Determining How Americans Get Hurt On the Job: *Evaluating unsupervised machine learning as a method for classifying injuries.*

Authors

Greg Skotzko

Mark Joy

Kevin Heslin

Venkat Sastry

Abstract

This project evaluates the feasibility of using unsupervised machine learning algorithms to identify groupings for individual injury codes, with K-means clustering of a dataset on occupational injuries made available by the U.S. Occupational Safety and Health Administration (OSHA). The objective was to determine whether unsupervised learning could identify clinically meaningful clusters of injury codes from the Occupational Injury and Illness Classification coding system used by OSHA to classify occupational injuries, thus introducing new efficiencies into standard approaches for summarizing information from large disease coding systems. The final K-Means feature set yielded a silhouette score of .848. These clusters grouped in a way that grouped event codes in a non-binned fashion. The most frequent injury codes were grouped together within one cluster, indicating a strong relationship. Further investigation of these codes found that all 3 relate to serious injuries caused by body parts being caught, compressed or pinched by machinery or equipment. We did not find that our k means results identify clinically meaningful clusters of individual injury codes. Instead, the results and visualizations show strong clustering by industry code and the event that caused the injury. Although this analysis didn’t see the clustering by injury code, we believe our unsupervised machine learning analysis represents a preliminary proof of concept.

Background

Numerous diagnostic coding systems are available to classify medical conditions, services, and procedures for clinical, administrative, and financial purposes. These systems are essential to the productivity of countless individuals working in the fields of health and social care, education, research, and information services, from small rural and inner-city clinics in the U.S. to the World Health Organization (WHO) in Geneva, Switzerland. Hospitals and other healthcare organizations use such codes in billing third-party payers for care provided to covered patients. Physicians using electronic medical record software to document the process of care are often prompted to select codes from drop-down code lists that correspond to the diagnoses they make and the services they provide for their patients. Epidemiologists, policy analysts, and other researchers analyzing healthcare administrative databases typically select subsets of related diagnostic codes to define study populations and create analytic samples.

Challenges for Clinicians and Researchers.

With more than 70,000 diagnosis codes, the International Classification of Diseases (ICD) system – overseen by the WHO and adopted by all developed and most developing countries – often includes multiple codes for various manifestations of the same underlying disease. For example, the 10th edition of the ICD-Clinical Modification system ("ICD-10-CM") introduced in the U.S. in October 2015 includes 27 different codes referring to opioid abuse/dependence, including such fine clinical distinctions as: *Opioid abuse with intoxication, uncomplicated* (code # F11120); *Opioid dependence with opioid-induced psychotic disorder with hallucinations* (F11251); and *Opioid dependence with unspecified opioid-induced disorder* (F1129). This level of granularity and detail is certainly a strength of the ICD and other coding systems; however, it also presents numerous challenges for those working in clinical care and research. Just months after the introduction of the ICD-10-CM, a physician with a large caseload of elderly patients with diabetes complained, "I just want to code for diabetes, plain and simple. I don't want to have to scroll through a drop-down menu of dozens of diabetes codes when I'm trying to attend to my patients' needs."[1] For researchers, the large volume of codes presents additional challenges in creating meaningful categories for producing summary health statistics.[2] Further, major revisions of the ICD and other coding systems can complicate trend analyses due to incompatibilities between different editions

[1] Dr. Claudia Steiner, 2016; personal communication.

[2] Khera R, Dorsey KB, Krumholz HM. Transition to the ICD-10 in the United States: An Emerging Data Chasm. JAMA. 2018; 320:133-134.

of the same system. Longitudinal databases that include diagnosis and procedure codes from different editions of a single coding system can introduce discontinuities in trend lines that are solely due to changes in coding systems, independent of true changes in the prevalence of conditions and use of care. [3]

Aggregation of Disease and Procedure Codes.

A standard approach to dealing with these challenges is to group individual codes into broad, aggregated classes that are clinically meaningful. Aggregation can help smooth the discontinuities in trend analysis estimates that arise from revisions in coding systems, if coding within the broader aggregated classes is relatively stable over time. Aggregation can also help minimize the types of misclassification error that result when individuals who assign diagnosis and procedure codes in clinical or administrative data systems do so without sufficient attention to detail. To return to the opioids example, a clinician could easily miscode a patient who has abuse with intoxication (code # F11120) as has having dependence with opioid-induced psychotic disorder (F11251), particularly if such miscoding has no implications for the amount of reimbursement that he or she will receive from third-party payers. A researcher using individual codes at such a fine level of detail – as though minor misclassifications in coding were not to be expected – is likely to produce extremely misleading analytic results. For this reason, an approach that involves the aggregation of all 27 opioids codes into one overall "opioid-related diagnosis" category is often advisable for analytic purposes, as a way to minimize the error that surely occurs at the level of the individual diagnosis code when clinicians and other professionals assign codes to encounter and billing records.

"Code Groupers."

