Sutenee Nopachinda
Udacity Machine Learning Nanodegree – Capstone Project Report
August 5, 2020

Crash Attribute Segmentation of New Jersey State Vehicle Accident Data

The following report outlines a project of using K-Means clustering algorithms to identify crash patterns across New Jersey municipalities by crash attribute segmentation.

Definition

Project Overview

Every year, the New Jersey Department of Transportation (NJDOT) publishes vehicle accident data which is derived from New Jersey Police Crash Investigation Report forms (NJTR-1) submitted by police officers. This data is used to support a variety of safety programs and planning efforts such as analysis of crashes at major intersections and calculations of fatalities and injuries in counties or municipalities.

In my work environment, the crash data is typically analyzed in a county or municipality scale using standard data analysis methods to calculate crash counts or injury counts. With these data analysis methodologies, it is difficult to examine crash patterns across the entire New Jersey State and its 21 counties. After completing the machine learning case studies in this program, I learned several algorithms that would help me explore this data in a macro level.

My motivation for this project is to explore data that I analyze daily with the machine learning algorithms I learned in this course. I want to learn more about attribute segmentation on a deeper level and see if I can utilize these new skills in a real-world data setting. I also want to improve on my data visualization skills and learn different tools to portray meaningful data to the end user.

The datasets used in this project are the 2014-2018 Crash data tables publicly provided by NJDOT. The Crash table is a summary of a crash accident and includes data columns such as location the accident took place, if alcohol was involved, the highest victim severity rating, and road conditions. Many values under these columns are numerical codes which are transcribed from New Jersey Police Crash Investigation Report forms.

As the goal of this project is to identify crash patterns across New Jersey counties, Crash tables for numerous counties will be used. For an initial test, up to five counties will be used for PCA and training a K-Means algorithm. Once the results and workflow are verified, the number of counties will be scaled up and the model will be re-trained again to identify crash patterns across all counties.

Problem Statement

The goal of this project is to identify and explore crash patterns across counties in New Jersey. As there are over 20 columns of textual data in NJTR-1 forms and over five million data entries across 21 counties, it is difficult to use standard data analysis methods to compare crash data between multiple counties and identify correlations between multiple crash attributes. Typically in my work environment, this data is filtered in a linear workflow – first by a specific county or location, then applying crash attributes such as alcohol involvement, pedestrian involvement, or person severity rating. After learning different machine learning algorithms from the case study examples, I thought this project would be a great opportunity to explore the data using unsupervised learning algorithms that I learned.

Using the Population Segmentation case study as inspiration, Principle Component Analysis (PCA) methods and K-Means clustering algorithms will be used to identify major principle components between crash attributes across New Jersey counties. After extracting meaningful results from the training model's attributes, visualization tools will be used to portray the results that would be easy to understand for the end user.

Evaluation Metrics

This project will explore the results from SageMaker and SciKit-Learn PCA methods and K-means algorithms. In the dimension reduction process, the top principle components each algorithm chooses can be compared and the explained variance of each algorithm can be calculated.

To evaluate resultant clusters from each K-means algorithm, visualization methods such as heatmaps, cluster graphs, and tables will be used to compare cluster labels. A summary of differences and similarities will be provided, as well as a discussion on crash patterns that were discovered.

Analysis

Data Exploration

The Crash table provided by NJDOT consists of rows signifying an individual accident and columns as crash attributes. Each crash has a unique identifier and two location columns consisting of a county and a municipality. Crash attribute columns signify the characteristics of a crash. These include alcohol involvement, weather conditions, road surface conditions, street names, and highest injury severity. The data types of these columns range from word descriptions, Yes/No values, and numerical codes. Table 1 shows sample rows in the raw Accidents table with the various data types found in the columns.

Table 1. Sample rows from the raw Accidents table

Crash ID	county	muni	severity_rating	alcohol_involved	xstreet
12-20-2018	12	20	1	N	BERGEN
03-04-2018	3	4	4	Y	NJ 4

While reviewing the raw data, a few design choices were considered, and problematic areas were identified. For design choice, I wanted to focus on accidents that contained a fatal or serious injury. Because many stakeholders who use this data for safety programs mainly analyze crashes with fatalities or serious injuries, I wanted to narrow my focus to these two types of accidents. Therefore, my dataset will consist of fatal or serious injury (SI) related crashes.

Below is a list of problematic areas which will be solved in the data pre-processing stage. These issues would need to be resolved so that the data is suitable for PCA and for training a K-means model.

1. Many columns do not pertain to crash attributes that can easily be clustered together. These include columns that are descriptions, file meta-data, or vehicle/persons counts. These columns will need to be removed to isolate crash attribute columns.
2. County and Municipality values are integers instead of readable titles. These integer values would need to be transcribed and concatenated so that rows can be grouped by County-Municipality in a future pre-processing step.
3. Some crash attribute columns are retired in (2017-2018) compared to (2014-2016). Values for these retired columns will need to be queried from the other available tables from the NJDOT site.
4. Each row signifies a crash accident which means there are multiple data points per County-Municipality. Additionally, crash attribute columns are categorical which is unfit for PCA models. Since this project focuses on finding crash patterns among individual municipalities, the data will need to be encoded numerically and grouped by municipality.

Exploratory Visualization

The tables and plots below are Crash data with a severity rating of Fatal or Serious Injury only. I wanted to explore the ratio between fatal and serious injury crashes. I also wanted to discover the municipalities that had the highest frequency of crashes and see if these municipalities would appear in any cluster from the K-means model.

Table 2 shows the number of Fatal and Serious Injury crashes in the 2014-2018 Accidents table. There are about 6,800 data points in this set with 36% being Fatal crashes.

Table 2. Total crash count of 2014-2018 Accident dataset

	Crash Count	
Serious Injury	4340	63.6%
Fatality	2479	36.4%
Total Crashes	6819	100%

Figure 1 shows the top 10 municipalities which have the most fatal and serious injury crashes. With more datapoints under each of these municipalities, I believe that these municipalities are areas with high-risk or higher traffic volumes. In the Model Evaluation section, it would be interesting to see if any of these municipalities fall into similar clusters.

Figure 2 shows a histogram of crash count distributions by municipalities. Out of 511 municipalities in the data, 238 municipalities have one to five fatal or SI crashes. Reasons for small amount of data points may be that fatal or SI crashes are less frequent or that the municipalities in this bucket are smaller and have do not have much heavy traffic compared to larger municipalities.

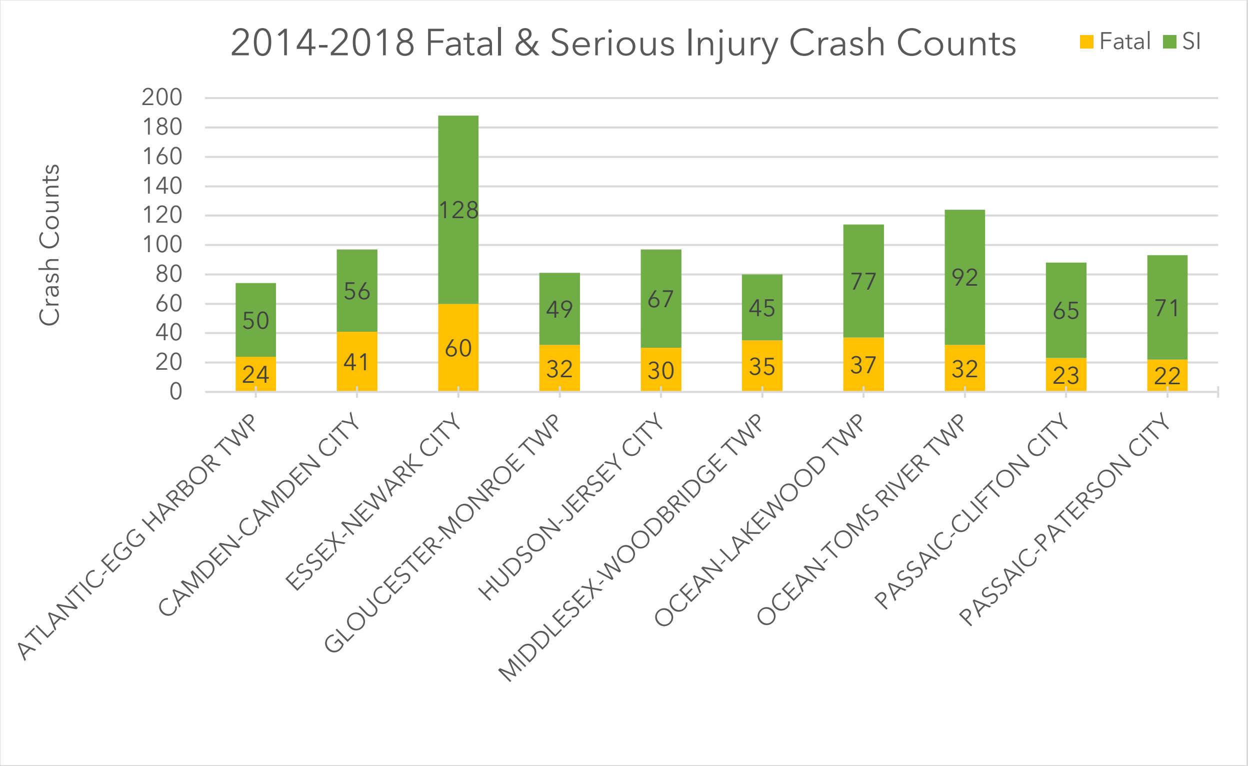


Figure 1. Top 10 municipalities with 2014-2018 Fatal & SI crashes

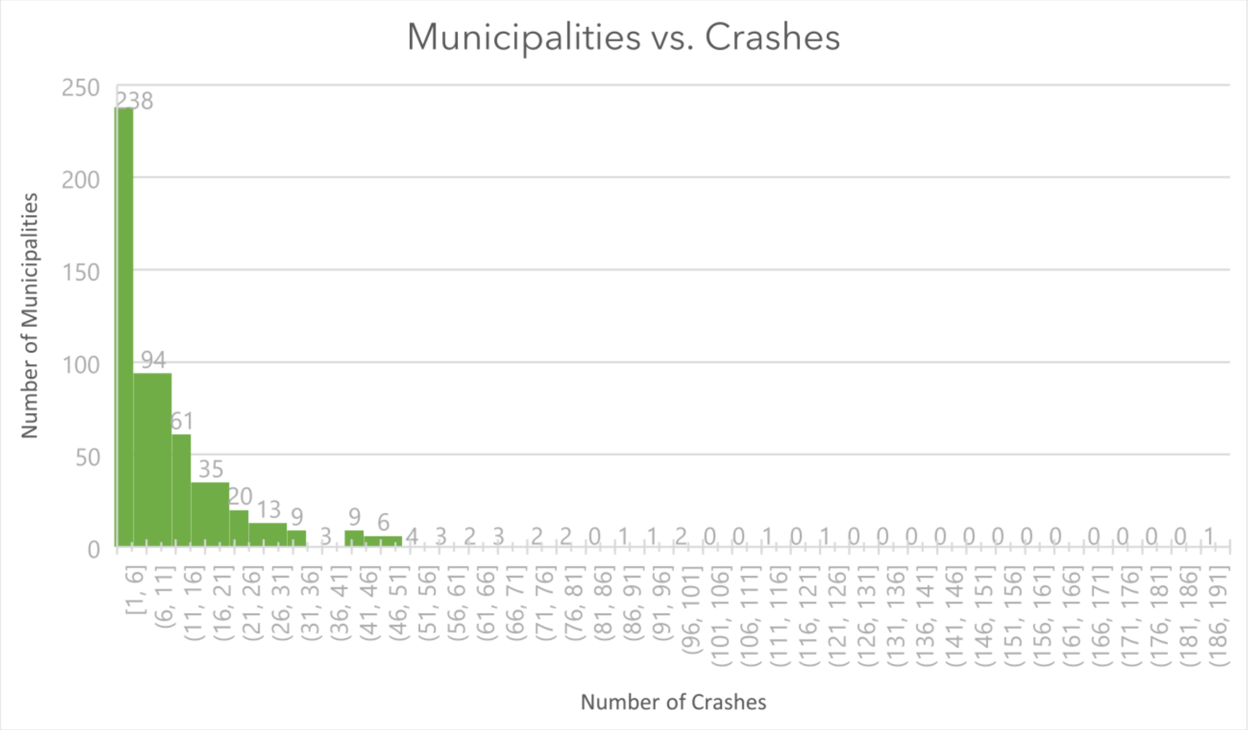


Figure 2. Histogram of crashes by municipality

Algorithms & Techniques

To identify crash patterns across multiple municipalities, a K-means algorithm was trained. K-means algorithms are ideal for identifying similarities or differences within groups; in this case, the algorithm will group municipalities based on similar crash attributes.

