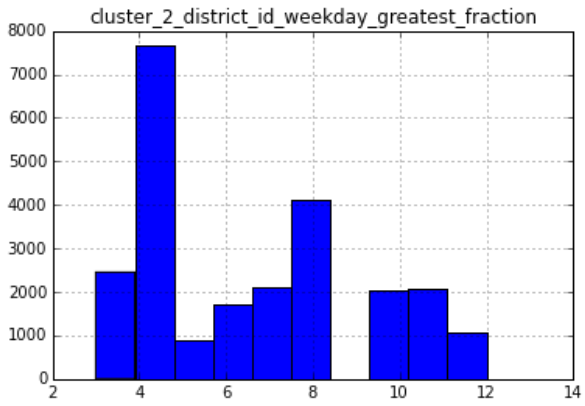


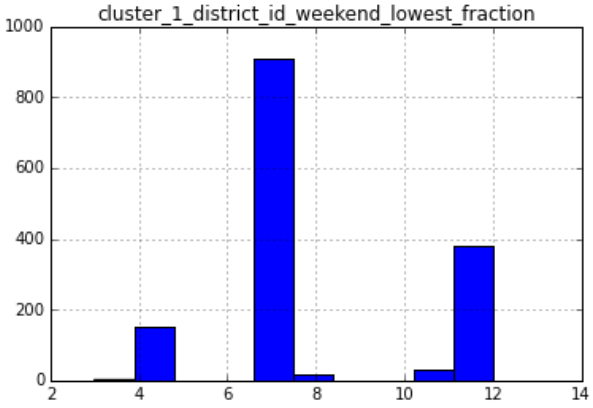
To visualize each cluster and behavior of the partitions, the mean + std were plotted, and the fraction of the population for each cluster was calculated:

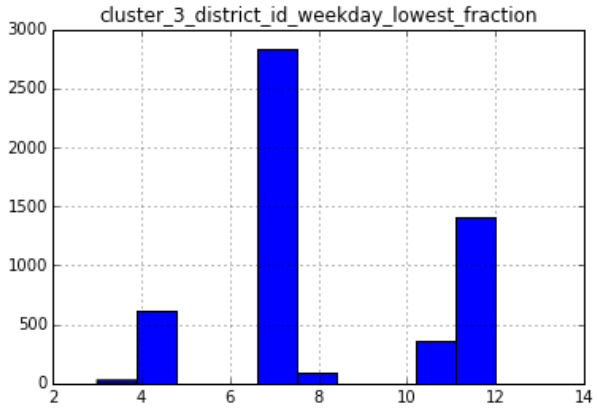

2014_0.142649794678_weekday_cluster_0_mean+std
2014_0.17819703216_weekday_cluster_1_mean+std
2014_0.213878496268_weekday_cluster_2_mean+std
2014_0.0471189149477_weekday_cluster_3_mean+std
2014_0.117664850431_weekday_cluster_4_mean+std
2014_0.181230106495_weekday_cluster_5_mean+std
2014_0.11926080502_weekday_cluster_6_mean+std
std
mean
veh/5m
time
0
200
400
600
800
01H
02H
03H
04H
05H
06H
07H
08H
09H
10H
11H
12H
13H
14H
15H
16H
17H
18H
19H
20H
21H
22H
23H

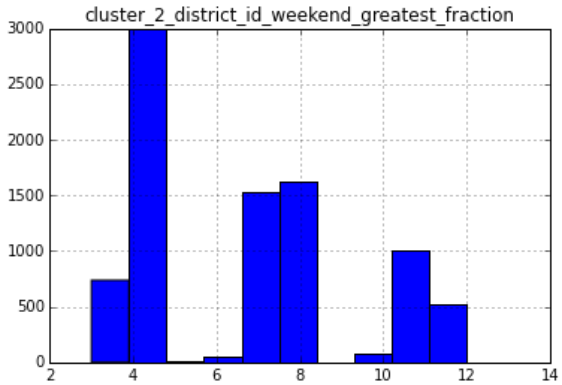

2014_0.120878411292_weekend_cluster_0_mean+std
2014_0.0326441833856_weekend_cluster_1_mean+std
2014_0.192228654288_weekend_cluster_2_mean+std
2014_0.111595844709_weekend_cluster_3_mean+std
2014_0.161787862653_weekend_cluster_4_mean+std
2014_0.190704972075_weekend_cluster_5_mean+std
2014_0.190160071596_weekend_cluster_6_mean+std
std
mean
veh/5m
time
800
700
600
500
400
300
200
100
0
−100
01H
02H
03H
04H
05H
06H
07H
08H
09H
10H
11H
12H
13H
14H
15H
16H
17H
18H
19H
20H
21H
22H
23H

