Improving Workplace Accident Fatality Classification Models

with Text Mining and Ensemble Methods

Thomas S. Wilk, Jr.

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Data Mining

Department of Mathematical Sciences

Central Connecticut State University

New Britain, Connecticut

November 2013

Thesis Advisor:

Dr. Daniel T. Larose

Department of Mathematical Sciences

Improving Workplace Accident Fatality Classification Models

with Text Mining and Ensemble Methods

Thomas S. Wilk, Jr.

An Abstract of a Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Data Mining

Department of Mathematical Sciences

Central Connecticut State University

New Britain, Connecticut

November 2013

Thesis Advisor

Dr. Daniel T. Larose

Department of Mathematical Sciences

Key Words: Data Mining, Text Mining, Predictive Modeling, Natural Language Processing, Ensemble Methods, Python, IPython Notebook, Pandas, Scikit-Learn

**ABSTRACT**

Using publically available OSHA accident investigation data and open source Python scientific computing tools this applied research project asserts the value of exploiting all types of data available when constructing predictive models. The goal of this project was to build the best classification model of accident outcome, either fatal or non-fatal, using data available for catastrophic accident investigations conducted by OSHA over the last few decades. Multiple feature sets were engineered using a variety of techniques that leveraged all types of data available, including structured, semi-structured and unstructured accident attributes.

This thesis proposes that features mined from each accident’s text-based attributes will capture concepts and information that are not present in each accident’s structured data attributes and that this infusion of new information will enable classification algorithms to better discriminate between fatal and non-fatal accidents, thereby improving model accuracy. Baseline classification models of accident outcome were trained on a feature set created from the structured accident attributes only. With the goal of improving baseline classification accuracy, a variety of text mining and data mining techniques were employed to create statistics-based and linguistics-based feature sets from accident keywords, descriptions and summaries. These efforts resulted in a measurable improvement in model accuracy.

This thesis also asserts the value of ensemble methods and proposes that combining multiple predictive models trained on the same feature set will obtain better performance than could be obtained from any of the component models independently. Ensemble methods in the form of voting models and mean response probability models capitalized on a confluence of results from multiple classifiers. Classifiers that internalized boosting techniques and ensemble learning, such as AdaBoost and Random Forest, were also utilized. Finally, combined feature sets composed of top performing predictors selected across the various text-based and structured data feature sets engineered in this thesis provided the greatest lift to model accuracy.

**TABLE OF CONTENTS**

ABSTRACT.....................................................................................................................................3

INTRODUCTION….......................................................................................................................6

1. Statement of Purpose.....................................................................................................6
2. Stated Hypothesis/Research Questions..........................................................................8
3. Statement of Need........................................................................................................10
4. Related Research..........................................................................................................10
5. Data Sources................................................................................................................12
6. Online Project Repository............................................................................................14

ANALYSIS....................................................................................................................................15

1. Structured Feature Set..................................................................................................15
2. Unstructured Text Preprocessing.................................................................................34
3. Keyword Feature Set....................................................................................................37
4. Linguistic Feature Set..................................................................................................42
5. Topic Feature Set.........................................................................................................48
6. Description SVD Feature Set.......................................................................................54
7. Summary SVD Feature Set..........................................................................................59
8. Combined Feature Set..................................................................................................64

CONCLUSION .............................................................................................................................69

FURTHER RESEARCH...............................................................................................................70

REFERENCES..............................................................................................................................72

APPENDIX A:Python Scientific Computing Resources and Packages.......................................74

APPENDIX B:Keyword Feature Set Additional Figures.............................................................75

APPENDIX C:Linguistic Feature Set Additional Figures...........................................................78

APPENDIX D:Topic Feature Set Additional Figures..................................................................79

APPENDIX E:Description SVD Feature Set Additional Figures................................................80

APPENDIX F:Summary SVD Feature Set Additional Figures...................................................83

APPENDIX G:Combined Feature Set Additional Figures...........................................................86

**INTRODUCTION**

**Statement of Purpose**

The primary purpose of this study was to conduct applied research using publically available data that combined text mining and machine learning, two related areas of academic and professional interest. Mining features from unstructured data to enhance the performance of predictive models built initially upon structured data only was a topic expected to be of considerable interest amongst members of the data mining community. A parallel purpose was to implement all phases of the analysis with freely available open source data mining tools, and by doing so demonstrate their viability as an alternative to proprietary applications. The Python scientific computing ecosystem comprises a set of powerful and flexible tools that were selected to implement this thesis. All project phases, including data wrangling, data analysis, machine learning and visualization, were implemented with Python based tools. The result of using publically available data and free, open source tools to conduct this particular project is a completely transparent, accessible and (hopefully) interesting thesis. In this spirit, project code, imperfections and all, are available to the reader should he or she desire to improve, reproduce or leverage any aspect of this project. All assets created to implement this thesis, in the form of IPython Notebooks and a single Python code module, are accessible by the reader at an online public GitHub repository.

In 1970, the Unites States Congress created the Occupational Safety and Health Administration (OSHA), a national public health agency dedicated to the basic proposition that no worker should have to choose between their life and their job. OSHA’s mission is “to assure safe and healthful conditions for working men and women by setting and enforcing standards and providing training, outreach, education and compliance assistance.” [*All About OSHA*]. To enforce standards OSHA conducts approximately 100,000 inspections annually and conducts investigations when catastrophic workplace accidents occur. Datasets comprising inspection and accident investigation case detail are made available to the general public via the U.S. Department of Labor’s Data Enforcement website [*OSHA Enforcement Data*]. The inspection dataset “includes information regarding the impetus for conducting the inspection, and details on citations and penalty assessments resulting from violations of OSHA standards.” The accident investigation dataset includes “textual descriptions of the accident, and details regarding the injuries and fatalities which occurred. “ The OSHA accident data is stable, voluminous, of high integrity, and publically available. Each accident contains a robust set of attributes that provide a mix of structured, semi-structured and unstructured data types. The OSHA accident data was well suited for a combined application of machine learning and text mining techniques, and was an ideal dataset for this thesis.