To prepare the data for the K-means model, the data must first be numerically normalized. The data was normalized between 0 and 1 using SciKit-Learn's "MinMaxScalar". This normalization step is important because K-means calculates Euclidian distances between data points when determining clusters. If the data is not normalized, features that have larger numerical values due to quantity would be given an unfair weight against all other features. Normalization solves this issue by giving a consistent range between all features.

After normalization, the data was dimensionally reduced using the Principle Component Analysis (PCA) technique. Because the encoded data has 174 columns, it is difficult for a K-means model to determine the most important features. PCA tries to reduce the number of similar or redundant features while maintaining the highest weighted features. The result is a smaller feature set which will help reduce potentially noisy clusters.

Benchmark

The benchmark models for this project will consist of models from Sagemaker and SciKit-Learn. PCA methods from each library will be used to compare dimension reduction results. After deciding on which transformed dataset to use, K-Means algorithms from each library will be used to explore how each algorithm clusters the transformed data. Afterwards, parameters for the SciKit-Learn model will be refined to determine differences between default K-means parameters verses choosing different parameters.

Methodology

Data Preprocessing

The Crash data was pre-processed using the following steps. Code for steps 1-4 are in the Juypter notebook "Data Processing and Exploration". Step 5 is documented in both "PCA KMeans" notebooks.

1) Assess retired or added columns

Crash data from 2017 to 2018 introduced new columns which were not present for the 2014-2016 dataset. For 2014-2016 data, the column "first_harmful_event" which describes the first vehicle's event, did not have values. To extract values for these years, the Vehicles table from NJDOT was used to query the crash's unique ID and the "first_harmful_event" value. These values were joined to the raw dataset associated by the crash's ID using SQL.

2) Remove non-crash attribute columns

Columns that were not crash attribute related were removed in Excel. Removed columns include file meta-data (UPDATE_DATE, DLN), text based descriptions (LOCATION, XSTREET_NAME), counts (NO_KILLED, PED_KILLED), and columns which contained incomplete data (LATITUDE, AGENCY_ORI).

Out of 76 columns, 21 columns were kept as shown in Table 3. The titles and code definitions can be found in the NJTR-1 forms.

Table 3. Crash attribute columns in the dataset

crashid (unique ID)	road_grade_code
MUN_CTY_CO (county)	road_surf_code
MUN_MU (municipality)	surf_cond_code
severity_rating5	light_cond_code
intersection	environ_cond_code
alcohol_involved	road_median_code
hazmat_involved	temp_traffic_zone
hs_collision_type	posted_speed
crash_type	first_harmful_event
road_sys_code	flg_cell_in_use
road_horiz_align_code	

3) Transcribe county and municipality codes into readable titles

To create readable county-municipality titles, the county and municipality codes were transcribed based on the New Jersey County and Municipality Codes document. All code combinations were transcribed in Excel using the VLOOKUP function. A "location" column was added to hold titles while the "MUN_CTY_CO (county)" and "MUN_MU (municipality)" were removed from the dataset.

4) Encode categorical columns and create municipality groups

Because training a K-means model requires numerical data, all categorical data in the Crash dataset must be encoded and normalized. All columns except "crashed" and "location" were encoded using One Hot Encoding[1]. Because each column contains multiple codes or values, this encoding was used to "flatten" the dataset, creating 174 encoded columns. Table 4 provides an example of encoded columns.

Table 4. Sample of encoded columns

Alcohol_involved_Y	Alcohol_involved_N	Crash_type_1	Crash_type_2	Crash_type_3
1	0	0	1	0

After encoding, the "crashid" and "location" columns were added back. The encoded values were verified on a "crashid" basis by spot checking the raw data. Once verified, the "crashid" column was removed and the "location" column was designated as the ID column for the data frame.

As a pre-processing step before normalizing the data, locations were grouped and each crash attribute column was summed (Table 5).

[1] Shoutout to my Proposal reviewer for suggesting this idea to me!

Table 5. Sample of grouped locations

location	alcohol_ involved_N	alcohol_ involved_Y	crash_ type_1	crash_ type_2	crash_ type_3
ATLANTIC-EGG HARBOR TWP	53	21	6	1	11
BERGEN-HACKENSACK CITY	31	1	6	1	5
MONMOUTH-NEPTUNE TWP	30	4	3	0	1

5) Normalize numeric columns

To prepare for PCA, all crash attribute columns were normalized using SciKit-Learn's "MinMaxScalar". This function transforms all numerical values into a range between 0 and 1, such that comparing the values of different features is more consistent. The resulting dataset is then used for PCA.

Implementation

The implementation process for both SageMaker and SciKit-Learn libraries are as follows. Code for this process are in both "PCA KMeans" Juypter notebooks.

1) Dimensionality reduction using PCA

The normalized crash dataset with 174 encoded columns was used to train a PCA model from SageMaker and SciKit-Learn libraries. After both models were trained, their principle component makeup and variance were analyzed.

2) Feature engineering and data transformation

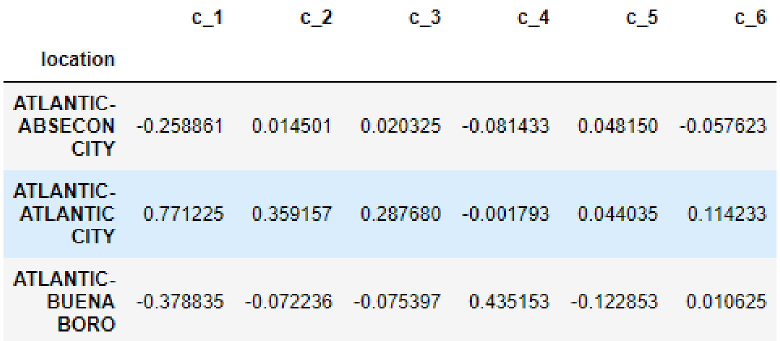
In this project, a 70% data variance was used as a benchmark to select "n" top principle components. SageMaker and SciKit-Learn use slightly different means of calculating explained variance. Because SciKit-Learn returns explained variance as a percentage, variance can be simply summed for n components until 70% is reached. Since, SageMaker returns variance the following formula was used to determine "n" principle components for 70% data variance:

$$\frac{\sum_n s_n^2}{\sum s^2}$$

where "s" is the approximate data variance covered by the first "n" principle components. The resulting number of principle components for both models was 25 with a variance of 70.95%.

After identifying "n", the top 25 components were extracted into a new data frame with rows representing a municipality and columns representing a component (Figure 3). Values represent the weight of the component for a municipality.

A component's makeup can be displayed using the "display_component" function which outputs a histogram of the features that are in a component and their weights. To reuse this function for the SciKit-Learn model, the "components_" data frame outputted by the PCA model had to be transposed such that features are represented as rows and components are represented as columns. Histograms of these 25 components from SageMaker and SciKit-Learn are in Appendix A.



	c_1	c_2	c_3	c_4	c_5	c_6
location						
ATLANTIC-ABSECON CITY	-0.258861	0.014501	0.020325	-0.081433	0.048150	-0.057623
ATLANTIC-ATLANTIC CITY	0.771225	0.359157	0.287680	-0.001793	0.044035	0.114233
ATLANTIC-BUENA BORO	-0.378835	-0.072236	-0.075397	0.435153	-0.122853	0.010625

Figure 3. Sample of transformed data set by SciKit-Learn PCA model

3) Clustering transformed data using K-means

After transforming the Crash data set into a data frame consisting of 25 principle components, this data will be used to train a K-means model to segment municipalities using their PCA attributes. The "k" clusters chosen for both SageMaker and SciKit-Learn models was 8 based on empirical data. After the K-means models were trained using default parameters, model attributes were extracted for visualization analysis.

4) Extracting trained model attributes and visualizing k clusters

Model attributes such as cluster makeup and centroids were extracted for visualization analysis. Plots including histograms, heatmaps, and tables were generated to understand groupings the K-means models created. These plots are shown in the Results section.

Refinement

The SciKit-Learn K-means model was used for refining model parameters and providing a comparison between models that used default parameters. Based on an initial cluster analysis, there were clusters with one or two municipalities which would not provide meaningful results. To create more generalized clusters, the following parameters were changed:

- Data variance was decreased to 60% for a new value of 15 top "n" principle components
- "n_init" parameter was increased to 300 from 100
- "max_iter" parameter was increased to 500 from 300 (default)
- "k" cluster value was changed to 5

The new PCA components from this refinement remained the same compared to PCA components from SageMaker and SciKit-Learn. Cluster makeup for the refined model slightly differed between the SageMaker and SciKit-Learn models. This is discussed in the Results section.

Results

Model Evaluation & Validation

The following sections evaluate the PCA and K-means models between SageMaker, SciKit-Learn, and the refined SciKit-Learn model.

1) PCA Evaluation

PCA components between SageMaker, SciKit-Learn, and the refined SciKit-Learn model were all the same. The only difference between the SageMaker model and both SciKit-Learn models were that some weights were negative in the SciKit-Learn models. The numerical weights, however, were the same. Figure 4 shows a PCA comparison where SciKit-Learn components are flipped. All PCA component charts are found in Appendix A.