To perform code aggregation in a reliable and valid way, a number of different "groupers" have been developed and tested for use in research. The Clinical Classification Software (CCS) developed by the Agency for Healthcare Research and Quality (AHRQ) and the Hierarchical Conditions Categories developed by the Centers for Medicare and Medicaid Services are two widely used groupers that analysts apply to hospital administrative data to organize and aggregate diagnosis codes into different

[3] Heslin KC, Owens PL, Karaca Z, Barrett ML, Moore B, Elixhauser A. Trends in opioid-related inpatient stays shifted after the U.S. transitioned to ICD-10-CM diagnosis coding in 2015. Medical Care 2017; 55: 918-923.; Heslin KC, Barrett ML. Shifts in alcohol-related diagnoses after the introduction of ICD-10-CM coding in U.S. hospitals: Implications for epidemiologic research. Alcoholism: Clinical and Experimental Research 2018; 42: 2205-2213.

disease categories.[4] Other groupers include the International Shortlist for Hospital Morbidity Tabulation[5] , the 3M All Patient Refined Diagnosis-Related Groups Grouper[6], and the Adjusted Clinical Groups Case-Mix System.[7] These tools serve a variety of different purposes and vary in terms of which clinical conditions are considered, the number of diagnosis codes that are included in each disease group, and the total number of groups. AHRQ's CCS grouper aggregates over 14,000 ICD-9-CM codes into 285 mutually exclusive categories, all of which are clinically homogeneous. Approximately 900 peer-reviewed journal articles have been published using the CCS.[8]

Problem Statement

The process of developing a code grouper is labor-intensive and costly. As noted above, the ICD-10-CM was released in October 2015. As of September 2019, AHRQ has still not released a final version of the CCS for ICD-10-CM. Part of the problem is no doubt due to the sheer volume of diagnosis and procedure codes that need to be reviewed and categorized by federal contractors with both clinical expertise and knowledge of previous coding systems. The costs of procuring the required level of expertise to develop new groupers by federal agencies could also contribute to delays in development, especially if these costs need to be absorbed within fixed annual agency budgets. While such expert reviewers are essential to developing a reliable and valid grouper, identifying new approaches that could potentially reduce the time-to-market for these important tools is necessary.

Capstone Objective

In this project, we evaluate the feasibility of using unsupervised machine learning algorithms to identify groupings for individual injury codes, with K-means clustering of a dataset on occupational injuries made available by the Occupational Safety and Health Administration, an agency within the U.S. Department of Labor. The objective is to determine whether unsupervised learning could reduce the substantial time and financial costs associated with standard approaches to developing groupers by producing a preliminary "first draft" of clusters of similar codes, which could then be reviewed and modified by

[4] Elixhauser A, Steiner C, Palmer L. Clinical Classifications Software (CCS), 2013. U.S. Agency for Healthcare Research and Quality. Available: http://www.hcup-us.ahrq.gov/.

[5] Wong A, Boshuizen HC, Schellevis FG, et al. Longitudinal Administrative Data Can Be Used to Examine Multimorbidity, provided false discoveries are controlled for. Journal of Clinical Epidemiology 2011; 64: 1109–17.

[6] Shen Y. Applying the 3M All Patient Refined Diagnosis Related Groups Grouper to Measure Inpatient Severity in the VA. Medical Care 2003; 41:103–110.

[7] Adams EK, Bronstein JM, & Raskind-Hood. Adjusted Clinical Groups: Predictive Accuracy for Medicaid Enrollees in Three States. Health Care Financing Review 2002; 24:43–61.

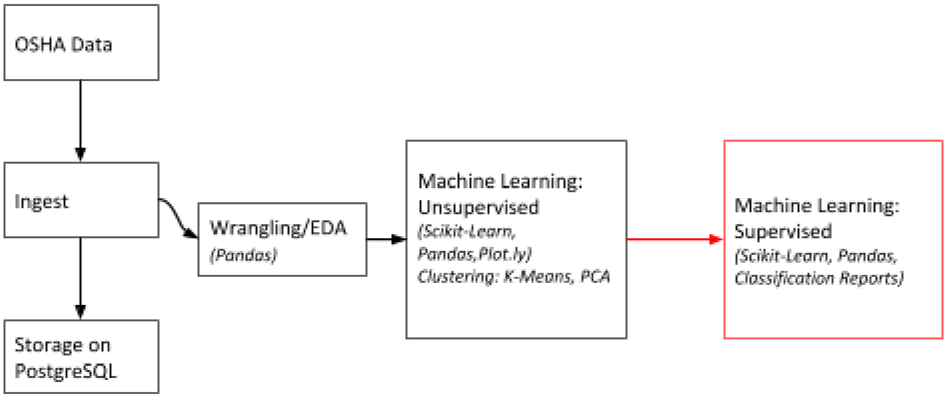
[8] For more information on the CCS, visit: https://www.hcup-us.ahrq.gov/reports/pubsearch/advanced.jsp

expert reviewers in real-world projects. If successful, we believe that unsupervised learning could introduce new efficiencies into the standard grouper development process and help alleviate the clinical, administrative, and analytic challenges involved in working with large disease coding systems.

Project Architecture

Before diving into the key processes, such as ingestion, wrangling, and machine learning, outlined below is our data pipeline (*Figure 1)* created to optimize the flow of our project and to ensure we have all of the right tools.