Samples were taken from each cluster group, in order to gain insight into the geographic location of these clusters. The clusters with the greatest and lowest fractions of each partition (weekday/weekend) were selected. The district id distribution was plotted:









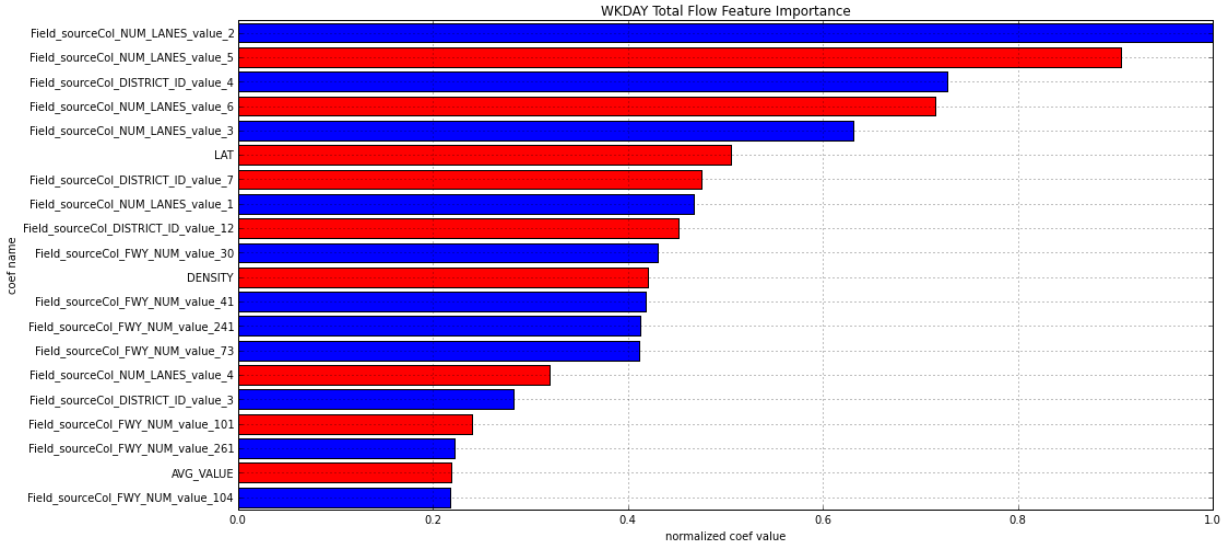
From the plots, cluster 2 and cluster 3 contain the greatest and lowest fraction of the weekday partition population respectively (top and bottom left). And cluster 1 and cluster 2 contain the lowest and greatest fraction of the weekend partition population (top and bottom right). Note that the cluster ids are assigned arbitrarily across clusters and are not correlated in any way.

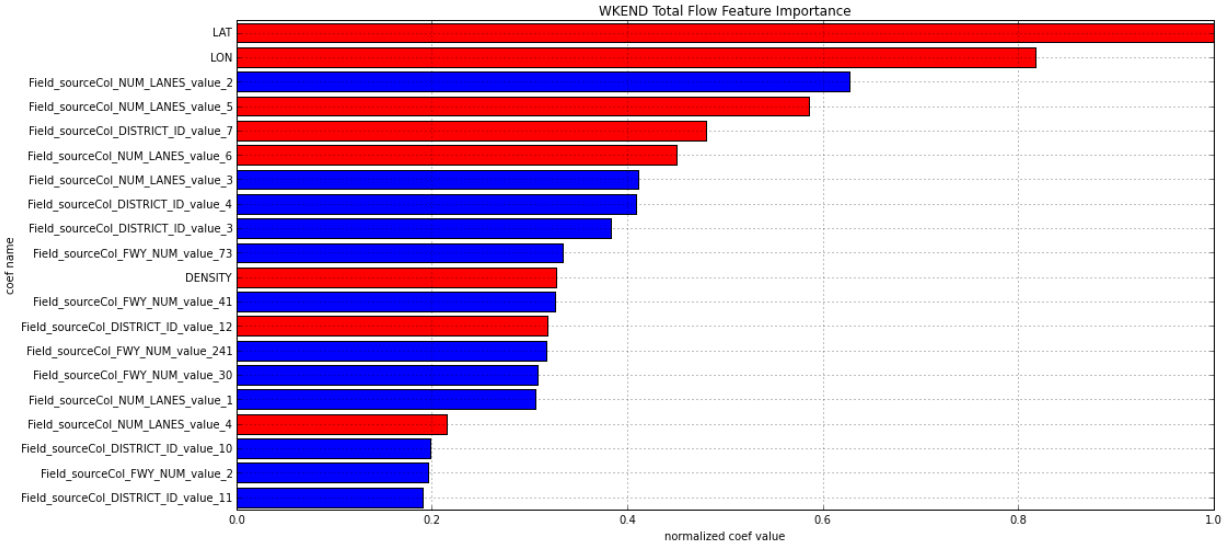
Interestingly, the majority of stations in both weekday and weekend partitions exhibit lower than average total flow (around 200 veh/5m at their peak) and is located in district 4 (San Francisco Bay Area). The minority of stations in both weekday and weekend partitions exhibit double the average total flow (around 600 veh/5m) and is located in district 7 (Los Angeles County).