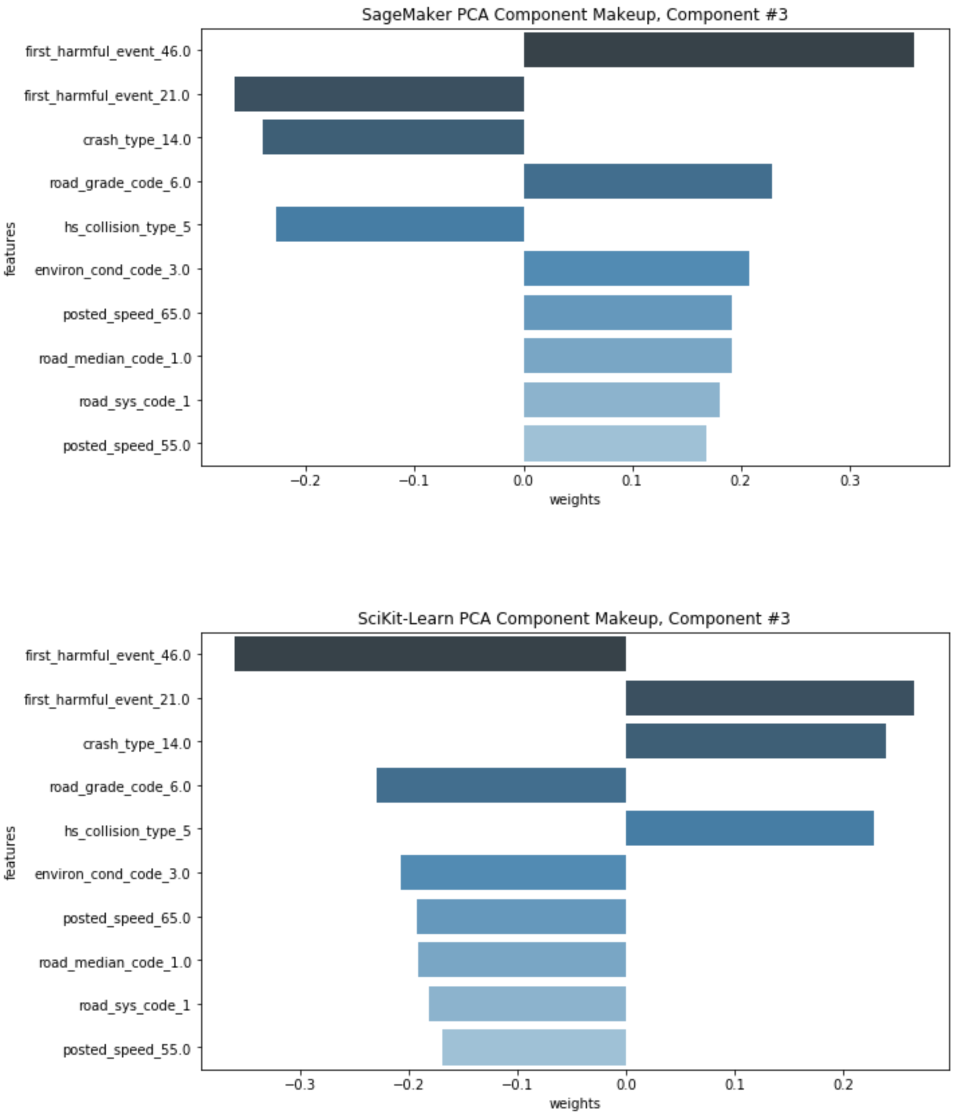


Figure 4. PCA component comparison: SageMaker (left), SciKit-Learn (right)

2) Cluster Makeup Evaluation

Clusters generated by SageMaker and both SciKit-Learn models differed in municipality makeup for each (Figure 5 and Excel in Additional Materials folder). It is important to note that cluster labels that a model assigns are primarily random; therefore, cluster labels for one model are independent of the other models. For example, SageMaker's cluster 5 is more similar to cluster 0 from both SciKit-Learn models but was assigned to 5. I chose not to group similar clusters together in this graph because the disparity becomes larger when looking at clusters with a smaller makeup.

The "k" chosen for the refined SciKit-Learn model was 5 and does not have cluster 6 nor 7 data.

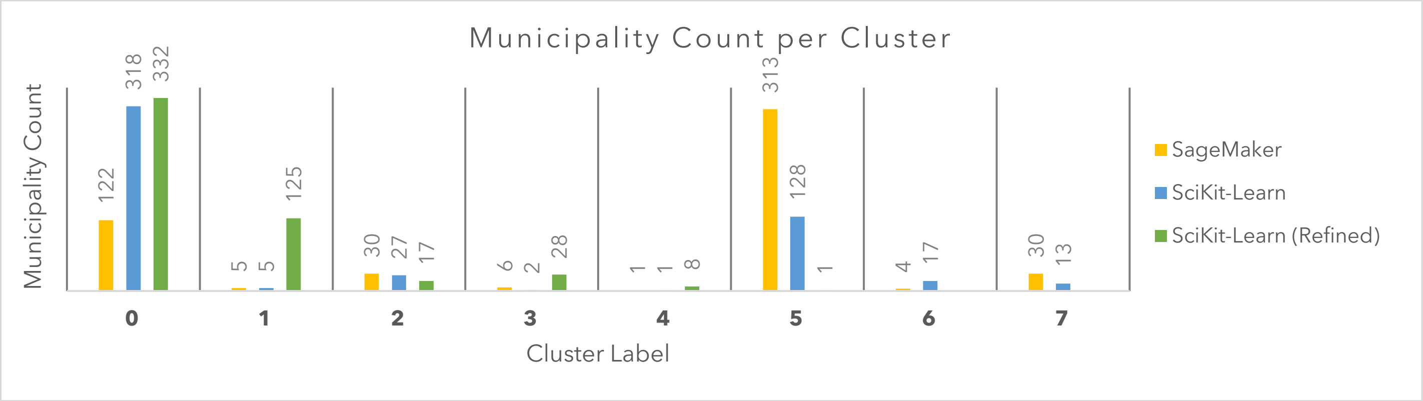


Figure 5. Cluster makeup by model. Note that cluster labels for one model are independent of the other models.

When analyzing the makeup of each cluster, my goal was to identify differences between each cluster and to identify which clusters contained the top 10 municipalities with the most fatal and SI crashes (Figure 1). The following process was used to compare cluster makeups:

1) Identify clusters with high PCA component values by analyzing heatmaps of cluster centroids and their location in PCA feature space.
2) For clusters with high attribute values, their models will be compared. Clusters with less than two municipalities or clusters with at least 100 municipalities will not be compared unless their attribute makeups are high. Clusters with less than two municipalities would not provide meaningful insights as there are not enough municipalities within the cluster for comparison.
3) As a comparison benchmark, SageMaker clusters will be used as a makeup baseline while both SciKit-Learning models will be compared for similarities or differences.

Based on heatmaps of PCA attribute values by cluster centroids for each model (FiguresFigure *6*, Figure *7*, and Figure *8*) clusters with large numbers of municipalities did not have any attributes with high values. Clusters 0 and 5 in SageMaker and SciKit-Learn did not have significant weights for any PCA component. This suggests that a majority of municipalities do not have similar crash attributes. One explanation is that these clusters could encapsulate most of the municipalities that have few crash data points. Crashes are "random" in nature such that there are a multitude of crash attributes that are recorded. For municipalities with few data points, there may not be many locations with similar crash attributes.

Crashes with less than two municipalities, such as Cluster 3 and 4 (Sagemaker and SciKit-Learn), were ignored as there are too few locations to identify similar crash attributes with.