(Figure 1: Severe Injury Data Pipeline)


OSHA Data
Ingest
Storage on PostgreSQL
Wrangling/EDA
(Pandas)
Machine Learning: Unsupervised
(Scikit-Learn, Pandas,Plot.ly)
Clustering: K-Means, PCA
Machine Learning: Supervised
(Scikit-Learn, Pandas, Classification Reports)

Data Ingestion & Wrangling

The data ingestion and wrangling took up the majority of our team's time, which based on what all of the professors in the program said about this process, is par for the course. For the purpose of our project, we used two data sources, the Occupational Safety & Health Administration's (OSHA) Severe Injury Report data and Census TIGER data, which are described in further detail below.

This section will address three core aspects of the ingestion and wrangling phases. First, we explore our data sources to gain a deeper understanding of the background and features of these datasets. Second, we address the challenges that our team faced during ingestion and wrangling. Third, we focus on feature engineering to prepare our data for machine learning.

Data Sources

OSHA Severe Injury Report Data

On January 1, 2015, OSHA mandated that, “employers must report to OSHA within 24 hours any work-related amputation, in-patient hospitalization, or loss of an eye.”[9] This was an effort to:

1. “Enable the agency to better target our compliance assistance and enforcement efforts to places where workers are at greatest risk, and;
2. 2. Engage more high-hazard employers in identifying and eliminating serious hazards.”[10]

The reporting program is going into its fourth year, and as of July 2019, has data reported up through 2018. Data was downloaded from OSHA’s Severe Injury Report website[11] and stored as a CSV file. This dataset contains 27 features, around 39,834 instances, and various data types. Among the features in the dataset, there is latitude, longitude, date, hospitalization, and amputation.[12] In addition to these features, OSHA uses the Occupational Injury and Illness Classification (OIIC) manual[13] that provides keyed values for the different codes used in the severe injury dataset and the 2017 North American Industry Classification System (NAICS) index[14] to enable linking between injury types and industry. Of the 39,834 observations in the dataset, 28 observations with invalid values for amputation and 248 observations with invalid values for hospitalization. Lastly, while it fell outside of the scope of this project, the dataset includes a narrative feature describing the injury event, so we have an opportunity to engage in some natural language processing work in a follow up project.

We found that the Severe Injury Reporting data required quite a bit of wrangling in which our team used pandas, a data analysis Python library.[15] Wrangling this dataset consisted of eliminating missing values,

[9] Michaels, David. Year One of OSHA’s Severe Injury Reporting Program: An Impact Evaluation. Occupational Safety & Health Administration 2016; 1-8. Available: https://www.osha.gov/injuryreport/2015.pdf

[10] Ibid.

[11] For more information on OSHA’s Severe Injury Report, visit: https://www.osha.gov/severeinjury/index.html

[12] Hospitalization and amputation are binary variables.

[13] The Occupational Injury and Illness Classification (OIIC) manual “contains the rules of selection, code descriptions, code titles, and indices, for the following code structures: Nature of Injury or Illness, Part of Body Affected, Source of Injury or Illness, Event or Exposure, and Secondary Source of Injury or Illness.” For more information, visit: https://www.bls.gov/iif/oshoiics.htm

[14] 2017 North American Industry Classification System (NAICS) index: “The North American Industry Classification System (NAICS) is the standard used by Federal statistical agencies in classifying business establishments for the purpose of collecting, analyzing, and publishing statistical data related to the U.S. business economy.” For more information on the NAICS index, visit: https://www.census.gov/eos/www/naics/

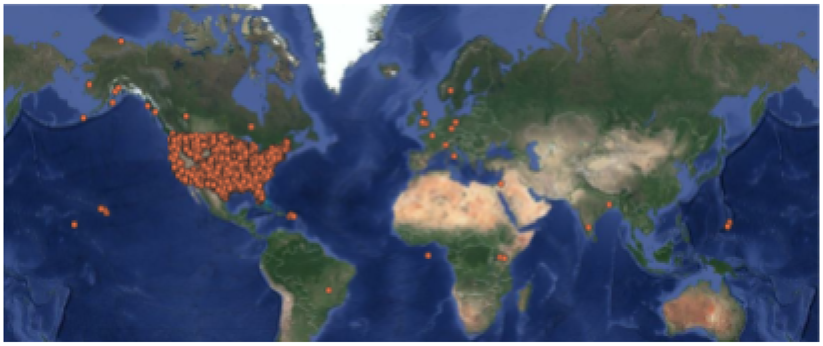
[15] For more information on pandas, visit: https://pandas.pydata.org/

converting data types, creating a new data frame that only included the relevant features, and lastly, exploring value counts of the binary features and removing the unique values that weren't 0 or 1.

Census TIGER Data

We also acquired Census TIGER data[16] so we could incorporate county-level names (*Figure 2*). We used QGIS[17], an open source graphic information system, to spatially join the county name to each incident report based on the Lat/Long of the incident. This was to ensure that the county name assigned reflected the location of the incident instead of potentially representing the City/State in which the company is registered.