Elastic Net Regression

Prior to executing regression, the data across Zillow, CHP incidents, CalTrans traffic data, Urban Areas, US Census, Zip Codes, and PCA Eigenvector Coefficients were combined into a set of CSVs for each year and weekday/weekend partition. Feature Engineering was completed against these CSV files using mechanisms such as hot one encoding, timestamp processing, PCA reconstructions, and Standard Scaling.

After Feature Engineering, Elastic Net Regression was executed. The following coefficients were identified:





Red represents positive coefficients (contributes to total flow increase), and blue represents negative coefficients (contributes to total flow decrease). The normalized coef value is obtained by normalizing the absolute value of the coefficient value with respect to the maximum value in each partition distribution; a value closer to 1 implies greater importance.

Looking more deeply into the weekday partition, the factors that contribute to increases in total flow are:

- Field_sourceCol_NUM_LANES_value_5
- Field_sourceCol_NUM_LANES_value_6
- LAT
- Field_sourceCol_DISTRICT_ID_value_7
- Field_sourceCol_DISTRICT_ID_value_12
- DENSITY
- Field_sourceCol_NUM_LANES_value_4
- Field_sourceCol_FWY_NUM_value_101
- AVG_VALUE

The factors that contribute to decreases in total flow are:

- Field_sourceCol_NUM_LANES_value_2
- Field_sourceCol_DISTRICT_ID_value_4
- Field_sourceCol_NUM_LANES_value_3
- Field_sourceCol_NUM_LANES_value_1
- Field_sourceCol_FWY_NUM_value_30
- Field_sourceCol_FWY_NUM_value_41
- Field_sourceCol_FWY_NUM_value_241
- Field_sourceCol_FWY_NUM_value_73
- Field_sourceCol_DISTRICT_ID_value_3
- Field_sourceCol_FWY_NUM_value_261
- Field_sourceCol_FWY_NUM_value_104

Interestingly, districts in Southern California (7 and 12), whether or not the freeway is Highway 101, and average value of home, are strong indicators of high total flow (high traffic volume) during weekdays. Meanwhile, a lower number of lanes, whether or not the highway is a toll road (73, 241, and 261 are toll roads), and districts in Northern California (3 and 4) are strong indicators of low total flow (low traffic volume).

Looking more deeply into the weekend partition, the factors that contribute to increases in total flow are:

- LAT
- LON
- Field_sourceCol_NUM_LANES_value_5
- Field_sourceCol_DISTRICT_ID_value_7
- Field_sourceCol_NUM_LANES_value_6
- DENSITY
- Field_sourceCol_DISTRICT_ID_value_12
- Field_sourceCol_NUM_LANES_value_4

The factors that contribute to decreases in total flow are:

- Field_sourceCol_NUM_LANES_value_2
- Field_sourceCol_NUM_LANES_value_3
- Field_sourceCol_DISTRICT_ID_value_4
- Field_sourceCol_DISTRICT_ID_value_3
- Field_sourceCol_FWY_NUM_value_73
- Field_sourceCol_FWY_NUM_value_41
- Field_sourceCol_FWY_NUM_value_241
- Field_sourceCol_FWY_NUM_value_30

- Field_sourceCol_NUM_LANES_value_1
- Field_sourceCol_DISTRICT_ID_value_10
- Field_sourceCol_FWY_NUM_value_2
- Field_sourceCol_DISTRICT_ID_value_11

Interestingly, districts in Southern California (7 and 12) and location (LAT and LON), are strong indicators of high total flow (high traffic volume) during weekends. Meanwhile, a lower number of lanes, whether or not the highway is a toll road (73, 241 are toll roads), and districts in Northern California (3 and 4) are strong indicators of low total flow (low traffic volume).