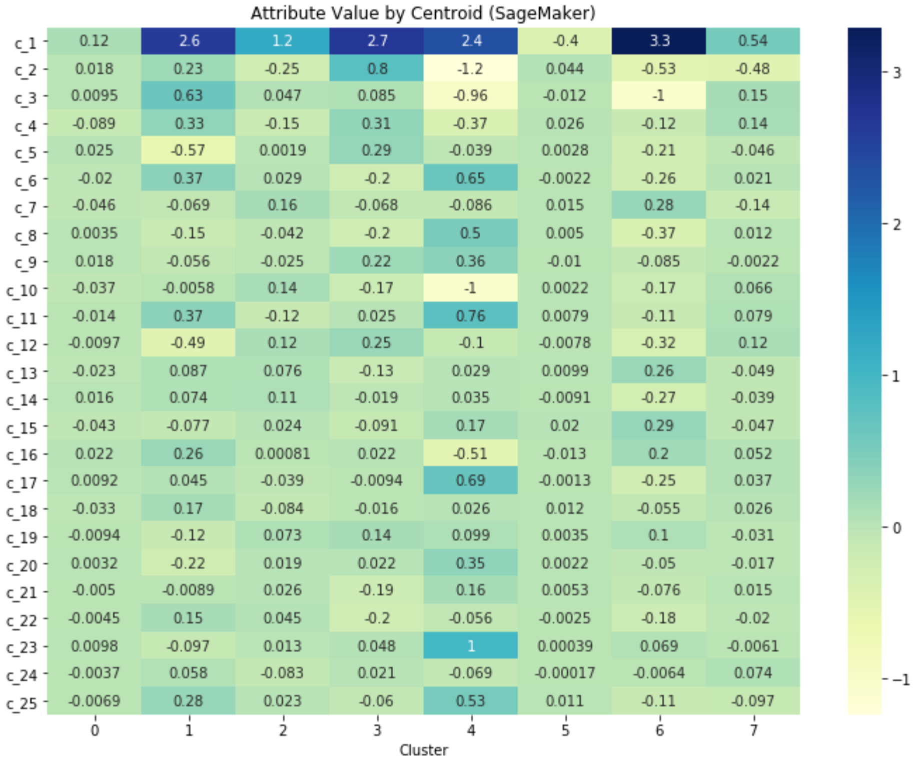


Figure 6. SageMaker PCA attribute values by cluster centroid

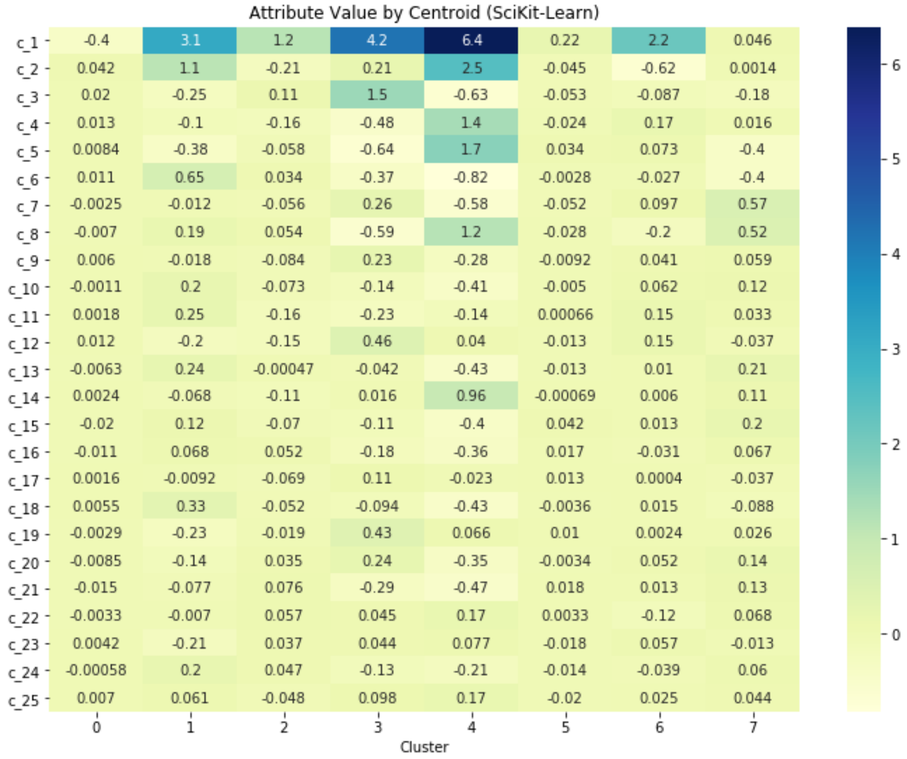


Figure 7. SciKit-Learn PCA attribute values by cluster centroid

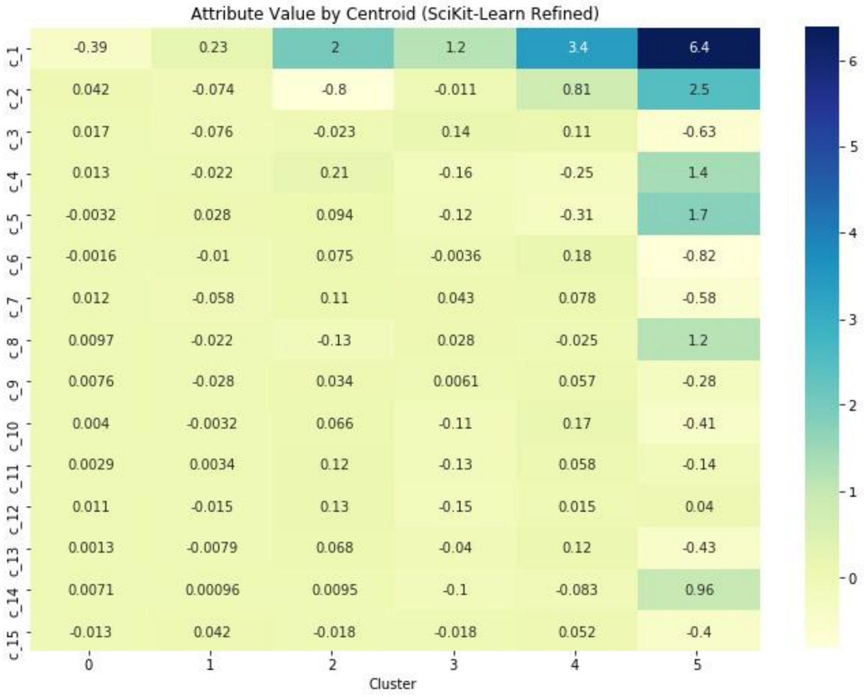


Figure 8. SciKit-Learn (Refined) PCA attribute values by cluster centroid

To compare cluster makeup between models, clusters from SageMaker were used as a baseline for comparison with both SciKit-Learn models. Based on municipalities in a Sagemaker cluster, SciKit-Learn clusters were chosen if they contained any of the same municipalities in the Sagemaker cluster. Tables Table *6*Table 7Table 8 show the comparisons between SageMaker's clusters 1, 3 and 6 with bold municipalities as one of the top 10 municipalities with highest fatal and SI crashes, and highlighted cells as municipalities that are present in the SageMaker cluster.

While it is evident that each model created clusters with different municipality makeups, it is interesting to note that many of the top 10 municipalities were clustered together. For example, SageMaker's cluster 6 contains 4 of the 10 municipalities together while the refined SciKit-Learn model contains 7 of the 10 municipalities. This may suggest that these municipalities have similar traits in either crash attributes or traffic logistics such as high traffic volume or street/highway infrastructure.

Another interesting observation is that both SciKit-Learn models isolated Essex-Newark City into its own cluster despite being a top 10 municipality. Looking at the heatmaps for cluster 4 (SciKit-Learn) and 5 (refined), this municipality has more PCA components with significant attribute values compared to other clusters with top 10 municipalities (cluster 4 from the refined model). This suggests that Essex-Newark City differs in the top 10 municipalities in that there are more crash attributes that are associated with this municipality compared to the others.

Comparing the refined SciKit-Learn model to the original, the refined model grouped more of the top 10 municipalities together. This is due to having a lower "k" value where the maximum centroids were 5 from 8. This shows how altering "k" can affect clustering, providing a different narrative for potential results.

Table 6. Cluster makeup comparison for SageMaker' Cluster 1

SageMaker	SciKit-Learn		SciKit-Learn (Refined)	
1	**1**	**6**	**4**	**2**
CAMDEN-CAMDEN CITY	**CAMDEN-CAMDEN CITY**	ATLANTIC-EGG HARBOR TWP	**CAMDEN-CAMDEN CITY**	**ATLANTIC-EGG HARBOR TWP**
CUMBERLAND-VINELAND CITY	**HUDSON-JERSEY CITY**	ATLANTIC-HAMILTON TWP	**HUDSON-JERSEY CITY**	ATLANTIC-HAMILTON TWP
MERCER-HAMILTON TWP	MIDDLESEX-EDISON TWP	BURLINGTON-MOUNT LAUREL TWP	MIDDLESEX-EDISON TWP	BURLINGTON-MOUNT LAUREL TWP
MONMOUTH-MIDDLETOWN TWP	**PASSAIC-CLIFTON CITY**	CAMDEN-CHERRY HILL TWP	**MIDDLESEX-WOODBRIDGE TWP**	CAMDEN-CHERRY HILL TWP
PASSAIC-CLIFTON CITY	**PASSAIC-PATERSON CITY**	CAMDEN-WINSLOW TWP	**OCEAN-LAKEWOOD TWP**	CAMDEN-WINSLOW TWP
		CAPE MAY-MIDDLE TWP	**OCEAN-TOMS RIVER TWP**	CAPE MAY-MIDDLE TWP
		CUMBERLAND-VINELAND CITY	**PASSAIC-CLIFTON CITY**	CAPE MAY-UPPER TWP
		GLOUCESTER-DEPTFORD TWP	**PASSAIC-PATERSON CITY**	CUMBERLAND-MAURICE RIVER TWP
		GLOUCESTER-FRANKLIN TWP		CUMBERLAND-VINELAND CITY
		GLOUCESTER-MONROE TWP		GLOUCESTER-DEPTFORD TWP
		MERCER-HAMILTON TWP		GLOUCESTER-FRANKLIN TWP
		MIDDLESEX-OLD BRIDGE TWP		**GLOUCESTER-MONROE TWP**
		MIDDLESEX-WOODBRIDGE TWP		MERCER-HAMILTON TWP
		MONMOUTH-HOWELL TWP		MIDDLESEX-OLD BRIDGE TWP
		MONMOUTH-MIDDLETOWN TWP		MONMOUTH-HOWELL TWP
		OCEAN-JACKSON TWP		MONMOUTH-MIDDLETOWN TWP
		PASSAIC-WAYNE TWP		OCEAN-JACKSON TWP

Table 7. Cluster makeup comparison for SageMaker's Cluster 3

SageMaker	SciKit-Learn			SciKit-Learn (Refined)		
3	**1**	**6**	**4**	**4**	**2**	**5**
CAMDEN-CHERRY HILL TWP	**CAMDEN-CAMDEN CITY**	**ATLANTIC-EGG HARBOR TWP**	**ESSEX-NEWARK CITY**	**CAMDEN-CAMDEN CITY**	**ATLANTIC-EGG HARBOR TWP**	**ESSEX-NEWARK CITY**
ESSEX-NEWARK CITY	**HUDSON-JERSEY CITY**	ATLANTIC-HAMILTON TWP		**HUDSON-JERSEY CITY**	ATLANTIC-HAMILTON TWP	
HUDSON-JERSEY CITY	MIDDLESEX-EDISON TWP	BURLINGTON-MOUNT LAUREL TWP		MIDDLESEX-EDISON TWP	BURLINGTON-MOUNT LAUREL TWP	
MIDDLESEX-EDISON TWP	**PASSAIC-CLIFTON CITY**	CAMDEN-CHERRY HILL TWP		**MIDDLESEX-WOODBRIDGE TWP**	CAMDEN-CHERRY HILL TWP	
MIDDLESEX-WOODBRIDGE TWP	**PASSAIC-PATERSON CITY**	CAMDEN-WINSLOW TWP		**OCEAN-LAKEWOOD TWP**	CAMDEN-WINSLOW TWP	
PASSAIC-PATERSON CITY		CAPE MAY-MIDDLE TWP		**OCEAN-TOMS RIVER TWP**	CAPE MAY-MIDDLE TWP	
		CUMBERLAND-VINELAND CITY		**PASSAIC-CLIFTON CITY**	CAPE MAY-UPPER TWP	
		GLOUCESTER-DEPTFORD TWP		**PASSAIC-PATERSON CITY**	CUMBERLAND-MAURICE RIVER TWP	
		GLOUCESTER-FRANKLIN TWP			CUMBERLAND-VINELAND CITY	
		GLOUCESTER-MONROE TWP			GLOUCESTER-DEPTFORD TWP	
		MERCER-HAMILTON TWP			GLOUCESTER-FRANKLIN TWP	
		MIDDLESEX-OLD BRIDGE TWP			**GLOUCESTER-MONROE TWP**	
		MIDDLESEX-WOODBRIDGE TWP			MERCER-HAMILTON TWP	
		MONMOUTH-HOWELL TWP			MIDDLESEX-OLD BRIDGE TWP	
		MONMOUTH-MIDDLETOWN TWP			MONMOUTH-HOWELL TWP	
		OCEAN-JACKSON TWP			MONMOUTH-MIDDLETOWN TWP	
		PASSAIC-WAYNE TWP			OCEAN-JACKSON TWP	

Table 8. Cluster makeup comparison for SageMaker's Cluster 6

SageMaker	SciKit-Learn		SciKit-Learn (Refined)	
6	**3**	**6**	**4**	**2**
ATLANTIC-EGG HARBOR TWP	**OCEAN-LAKEWOOD TWP**	**ATLANTIC-EGG HARBOR TWP**	**CAMDEN-CAMDEN CITY**	**ATLANTIC-EGG HARBOR TWP**
GLOUCESTER-MONROE TWP	**OCEAN-TOMS RIVER TWP**	ATLANTIC-HAMILTON TWP	**HUDSON-JERSEY CITY**	ATLANTIC-HAMILTON TWP
OCEAN-LAKEWOOD TWP		BURLINGTON-MOUNT LAUREL TWP	MIDDLESEX-EDISON TWP	BURLINGTON-MOUNT LAUREL TWP
OCEAN-TOMS RIVER TWP		CAMDEN-CHERRY HILL TWP	**MIDDLESEX-WOODBRIDGE TWP**	CAMDEN-CHERRY HILL TWP
		CAMDEN-WINSLOW TWP	**OCEAN-LAKEWOOD TWP**	CAMDEN-WINSLOW TWP
		CAPE MAY-MIDDLE TWP	**OCEAN-TOMS RIVER TWP**	CAPE MAY-MIDDLE TWP
		CUMBERLAND-VINELAND CITY	**PASSAIC-CLIFTON CITY**	CAPE MAY-UPPER TWP
		GLOUCESTER-DEPTFORD TWP	**PASSAIC-PATERSON CITY**	CUMBERLAND-MAURICE RIVER TWP
		GLOUCESTER-FRANKLIN TWP		CUMBERLAND-VINELAND CITY
		GLOUCESTER-MONROE TWP		GLOUCESTER-DEPTFORD TWP
		MERCER-HAMILTON TWP		GLOUCESTER-FRANKLIN TWP
		MIDDLESEX-OLD BRIDGE TWP		**GLOUCESTER-MONROE TWP**
		MIDDLESEX-WOODBRIDGE TWP		MERCER-HAMILTON TWP
		MONMOUTH-HOWELL TWP		MIDDLESEX-OLD BRIDGE TWP
		MONMOUTH-MIDDLETOWN TWP		MONMOUTH-HOWELL TWP
		OCEAN-JACKSON TWP		MONMOUTH-MIDDLETOWN TWP
		PASSAIC-WAYNE TWP		OCEAN-JACKSON TWP

4) PCA Components and Cluster Evaluation

To identify crash patterns among municipalities, attributes with the highest values were inspected with its corresponding cluster (Figures Figure *6*, Figure *7*, and Figure *8*). Across SageMaker and SciKit-Learn models, components 1, 2, and 3 had the high values which suggests municipalities that fall under this component have crashes that relate to the attributes within this component (Figure 9).