(Figure 2: Geospatial Distribution of Injury Records)



Based on Figure 2, we observed that there are records for US Company employees that were injured in other countries. As a result of this, we realized that this means the zip codes reflect the zip code of the company registration, not the zip code of the incident. Our team decided to narrow the scope of the project to the injuries that occurred within the US. To do this, we created a feature to flag whether instances occurred outside of the US (0 = No, 1 = Yes). After exploring this feature, we found that 58 observations represented occupational injuries occurring outside the United States. We dropped the 58 instances and then deleted the feature considering it was no longer needed.

Feature Engineering

The final aspect of wrangling that we needed to engage in was feature engineering on the subset of variables. Two non-numerical features "State" and "County" were encoded in preparation for exploratory data analysis. To do this, we used Scikit-Learn, a Python library that enables machine learning.[18] Because machine learning algorithms can't function well with categorical data and better understand numeric data, we used a transformer method within Scikit-Learn, Label Encoder.[19] Label

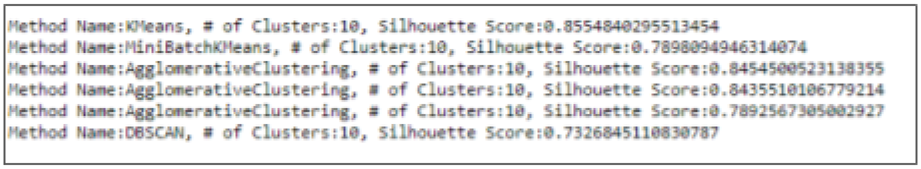
[16] "The geodatabases contain national coverage (for geographic boundaries or features) or state coverage (boundaries within state)." For more information on the TIGER/Line Geodatabase, visit: https://www.census.gov/programs-surveys/geography/technical-documentation/complete-technical-documentation/tiger-geodatabase-file.html
[17] For more information on QGIS, visit: https://qgis.org/en/site/
[18] For more information on Scikit-Learn, visit: https://scikit-learn.org/stable/index.html
[19] For more information on Scikit-Learn's LabelEncoder, visit: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

Encoder converted ‘State’ and ‘County’, the categorical features, into numerical data. We did stop at this step and because of this, our machine learning algorithms may have potentially misinterpreted an ordinal relationship between the encoded features. To avoid this potential pitfall, One-Hot Encoding[20] should be used to further transform the data to better fit a vectorized space. One-Hot Encoding is a method that “maps each category to a vector that contains 1 and 0 denoting the presence or absence of the feature.”[21] At this point our data was cleaned and appropriately encoded to ensure optimal machine learning modeling results.

Data Analysis

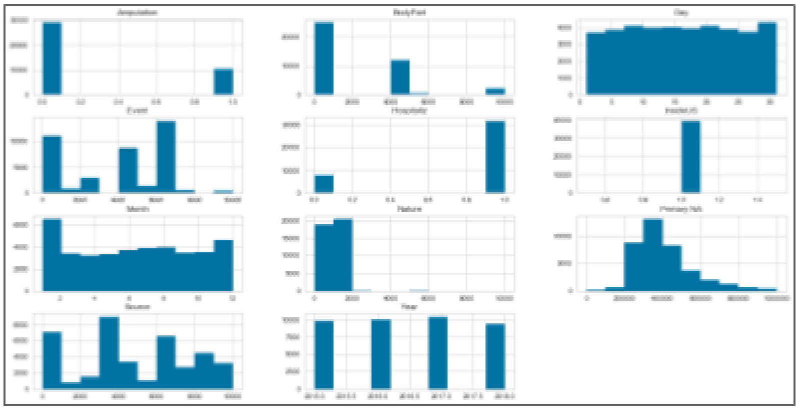
The exploratory data analysis of the project took up the majority of our time over. The tools and packages used during this phase of the project ranged from pandas, to a range of tools from the Scikit-Learn library, such as, Yellowbrick, for machine learning diagnostic visualizations, to a whole host of clustering models, such as K-means[22], Mini-Batch K-means[23], Agglomerative[24], and DBScan.[25] See Figure 3 for the scores of these clustering models.

(Figure 3: Scoring of multiple clusterer models)



```
Method Name:KMeans, # of Clusters:10, Silhouette Score:0.8554840295513454
Method Name:MiniBatchKMeans, # of Clusters:10, Silhouette Score:0.7898094946314074
Method Name:AgglomerativeClustering, # of Clusters:10, Silhouette Score:0.8454500523138355
Method Name:AgglomerativeClustering, # of Clusters:10, Silhouette Score:0.8435510106779214
Method Name:AgglomerativeClustering, # of Clusters:10, Silhouette Score:0.7892567305002927
Method Name:DBSCAN, # of Clusters:10, Silhouette Score:0.7326845110830787
```

(Figure 4: Histogram of numeric features)



Exploratory Data Analysis

Vertically plotted histograms were produced for 13 attributes (Figure 4). The attributes represented in the histograms range from whether an

[20] For more information on Scikit-Learn’s One-Hot Encoder, visit: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

[21] Roy, Baijayanta. All about Categorical Variable Encoding. Towards Data Science 2019. Available: https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02

[22] For more information on Scikit-Learn’s K-means, visit: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