Using Elastic Net Regression and its resulting coefficients, one can identify the factors that have the most influence on traffic and total flow (addressing question 4).

Visualization

Two visualizations were created to help with interpreting our findings.

The following visualization (https://youtu.be/KeEEMWgGXC8), constructed with Bokeh, provides a mechanism to understand the total flow eigenvector dynamics, and to determine what each eigenvector represents.

The following Traffic GIS visualization (https://youtu.be/PFqy4ycDOsA), constructed with Javascript and other Javascript technologies, provides the following functionality:

- Display each PeMS traffic station on a map using its latitude/longitude
- Load yearly (2008-2015) traffic volume data sets
- Color-encode traffic stations using its V0 and V1 eigenvector coefficients via a diverging color scheme
- Filter traffic stations by direction, freeway number, or coefficient value
- Find traffic stations by ID and zoom to station
- Reconstruct traffic volume readings of a station for any day

Using the Traffic GIS visualization tool, one can compare traffic volume across stations. The tool allows the end user to select a station day, and for that station reconstruct the traffic volume for a selected day. Multiple days can be plotted for a given station, and multiple stations can be displayed so station volumes can be compared and contrasted. The reconstruction capability is demonstrated in the recording at https://youtu.be/PFqy4ycDOsA?t=427.

Using the Traffic GIS visualization tool, one can also visualize traffic patterns across California. For example, in the demonstration video, the 2015 dataset filtered on eastbound San Diego stations and viewing the V1 coefficient values, shows that most freeways in San Diego County have high traffic volume during the afternoon (https://youtu.be/PFqy4ycDOsA?t=228) Panning the map to the Los Angeles area, it is quickly noticed that LA eastbound stations exhibit similar patterns; almost all eastbound LA highways are most voluminous during the afternoon. The one exception that is observed out of the norm is the northern LA County Area; the eastbound highways in the northern areas show higher volumes during the morning (https://youtu.be/PFqy4ycDOsA?t=248).

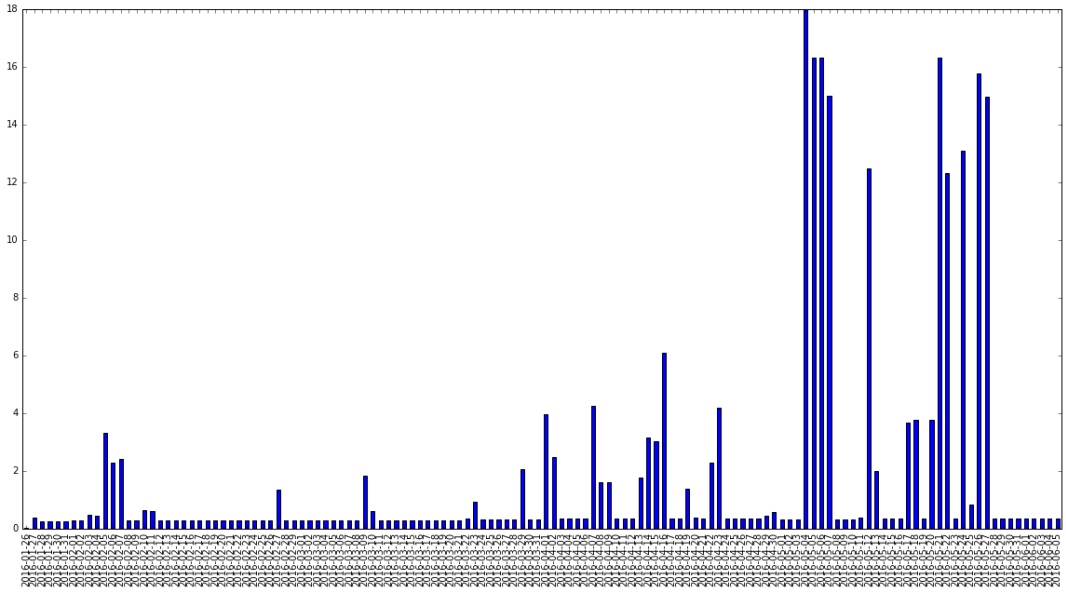
Another example of traffic pattern detection using the Traffic GIS visualization tool, is being able to infer destinations. The video at https://youtu.be/PFqy4ycDOsA?t=324 shows the San Diego northbound sections of I-5 and I-15. Filtering the dataset to this subset clearly identifies a pattern of traffic; morning traffic is south of the La Jolla/Miramar area, and afternoon traffic is north of the same boundary. This phenomenon can be observed using the tool, and similar behavior can be seen throughout California.