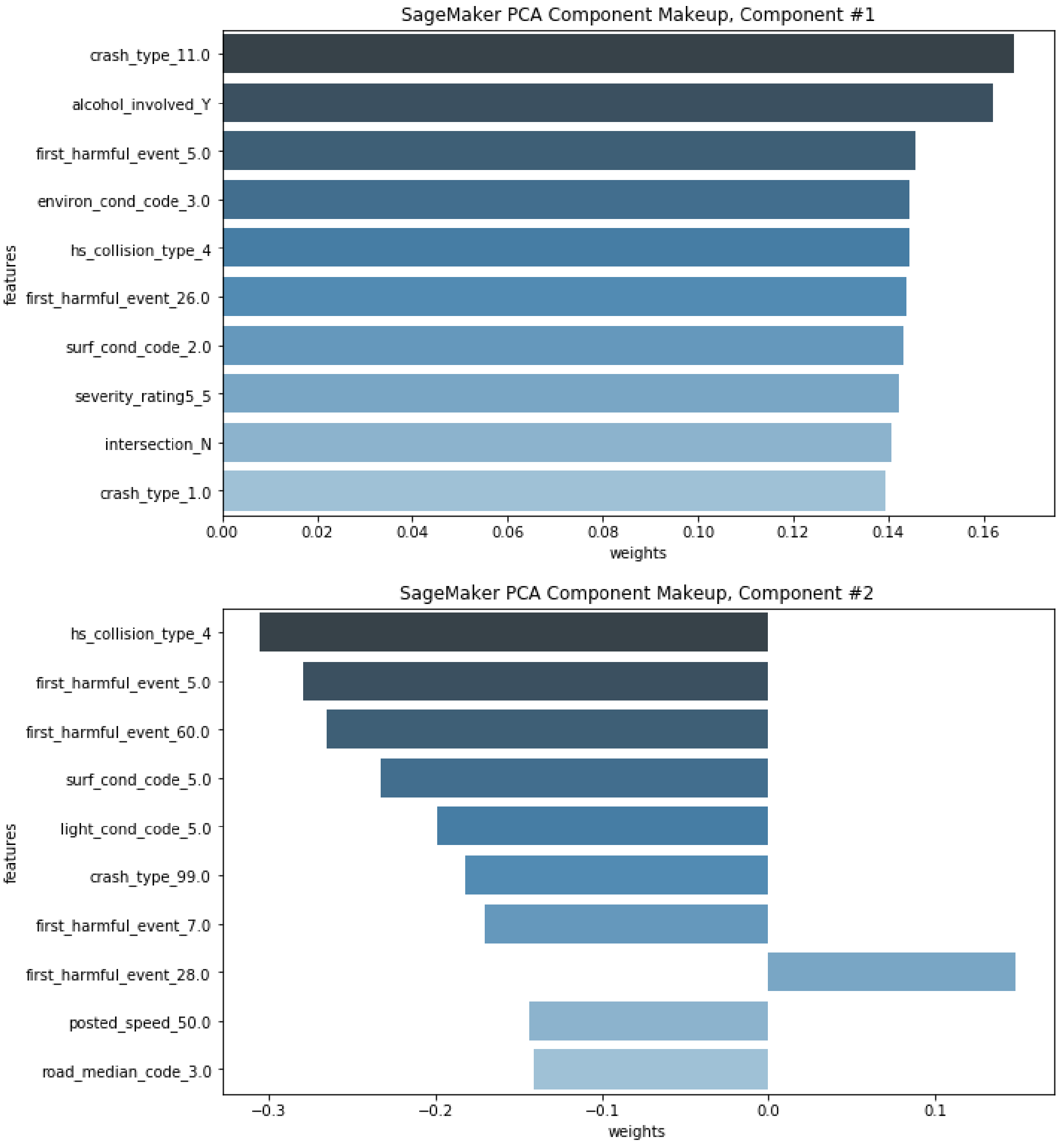


Figure 9. Component makeup for 1 (left) and 2 (right)

For example, municipalities under SageMaker's clusters 1, 2, 3, and 6 have high attribute values for component 1. Similarly, clusters 1, 2, and 6 in the SciKit-Learn model and clusters 2, 3, and 4 in the refined model also have high values for component 1.

While analyzing the crash attributes under component 1 (Table 9), two interesting features are present: "alcohol_involved_Y" (alcohol was involved in a crash) and "severity_rating_5_5" (a fatality occurred in a crash). Fatalities due to alcohol involvement are major statistics for stakeholders when creating safety policies to combat drunk driving. Additional features provide more context of road conditions under this component: crashes tend be hitting a fixed object or a rear end accident in which a vehicle ran off the road or hit another vehicle. These crashes tend to occur in rain or snowy conditions.

Component 2 (Table 9) suggests crashes with vehicles head-on or "other" in a slush road condition with dark (no streetlight) light conditions over a grassy median. The first event of a vehicle includes running off-road, hitting a tree, crossing the median, or hitting a parked vehicle. The posted speed of these crashes tends to be 50 mph.

Table 9. Code translations for components 1 and 2

Component 1	
crash_type_11	Fixed Object
alcohol_involved_y	Alcohol was used
first_harmful_event_5	Ran off-road (right)
enviorn_cond_code_3	Snow
hs_collision_type_4	Head-on/Angular
first_harmful_event_26	Vehicle in transport
surf_cond_code_2	Rain
severity_rating_5_5	Fatality
intersection_n	Not at an intersection
crash_type_1	Rear end

Component 2	
hs_collision_type_4	Head-on/Angular
first_harmful_event_5	Ran off-road (right)
first_harmful_event_60	Tree
surf_cond_code_5	Slush
light_cond_code_5	Dark (no streetlights)
crash_type_99	Other (not listed)
first_harmful_event_7	Crossed median
first_harmful_event_28	Parked vehicle
posted_speed_50	50 mph limit
road_median_code_3	Grass Median

Additional components and their features can be transcribed using the component makeup charts and codes from the NJTR-1 forms. Clusters relating to each component can be identified using the accompanying Excel in the Additional Materials folder.

Justification

PCA components using PCA models from SageMaker and SciKit-Learn were all similar in component makeup and weights. The only difference was that some features in SciKit-Learn components had opposite weights compared to the SageMaker model.

Clusters between SageMaker and SciKit-Learn K-means models differed which influenced the attribute weights by cluster centroids. The refined SciKit-Learn model reduced the number of clusters from 8 to 5, increased the number of iterations, and decreased the data variance from 70% to 60%. The resulting clusters from this refined model contained more municipalities that encapsulated the top 10 locations with the most fatal and serious injury crashes. Comparing the SageMaker model to the SciKit-Learn models, the SageMaker model had more of the top 10 municipalities in clusters than the unrefined SciKit-Learn model.

Conclusion

Reflection

The goal of this project was to identify crash patterns among municipalities in New Jersey using K-means models from SageMaker and SciKit-Learn libraries. This project was broken down to the following steps:

1) Clean Crash data by removing non-crash attribute columns
2) Encoding all categorical columns and normalize columns
3) Dimensionally reduce features by using Principle Component Analysis from SageMaker and SciKit-Learn libraries
4) Train a K-means model from SageMaker and Scikit-Learn libraries using the transformed data from step 4
5) Extract K-means attributes and use visualizations to understand crash patterns

The most challenging aspect in this project was understanding the outputs from PCA and K-means such that I could extract potential crash patterns and explain them in laymen terms. Organization of cluster makeups and relating them to their specific components was difficult because there was a lot of data to keep track of such as cluster comparisons and feature transcribing. The visualizations created such as the component makeup bar charts and attribute value heatmaps helped me interpret the results.

An interesting aspect about this project is how useful the attribute value heatmaps can be as a starting point for engineers to investigate crashes in multiple municipalities. In the example exploring component 1, it was interesting to see how a group of seemingly independent crash attributes related to a cluster of municipalities. An engineer can use this heatmap to analyze municipalities within a cluster, query the attributes described by the component, and investigate problematic areas where safety can be improved.

Using the model artifacts generated by the K-means models, my next goal is to create a data-driven web application to showcase the municipality clusters the model generated using interactive visualization frameworks and geographic information systems.