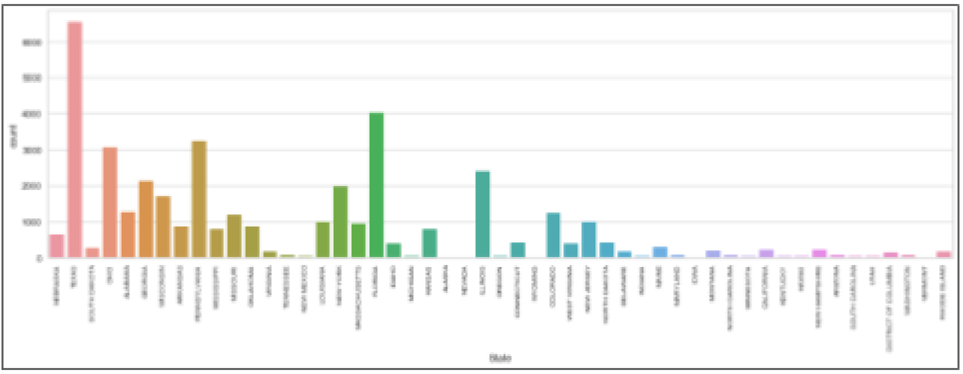
[23] For more information on Scikit-Learn’s Mini-Batch K-means, visit: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MiniBatchKMeans.html

[24] For more information on Scikit-Learn’s Agglomerative Clustering, visit: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

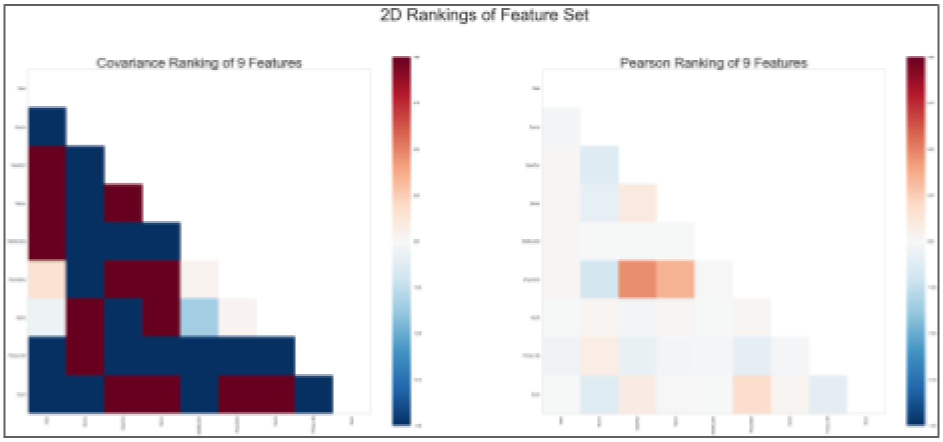
[25] For more information on Scikit-Learn’s DBScan Clustering, visit: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

amputation or hospitalization occurred, to the nature and injury event, to the industry codes to the date (day, month, and year) on which the injury took place. We also produced a bivariate histogram displaying the number of injuries occurring in each state included in the dataset (Figure 5). Among the states for which the most injuries occurred, Texas, Ohio, Pennsylvania, and Florida appear to rank highest. For two subsets of features, we produced a Covariance Ranking matrix and a Pearson Ranking matrix. Although the focus of our project was unsupervised machine learning, we thought it was advantageous to see if there were any initial insights we could glean from such visualizations. **Figure 6** depicts one set of visualizations attempted excluding the target in this case, which was whether hospitalization occurred. Based on the graphs, we can see some relationships between features but as outlined below in Next Steps, we would re-run these models after we've defined our clusters in the unsupervised machine learning portion of our project. As part of the K-means clustering analysis, we calculated and plotted Distortion Scores to identify the "elbow" or optimal number of clusters to include as a parameter in the final analysis (Figure 7). KElbow Visualization of both KMeans and MiniBatchKMeans showed an elbow at the k=10 mark, the resulting silhouette scores for both models with clusters yielded silhouette scores above .80. We then produced Silhouette plots for K = 10 . We plotted the Intercluster Distance for the K = 10 clusters to visualize the degree of overlap between them (Figure 8).

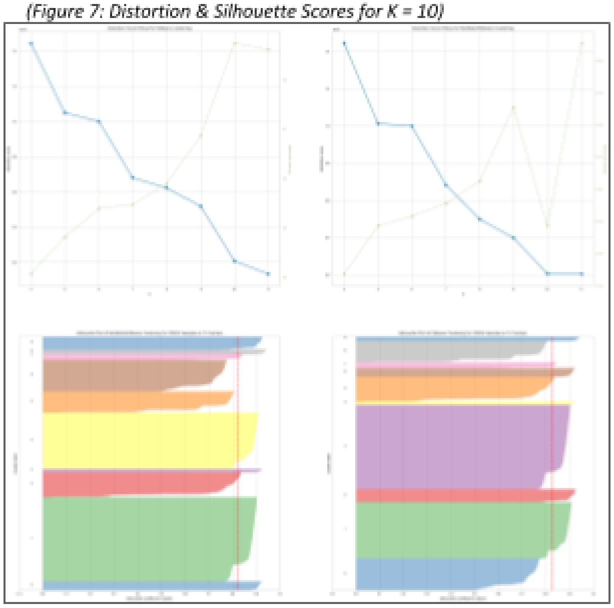
(Figure 5: Injury count by state)