7. Performance and Evaluation

Normalized Root Mean Square Error was utilized as the performance measurement of the PCA model and its reconstructions; see Analysis Methods for more details

Model execution at scale was done using Spark; see Analysis Methods for more details.

With respect to budget, Spark Databricks Clusters were started on demand, and upon completion of execution, immediately terminated. To keep costs low, we tried to use spot instances as much as possible. Even though we stored a large amount of raw traffic data in S3, our daily S3 costs were very low and averaged 34 cents a day. The total cost per day for the Databricks clusters depended on the day we used them and whether we started spot or on-demand instances. Below is the daily cost history of AWS usage for 2016. As of June 5, 2016, we had used $245.20 of the allotted $2,000. Each bar represents the daily AWS cost for 2016, and the highest single day cost was about $18. The higher costs during the latter parts of the year were likely due to starting on-demand instances rather than spot instances.



8. Conclusions

- Principal Component Analysis (PCA) is an effective way to model traffic patterns as well as providing an efficient way of storing enough information necessary to describe traffic flow for a given station-day
- KMeans++ is an eloquent way of clustering traffic data, and was successful in identifying traffic behavior throughout California for a particular day of week
- Elastic Net Regression is effective in identifying feature importance and executing feature selection
- Visualization tools are necessary to convey findings and understand results in a cogent and impactful manner
- Spark, and similar Big Data technologies, are necessary when analyzing large amounts of data

References

Other Links

http://robotics.eecs.berkeley.edu/~varaiya/papers_ps.dir/PeMS_TRB2002.pdf
http://cvrr.ucsd.edu/events/PeMS-Pravin.pdf

Appendices

DSE MAS Knowledge Applied to the Project

The following classes provided the most utility in completing the Capstone:

- DSE 200: Data exploration, data preprocessing, Feature Engineering, Python, matplotlib, numpy, Pandas
- DSE 210: PCA
- DSE 201: Postgres
- DSE 220: Feature Engineering, K-Means
- DSE 230: AWS, S3, MapReduce (understanding Spark)
- DSE 203: Loading data into Postgres
- DSE 241: Bokeh, JavaScript (D3, C3), effective visualization paradigms (diverging color scheme)

Data and Software Archive for Reproducibility

Project Links

- Source Code: https://github.com/conwaywong/dse_capstone
- Wiki: https://github.com/conwaywong/dse_capstone/wiki
- Task Tracker: https://kanbanflow.com/board/2e14fd8abb54855cf087bf25645945d3
- Group Messaging: https://dse-capstone.slack.com
- Databricks Notebooks: Databricks -> jgilliii -> Traffic Capstone
- Visualization Demo Recordings
    - Eigenvector Dynamics: https://www.youtube.com/watch?v=KeEEMWgGXC8
    - JavaScript GIS: https://www.youtube.com/watch?v=PFqy4ycDOsA

Tools

Jupyter Notebook

Jupyter Notebooks are utilized to visualize exploratory data analysis results using matplotlib and other python visualization libraries. Additionally, we learned how to integrate Jupyter notebooks with Apache Spark (Scala) so we could develop and debug Spark jobs on our local machines before deploying them into Databricks. The steps to setup the Spark/Jupyter integration are documented at https://gist.github.com/conwaywong/f1ab7cb4d131e86be6d4

Amazon S3

Amazon S3 is leveraged as the storage service for our Traffic Data. All raw PeMS data and our results are stored in S3 buckets. These S3 buckets are used by Spark Databricks Notebooks; the notebooks read all data from S3 and generate data to S3. The parent bucket for all data related to our Capstone is found at s3://dse-team2-2014/

Spark Databricks Notebook

Spark Databricks Notebooks are created using the Databricks service. From within Databricks, Spark Clusters are instantiated to be able to execute the notebooks. The notebooks call into the Scala jars libraries we have developed which get attached to the cluster. These jars contain the core Machine Learning algorithms that leverage Spark's Scala MLib API for parallel execution. The results are placed in S3 and downloaded to local machines where we make use of Jupyter Notebooks visualize the data with matplotlib.