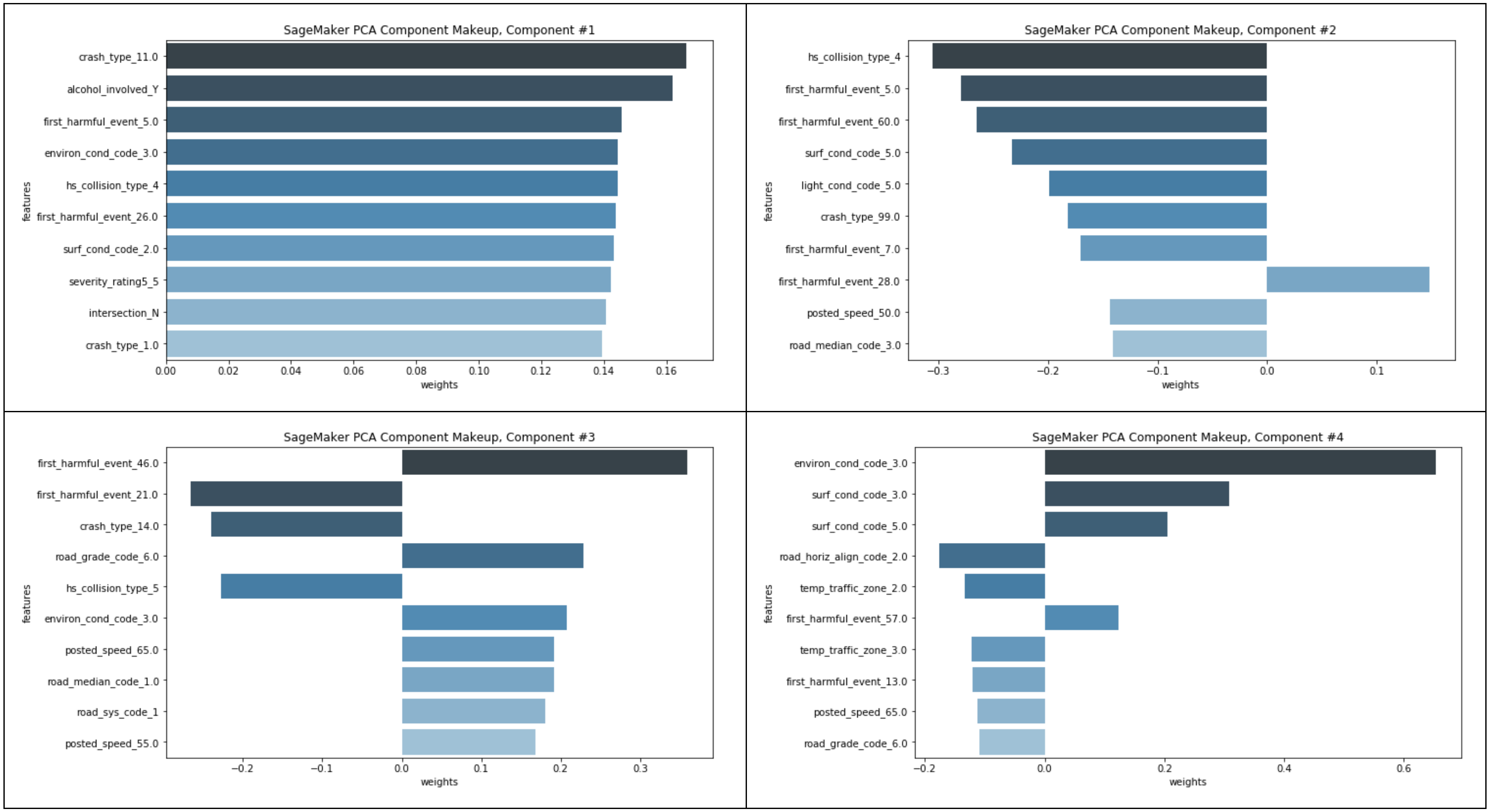
Improvement

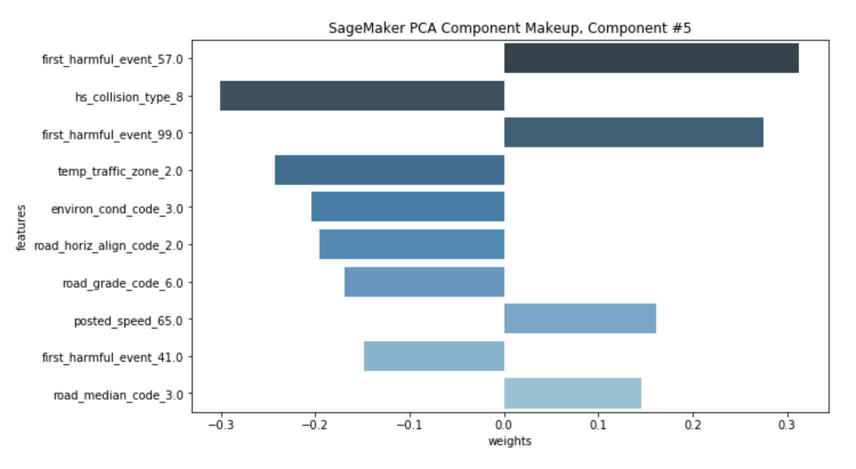
Knowing that the majority of municipalities have few data points, I would exclude those municipalities and focus on those that have a larger amount of data points. I believe this change would provide more meaningful centroid and attribute value groupings. Clusters with a high number of municipalities had no components with large weight values, meaning that the municipalities in the cluster either had no similarities or there was spare data to identify with.

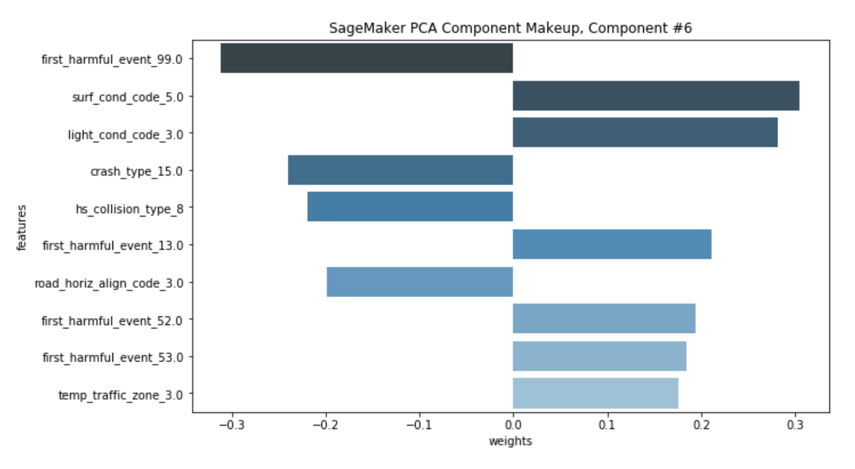
For a more general view of crash patterns, all severity types can be included to have more data for K-means training. This would greatly increase the normalization ranges for all municipalities and provide a more balanced feature set than the fatal/SI data I used.

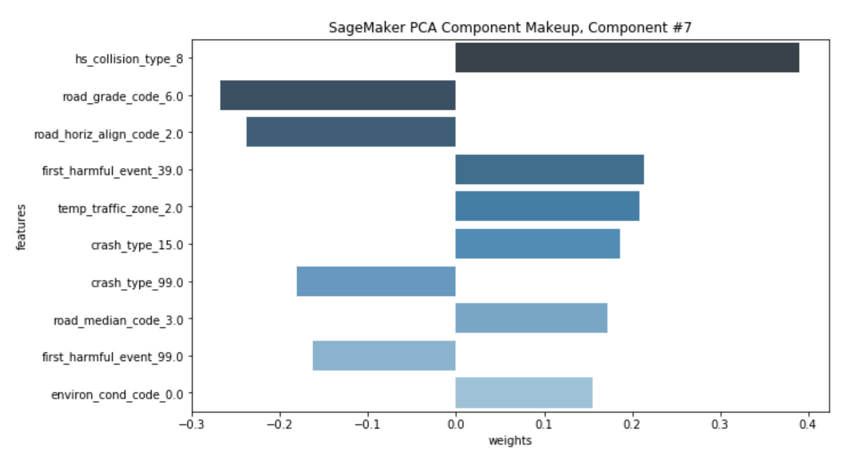
Appendix A: PCA Component Charts

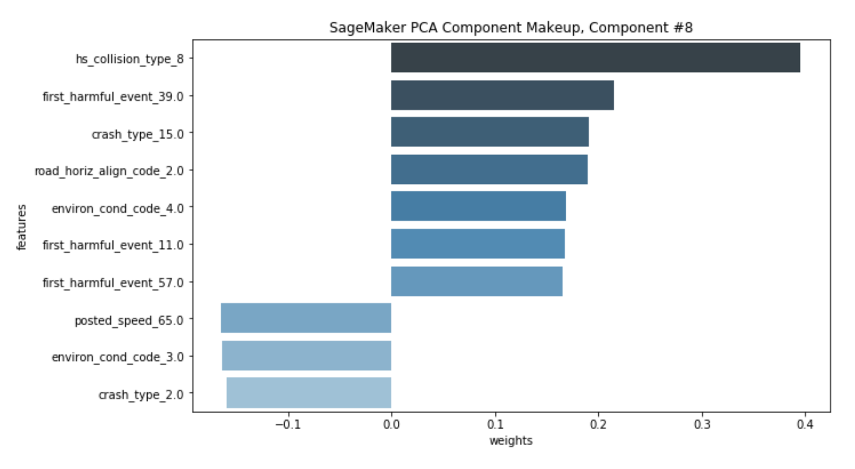
The following charts show the component makeup of the 25 principle components from PCA using SageMaker. Because the component makeup from the SciKit-Learn models are very similar, only the SageMaker components will be shown. The SciKit-Learn components can be found in the Additional Materials folder which is part of this project package.

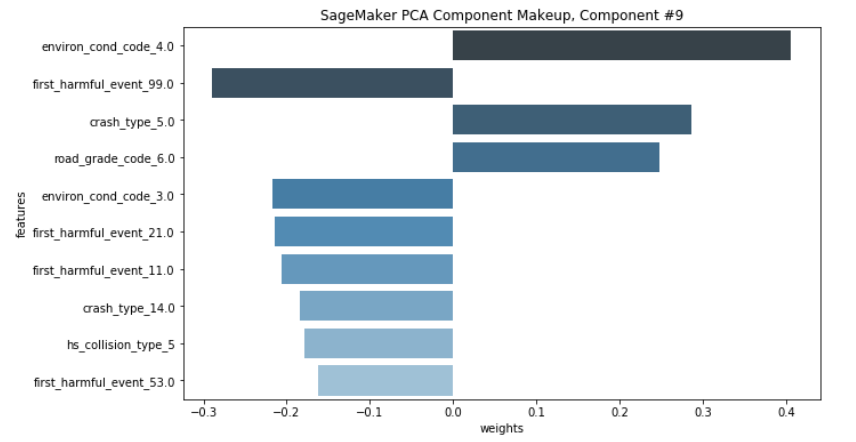


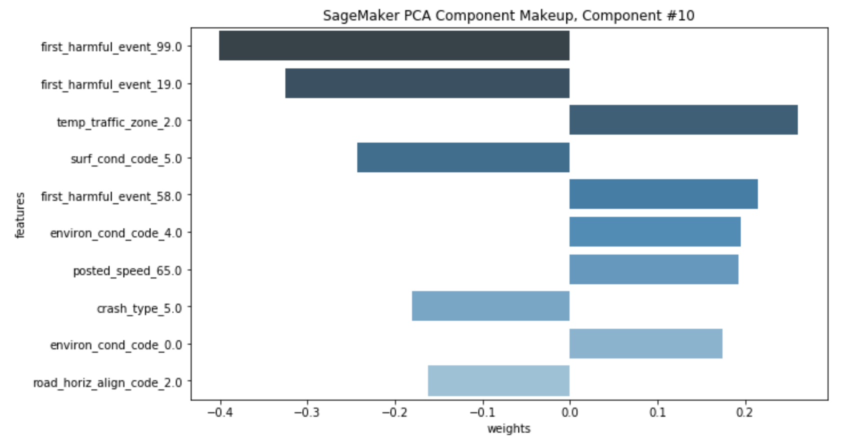

SageMaker PCA Component Makeup, Component #5
features
first_harmful_event_57.0
hs_collision_type_8
first_harmful_event_99.0
temp_traffic_zone_2.0
environ_cond_code_3.0
road_horiz_align_code_2.0
road_grade_code_6.0
posted_speed_65.0
first_harmful_event_41.0
road_median_code_3.0
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
weights

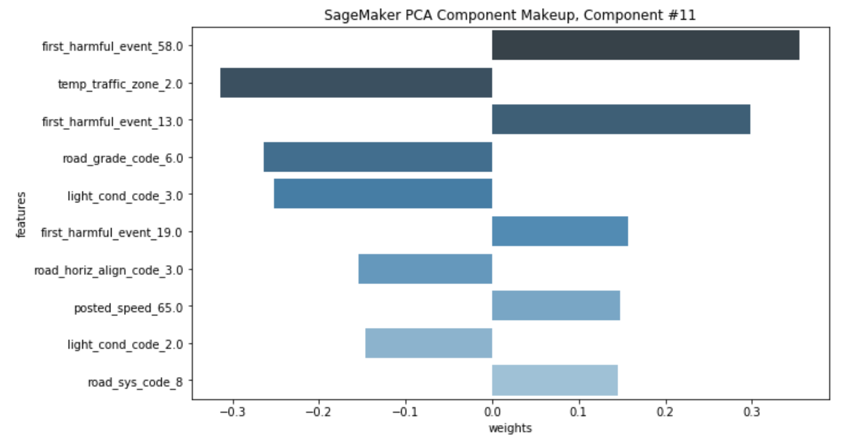

SageMaker PCA Component Makeup, Component #6
features
first_harmful_event_99.0
surf_cond_code_5.0
light_cond_code_3.0
crash_type_15.0
hs_collision_type_8
first_harmful_event_13.0
road_horiz_align_code_3.0
first_harmful_event_52.0
first_harmful_event_53.0
temp_traffic_zone_3.0
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
weights