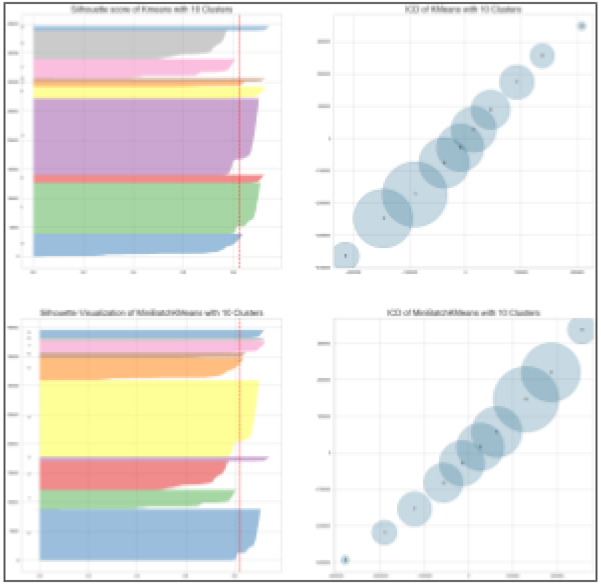
(Figure 6: Covariance Ranking of 9 features - excluding the target at the time (hospitalization))



(Figure 7: Distortion & Silhouette Scores for K = 10)

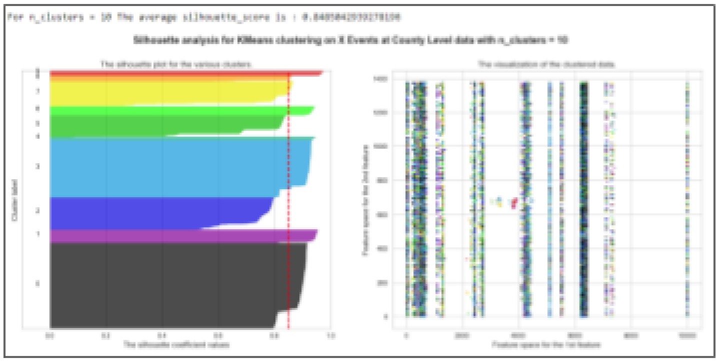


(Figure 8: Silhouette scores and Intercluster distance)



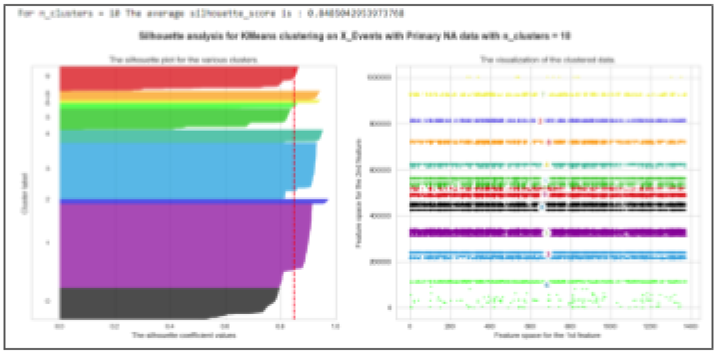
Evaluating Feature Sets for the K-Means Analysis

After having settled on 10 clusters for the K-means analysis, we evaluated which set of features to cluster. To that end, a function was written to run multiple feature sets through K-means analyses with a range of clusters (k) from 6-12, by 2s. From the results of each feature set analysis, the clusters were plotted. We calculated the Silhouette score associated with each feature set and calculated the average score for all sets. While all the feature-sets scored reasonably well from K-means and MiniBatchKMeans, plotting the clusters indicated which features scored highly but did not cluster well (Figures 9 and 10).



```
For n_clusters = 10 The average silhouette_score is : 0.8485042053278156
```

(Figure 9: Example of a high scoring but poorly clustered feature set. In this example, unique injury codes with county, NAICS code, Hospitalization and Amputation were included.)



```
For n_clusters = 10 The average silhouette_score is : 0.8485042053973768
```

(Figure 10: Our chosen feature set- almost identical silhouette score as larger feature set but with distinct clusters.)

Principal Components Analysis (PCA) was used to reduce the dimensionality of the feature space, after which we plotted the centroids from the PCA-reduced data. We then calculated Silhouette scores, average scores, and plots for K = 4, then 6, then 8, then 10, and finally 12 clusters for each feature set.