Scala IDE

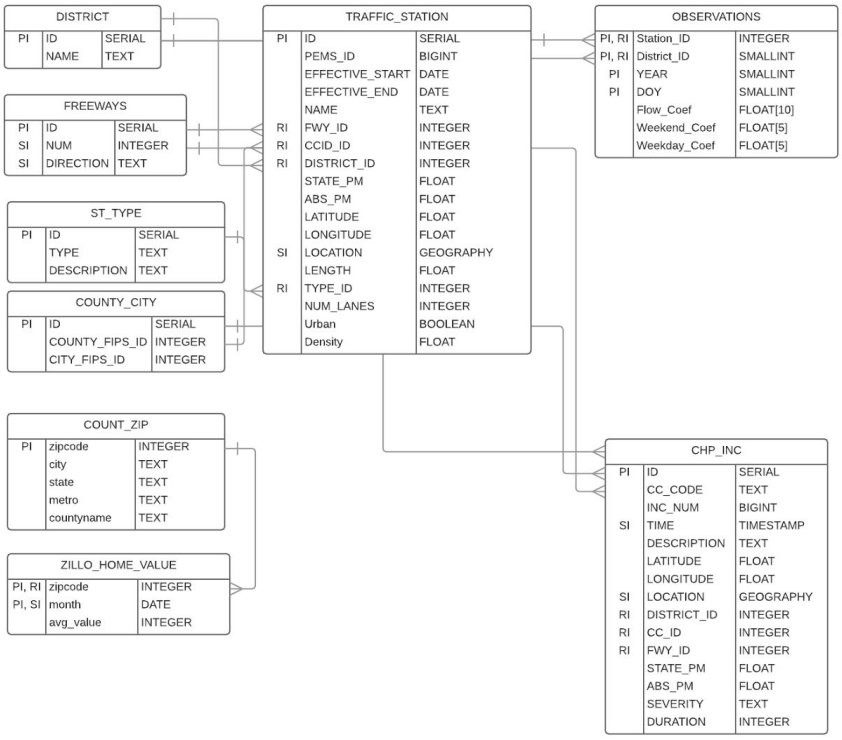
Due to the limited availability of the Python Machine Learning (MLib) library, our team decided to implement our code in scala. Unfortunately, none of the team members had experience with Scala prior to beginning this project. To improve development speed and effectiveness, Scala IDE was installed and configured on each team member's laptop. Packaging of our coded libraries (jar artifact) was built using Scala IDE and Maven, imported into Databricks, and registered with a Databricks cluster at runtime.

Technologies

PostgreSQL

The RDBMS used for processing the joins of all feature and fact tables from the various data sets. See the entity relationship diagram below for the various relationships and table definitions used.

Entity Relationship Diagram



Leaflet

The slippy map implementation for the JavaScript visualization tool leverages Leaflet for its base layers and other secondary functions such as adding a marker for a selected station. No advanced features of Leaflet were used other than using it as a Web Map Service container.

D3

The data files generated by our Spark jobs were generated in CSV format, and D3 was mainly used in the JavaScript visualization tool to parse the CSV records into JavaScript objects. Internally, the traffic stations were grouped in a D3 quadtree, and D3 was used to overlay circles at each station's respective latitude/longitude on the Leaflet map. Although not technically a part of D3, D3-queue was also utilized to load CSV files in a blocking paradigm to overcome the inherent asynchronous nature of D3.

jQuery UI

A majority of the interactive components included in the JavaScript visualization are JQuery UI widgets. Many of the widgets provided by JQuery UI are highly customizable and extremely easy to integrate into a JavaScript browser-based thin client application. The following JQuery UI widgets were used in the JavaScript vis tool:

- Autocomplete: search traffic station by station identifier
- Button: widget for filtering station direction and eigenvector selection
- Datepicker: select day to reconstruct in station pop-up frame
- Dialog: pop-up reconstruction dialog frame
- Progressbar: load data status progress bar
- Slider: filter displayed stations by eigenvector coefficient

Bokeh

Bokeh was mainly used in Eigenvector Analysis. The use of bokeh server and bokeh widgets were essential in the creation of the resulting visualization.