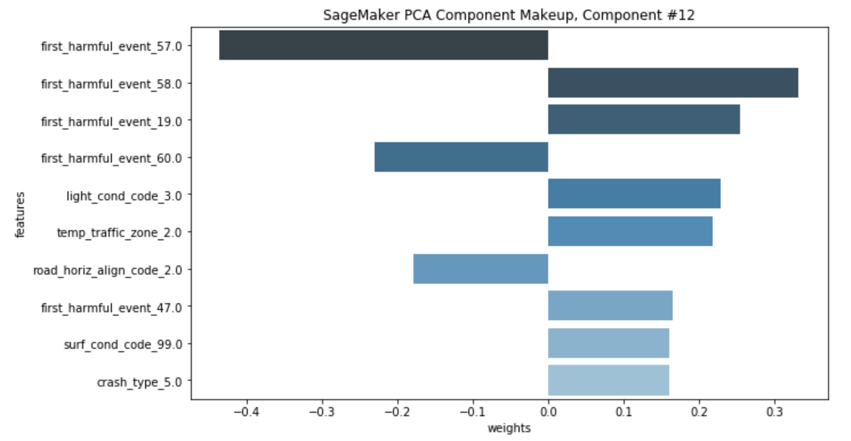

SageMaker PCA Component Makeup, Component #7
features
hs_collision_type_8
road_grade_code_6.0
road_horiz_align_code_2.0
first_harmful_event_39.0
temp_traffic_zone_2.0
crash_type_15.0
crash_type_99.0
road_median_code_3.0
first_harmful_event_99.0
environ_cond_code_0.0
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
weights

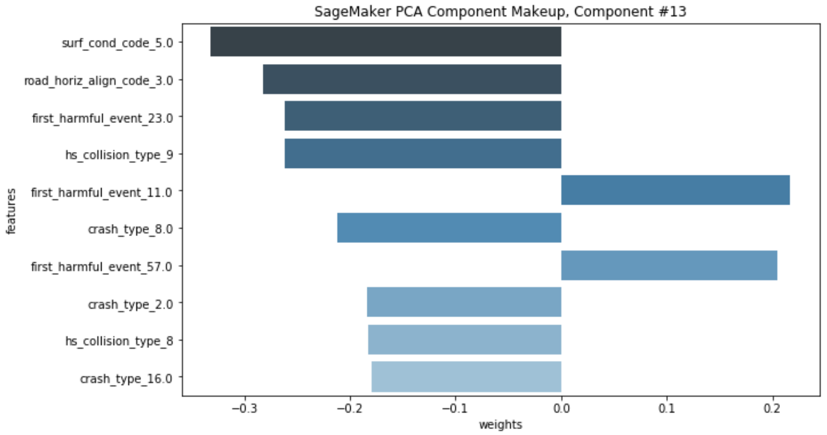

SageMaker PCA Component Makeup, Component #8
features
hs_collision_type_8
first_harmful_event_39.0
crash_type_15.0
road_horiz_align_code_2.0
environ_cond_code_4.0
first_harmful_event_11.0
first_harmful_event_57.0
posted_speed_65.0
environ_cond_code_3.0
crash_type_2.0
−0.1
0.0
0.1
0.2
0.3
0.4
weights

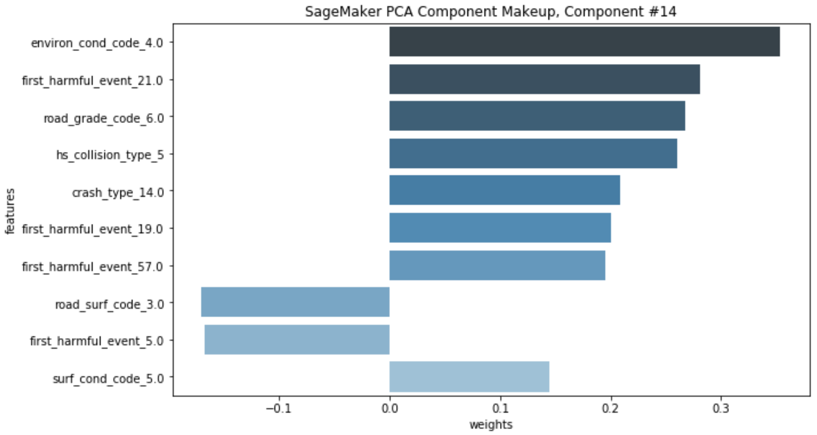

SageMaker PCA Component Makeup, Component #9
environ_cond_code_4.0
first_harmful_event_99.0
crash_type_5.0
road_grade_code_6.0
environ_cond_code_3.0
first_harmful_event_21.0
first_harmful_event_11.0
crash_type_14.0
hs_collision_type_5
first_harmful_event_53.0
features
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
weights

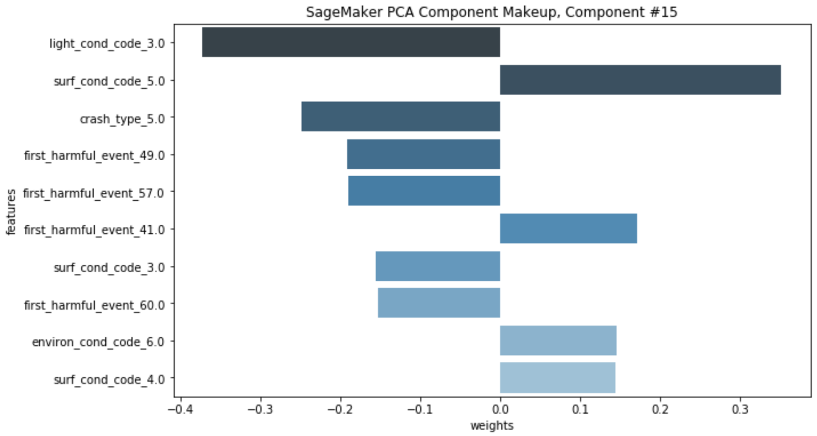

SageMaker PCA Component Makeup, Component #10
first_harmful_event_99.0
first_harmful_event_19.0
temp_traffic_zone_2.0
surf_cond_code_5.0
first_harmful_event_58.0
environ_cond_code_4.0
posted_speed_65.0
crash_type_5.0
environ_cond_code_0.0
road_horiz_align_code_2.0
features
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
weights

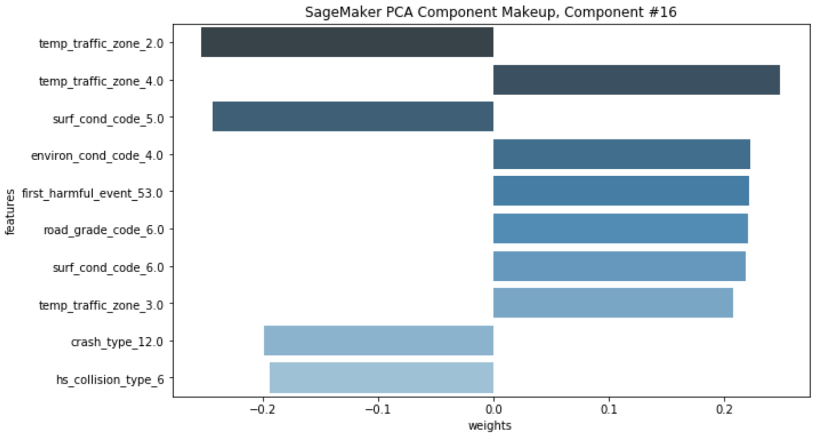

SageMaker PCA Component Makeup, Component #11
first_harmful_event_58.0
temp_traffic_zone_2.0
first_harmful_event_13.0
road_grade_code_6.0
light_cond_code_3.0
first_harmful_event_19.0
road_horiz_align_code_3.0
posted_speed_65.0
light_cond_code_2.0
road_sys_code_8
features
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
weights

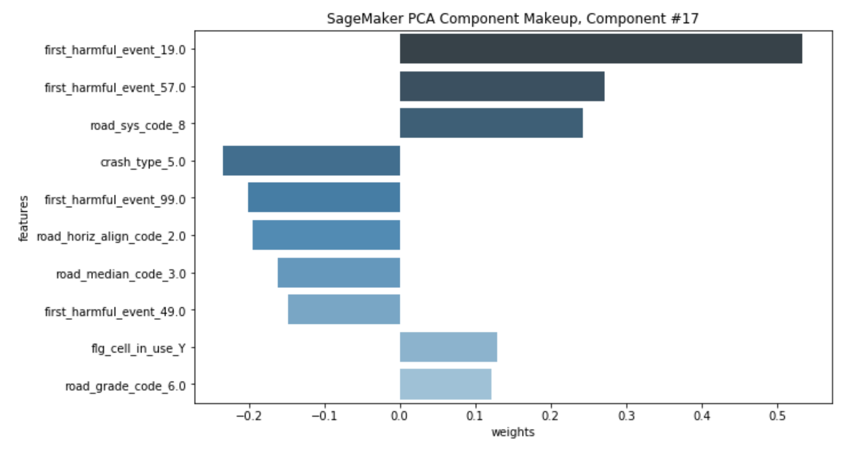

SageMaker PCA Component Makeup, Component #12
first_harmful_event_57.0
first_harmful_event_58.0
first_harmful_event_19.0
first_harmful_event_60.0
light_cond_code_3.0
temp_traffic_zone_2.0
road_horiz_align_code_2.0
first_harmful_event_47.0
surf_cond_code_99.0
crash_type_5.0
features
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
weights

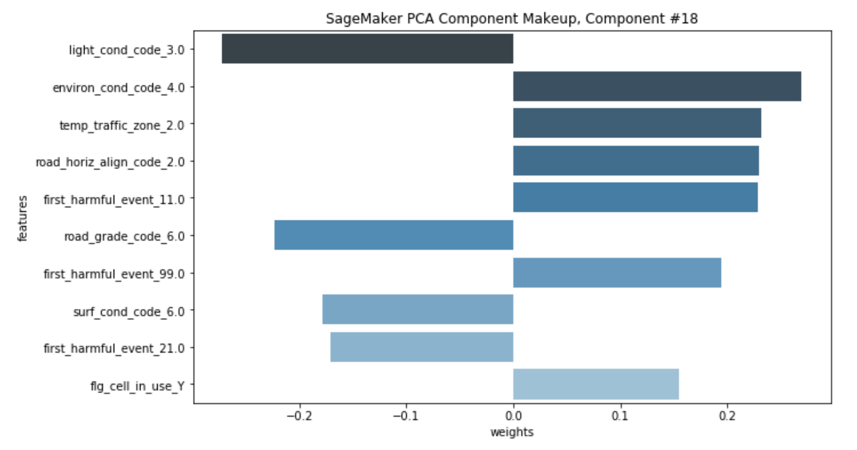

SageMaker PCA Component Makeup, Component #13
features
surf_cond_code_5.0
road_horiz_align_code_3.0
first_harmful_event_23.0
hs_collision_type_9
first_harmful_event_11.0
crash_type_8.0
first_harmful_event_57.0
crash_type_2.0
hs_collision_type_8
crash_type_16.0
−0.3
−0.2
−0.1
0.0
0.1
0.2
weights