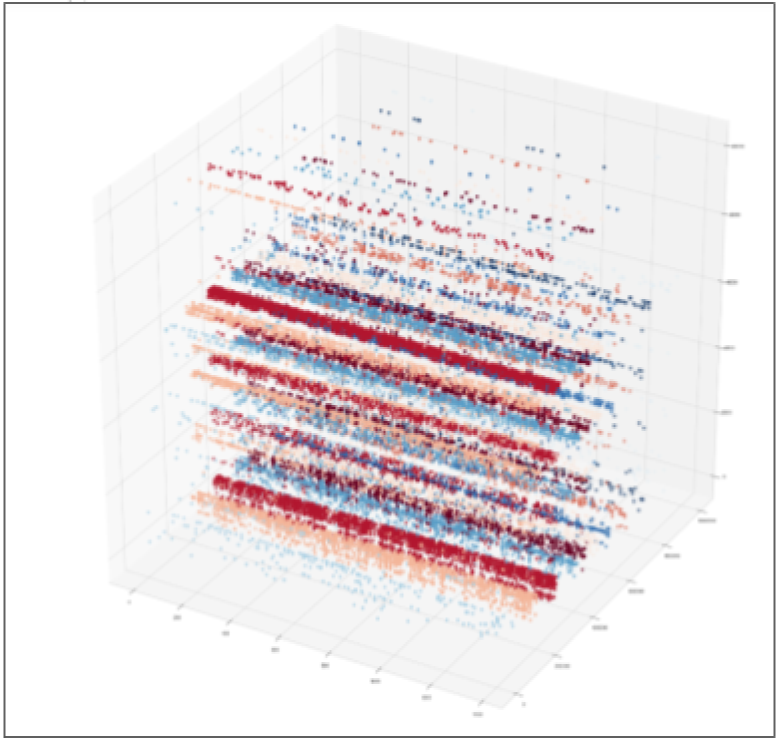
Turning Clusters into Features

By retaining the unique OSHA incident ID , “UPA”, as the index, each of the 10 clusters identified in the K-means analysis was added to the feature-set and then joined to the original data-frame as a new feature for further analysis. The resulting data frame was then saved out to a CSV file for further use in future step implementing supervised machine learning, such as KNearest Neighbors, Bagging and XGBoost models.

Results & Visualizations

Multiple permutations of KMeans and MiniBatchKMeans, implementing uniquely different feature-sets and cluster values yielded a KMeans model using County, Event Code (specific injury) and Primary North American Industry Classification System (NAICS) code. Implementing KMeans with 10 clusters on this feature-set yielded a silhouette score of .848. These clusters grouped in a way that grouped event codes in a non-binned fashion (as you would find in a frequency histogram of Event codes) (Figures 11 & 12).

(Figure 11: 3D Plot from MatPlotLib of Event Code, Primary NAICS Code, and County.)

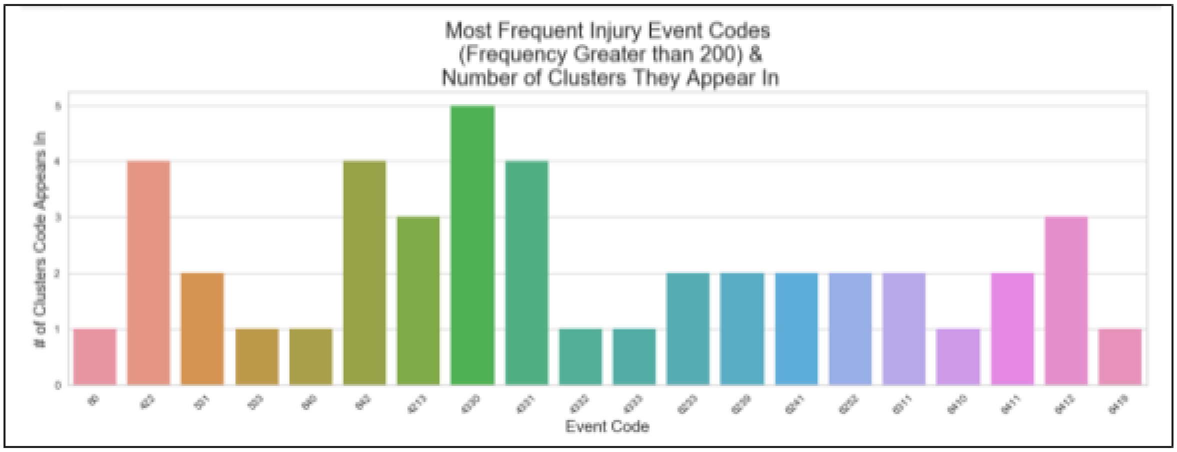


(*Figure 12: Cluster feature sets)*

Feature Set	**# of Clusters**	**Silhouette Scores**	**Good Clusters**
Event Code, State	6	.882	No
Event Code, County, NAICS Code	10	.848	**Yes**
Event Code, State, Latitude, Longitude	6	.879	No
Event Code, State, County, Body Part, Nature of Injury, Amputation, NAICS Code	10	.823	No

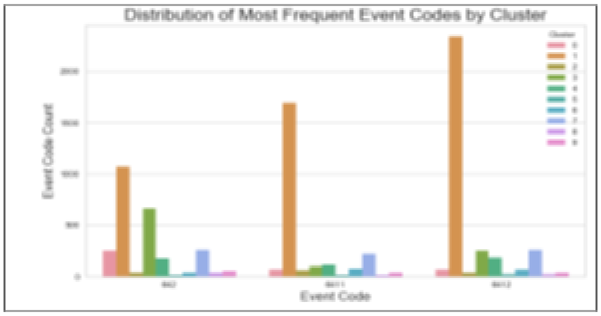
Further analysis of the clustered feature set showed that some of the codes have a frequency of occurrence greater than 200 within a cluster (Figure 13). 20 event codes have a frequency greater than or equal to 200, 70% grouped into two or fewer clusters.