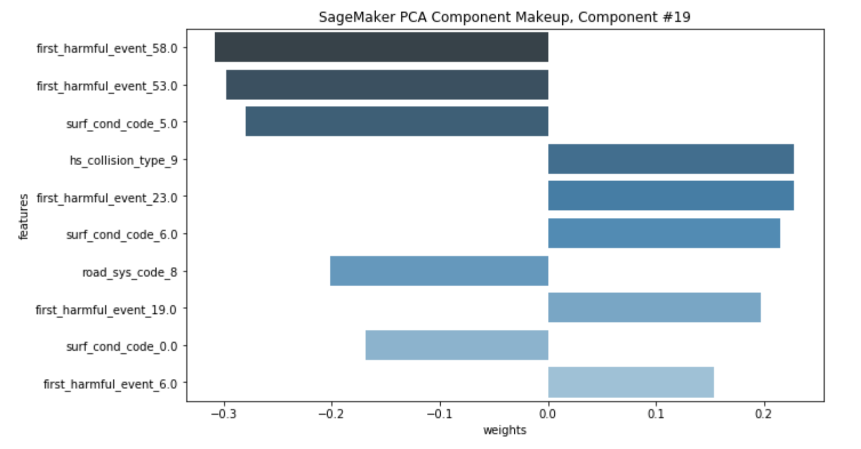

SageMaker PCA Component Makeup, Component #14
features
environ_cond_code_4.0
first_harmful_event_21.0
road_grade_code_6.0
hs_collision_type_5
crash_type_14.0
first_harmful_event_19.0
first_harmful_event_57.0
road_surf_code_3.0
first_harmful_event_5.0
surf_cond_code_5.0
−0.1
0.0
0.1
0.2
0.3
weights

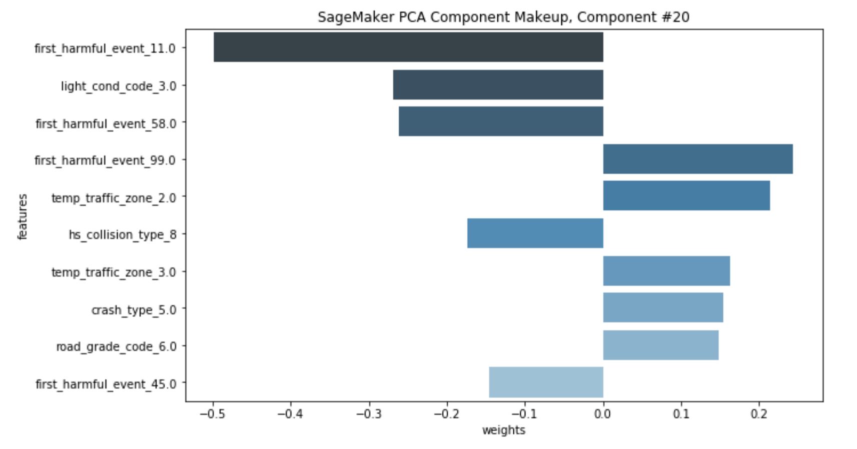

SageMaker PCA Component Makeup, Component #15
features
light_cond_code_3.0
surf_cond_code_5.0
crash_type_5.0
first_harmful_event_49.0
first_harmful_event_57.0
first_harmful_event_41.0
surf_cond_code_3.0
first_harmful_event_60.0
environ_cond_code_6.0
surf_cond_code_4.0
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
weights

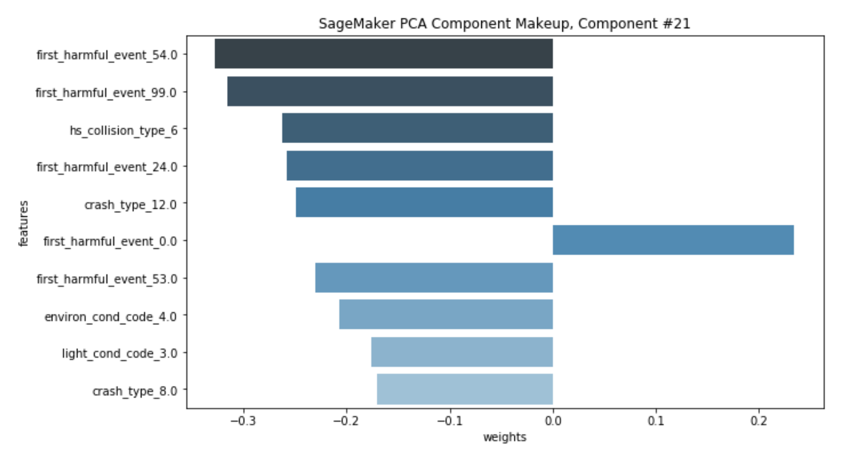

SageMaker PCA Component Makeup, Component #16
features
temp_traffic_zone_2.0
temp_traffic_zone_4.0
surf_cond_code_5.0
environ_cond_code_4.0
first_harmful_event_53.0
road_grade_code_6.0
surf_cond_code_6.0
temp_traffic_zone_3.0
crash_type_12.0
hs_collision_type_6
−0.2
−0.1
0.0
0.1
0.2
weights

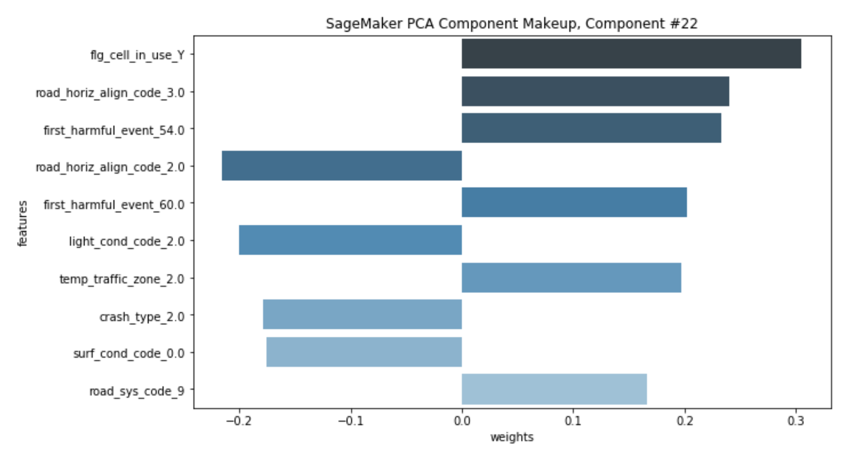

SageMaker PCA Component Makeup, Component #17
features
first_harmful_event_19.0
first_harmful_event_57.0
road_sys_code_8
crash_type_5.0
first_harmful_event_99.0
road_horiz_align_code_2.0
road_median_code_3.0
first_harmful_event_49.0
flg_cell_in_use_Y
road_grade_code_6.0
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
weights

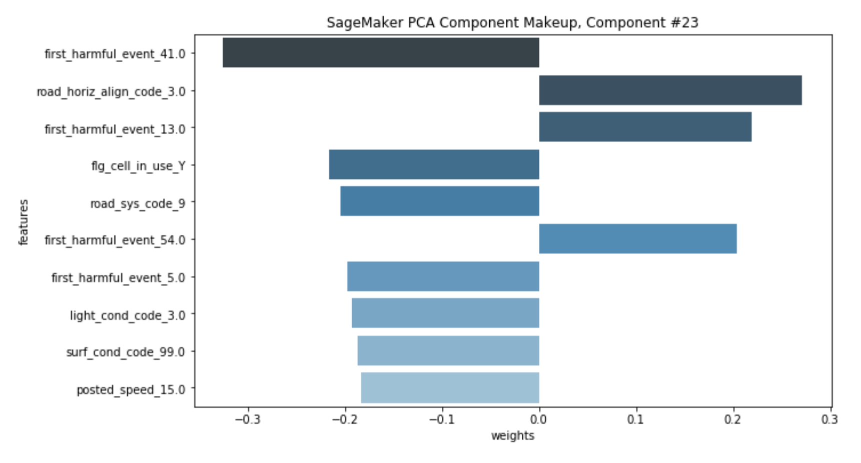

SageMaker PCA Component Makeup, Component #18
features
light_cond_code_3.0
environ_cond_code_4.0
temp_traffic_zone_2.0
road_horiz_align_code_2.0
first_harmful_event_11.0
road_grade_code_6.0
first_harmful_event_99.0
surf_cond_code_6.0
first_harmful_event_21.0
flg_cell_in_use_Y
−0.2
−0.1
0.0
0.1
0.2
weights

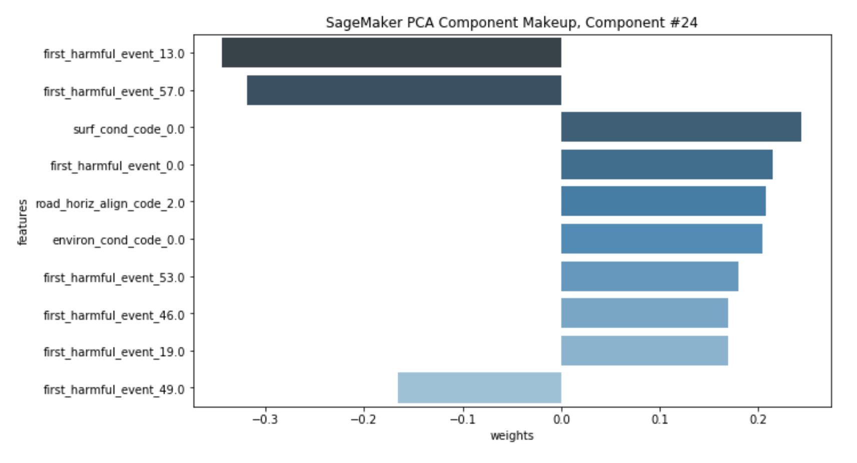

SageMaker PCA Component Makeup, Component #19
features
first_harmful_event_58.0
first_harmful_event_53.0
surf_cond_code_5.0
hs_collision_type_9
first_harmful_event_23.0
surf_cond_code_6.0
road_sys_code_8
first_harmful_event_19.0
surf_cond_code_0.0
first_harmful_event_6.0
−0.3
−0.2
−0.1
0.0
0.1
0.2
weights

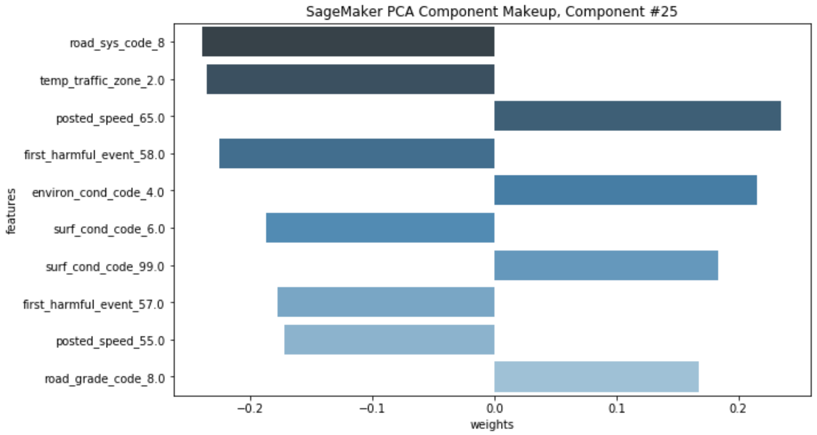

SageMaker PCA Component Makeup, Component #20
features
first_harmful_event_11.0
light_cond_code_3.0
first_harmful_event_58.0
first_harmful_event_99.0
temp_traffic_zone_2.0
hs_collision_type_8
temp_traffic_zone_3.0
crash_type_5.0
road_grade_code_6.0
first_harmful_event_45.0
−0.5
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
weights


SageMaker PCA Component Makeup, Component #21


SageMaker PCA Component Makeup, Component #22


SageMaker PCA Component Makeup, Component #23


SageMaker PCA Component Makeup, Component #24


SageMaker PCA Component Makeup, Component #25
road_sys_code_8
temp_traffic_zone_2.0
posted_speed_65.0
first_harmful_event_58.0
environ_cond_code_4.0
surf_cond_code_6.0
surf_cond_code_99.0
first_harmful_event_57.0
posted_speed_55.0
road_grade_code_8.0
features
−0.2
−0.1
0.0
0.1
0.2
weights