(Figure 13: Injury event code frequency)

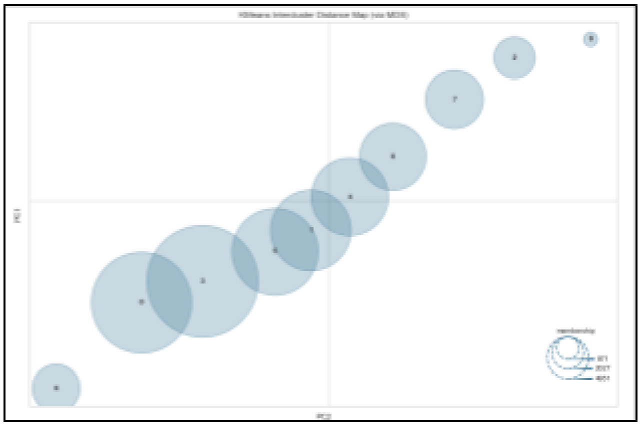


The most frequent injury codes were grouped together within one cluster, indicating a strong relationship. Further investigation of these codes found that all 3 relate to serious injuries caused by body parts being caught, compressed or pinched by machinery or equipment. This information could be utilized by safety administrators or regulators to shape and improve safety and maintenance procedures - such as strongly training employees to observe “Lock Out-Tag Out” procedures. The distribution of the most frequent Event codes (Figure 14) across clusters helps explain the overlap of clusters observed in Intercluster Distance Maps (Figure 15).

(Figure 14: Event code frequency by cluster)



(Figure 15: Intercluster distance map)



Conclusion

The goal of our capstone project was to demonstrate that unsupervised machine learning could reduce the substantial time and financial costs associated with standard approaches to developing groupings. Specifically, the team set out with the objective to evaluate the feasibility of using K-means clustering to identify clinically meaningful clusters of individual injury codes. Rather than yielding clusters if injury codes, what our results and visualizations show are clustering by industry code and the event that caused the injury. While we didn't see the clustering by injury code, we believe our unsupervised machine learning analysis represents a preliminary proof of concept. We believe that with subsequent analyses using our framework will yield a time & cost-saving alternative to the standard grouping approaches.

Lessons Learned

Our biggest lesson learned is to thoroughly explore your intended dataset. We revised the data source three times due to the nature of data. While we maintained a focus on injury, we shifted from Emergency Rooms to OSHA Severe Injuries. Another lesson learned was that wrangling the data IS a very time-intensive aspect of the work. Lastly, during the wrangling and machine learning portions of the work, we learned that, if we had more time, we would have spent more time on feature selection and/or gradient boosting, and would have engaged supervised machine learning to yield some deeper insights about the now-labeled clusters produced in the K-means analysis.

Next Steps

Additional Features

Considering that there are more than 5000 fatalities and many more injuries every year, there is certainly more to be learned about the data and how information derived from it could be used to reduce injuries and fatalities. Industry information is an example that can be used to add an extra dimension to these data.

The clustering of the data used the date, event type and location information roughly. As we analyze these data further, we could add industry information, company names, inspection data, employer information, and safety training data to reveal information about the injuries that are still hidden within the data.

Additionally, it may be advantageous to explore adding weather data into the mix. Incorporating weather data into the clusters could add a deeper layer of of context/dimension. For example, using weather, industry information, and the narrative of the event would only deepen our understanding of the clusters in a follow up K-means analysis.

Sample Weighting

For the entirety of our analysis, we left sample weighting of KMeans to the default setting. This ensured that all X’s were given the same weight as KMeans ran through multiple iterations. It may be worth exploring the weighting system to see if certain X’s score and cluster better (or worse).

Dimensionality Reduction with Neural Networks

Though we used PCA for our capstone project, as we add more features, we think that we might be able to get better results if we use some sort of neural network dimensionality reduction before we apply k-means clustering. Using neural networks to reduce dimensionality, under the right conditions, might improve our results. Preliminary research indicates that auto-encoders which fall under ANN model family (Artificial Neural Networks), could help in reducing the number of dimensions and if we apply k-means on those reduced dimensions instead of the raw data, we might be able to get clusters which we could not find earlier. To do this, we would most likely use TensorFlow, a machine learning Python library, which could help us construct our neural network in only a few lines of code.[26]

The key factors we have determined is the number of neurons, the number of layers, the number of epochs (number of times neuron will look into the model and try to fit it) and the learning rate. This will be needed for us to make sure that the neural network is able to process the data within a reasonable time frame.

Natural Language Processing

One last avenue we would take in our next steps is further exploration of the narrative feature in our dataset. The narratives contain text-rich data that if tapped into might yield a wealth of new and useful insights. Applying classification models to the text narratives for the injuries to extract specific information that can be useful to medical facilities. We would do this by using the Natural Language Toolkit (NLTK), a Python library used for natural language processing (NLP) and machine learning.[27]

Github Location of Project Code

https://github.com/georgetown-analytics/Injury-Prediction

[26] For more information on TensorFlow, visit: https://www.tensorflow.org/

[27] For more information on NLTK, visit: https://www.nltk.org/