The goal of this project was to build the best classification model of accident outcome, either fatal or non-fatal, given a workplace accident occurred and was investigated by OSHA. Multiple feature sets used to model accident outcome leveraged all types of data available, including structured, semi-structured and unstructured accident attributes. Although not as actionable in its raw form, the unstructured data, after being pre-processed, transformed and optimized for modeling, was expected to embody new information not present in the structured data that would improve classification model performance beyond that of models built with the structured data only. This is in fact the central research question and project challenge.

**Stated Hypothesis/Research Questions**

The central hypothesis of this thesis is that features mined from each accident’s text-based attributes would capture concepts and information that were not present in each accident’s structured data attributes and that this additional information would enable classification algorithms to better discriminate between fatal and non-fatal accidents, thereby improving model accuracy. Stated as a question,

**Research Question One:** Will inclusion of features mined from unstructured data attributes associated with each accident improve the accuracy of predictive models constructed with structured accident data attributes only?

Initial exploration of the OSHA accident data as a foundation for this thesis revealed that answering this question and validating the hypothesis would not prove to be a straight-forward task. OSHA has been in operation for decades. Their efforts have decreased workplace accidents and fatalities over time and have contributed to safer workplace conditions for all employees in most industries. Analysis of data collected routinely by OSHA enables identification of the causes of catastrophic workplace accidents and the detection of emerging trends. As would be expected, the accident and injury data captured by OSHA was comprehensive along multiple dimensions. Using a data mining platform of choice and a minimal investment of time, a proficient practitioner is able to download the structured accident data, apply minimal transformation, and generate predictive models that classify accidents as fatal or non-fatal with accuracy rates of 80%+ on a test holdout set. Although the text-based OSHA accident data was ideally suited for an application of predictive text mining techniques, the structured data attributes on their own were already strong predictors of accident outcome. Features mined from the text-based accident data would not be expected to prove their worth easily. Although a cause of initial hesitation, the OSHA data was ultimately selected for his thesis precisely because the structured variables were so information-rich. Top performing models fit to structured data features only would provide solid baselines from which to gauge the efficacy of models infused with features derived from unstructured text.

Ensemble methods combine multiple predictive models to obtain better performance than could be obtained from any of the component models independently. Ensemble methods in the form of voting models and mean response probability models capitalize on a confluence of results from multiple classifiers. Combined feature sets that leverage top predictors from constituent feature sets are another example of ensemble methods. Additionally, classification models, such as AdaBoost and Random Forest classifiers, internalize boosting techniques and ensemble learning to achieve greater predictive gains. The secondary hypothesis of this thesis was that application of ensemble methods, in the three forms stated above, would in fact provide additional lift to classification model performance. Stated as a question,

**Research Question Two:** Will a confluence of model results from different classification algorithms trained on the same feature set, each independently deployed to address the challenge of research question one, produce more accurate classifications of accident outcome in concert than on their own? Additionally, will the combination of top-performing predictor variables from statistics-based text mining feature sets, a linguistics-based text mining feature set and the structured data feature set achieve yet more gains?

**Statement of Need**

The proprietary nature of successful text mining and machine learning models deployed by for-profit organizations tends to inhibit their publication. Likewise, the cost associated with proprietary data mining applications tends to limit their ownership and broad usage by individuals. By contrast, this study offers up a transparent application of text mining and machine learning methods, using publically available datasets of a serious nature, implemented with software that any individual can download and use. This author believes that veteran data miners and newcomers alike can benefit from exposure to this project and the publication of its code, methodology and results. This project is a small contribution of thanks to the open source data science community whose hard work made it possible.

**Related Research**

This thesis is an application of mainstream data mining and text mining techniques. There is nothing exotic or controversial happening here. Related research for this project can be segmented into three categories. The first category was comprised of Python scientific computing books, documentation and tutorials. The second category was comprised of eight foundational data mining, text mining and natural language processing books. A plethora of online, open-source publications, examples, tutorials, question-answer forums and blogs comprised the third category.

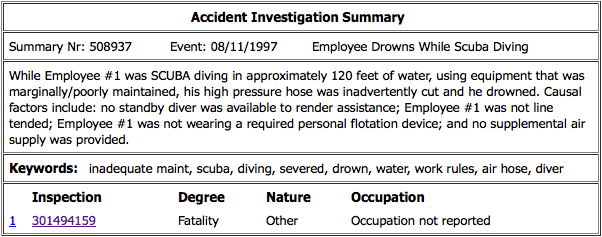
A large chunk of background research required an investment of hundreds of hours of learning to implement in the Python scientific computing environment general data analysis and predictive modeling techniques implemented by this author in past projects using proprietary applications such as Excel, SQL Server and IBM SPSS Modeler. An additional investment of time was required to learn how to leverage advanced functionality that distinguished the Python scientific computing environment from other applications. Appendix A lists the various resources and packages that comprised the scientific computing ecosystem of the Python programming language used in this thesis. These resources were utilized to implement all project phases. Scikit-learn, a machine learning library, and Pandas, a data analysis toolset, were two Python resources that were used extensively in this project and required a considerable investment of time researching and hacking to earn a level of modest proficiency required to implement most phases of this project. One awesome benefit of the open source nature and popularity of the Python ecosystem is freely available software, online documentation, tutorials and examples. In lieu of citing resources here, the interested reader is encouraged to follow the links provided in Appendix A, or conduct a quick web search, to learn more.

Foundational data mining, text mining and natural language processing textbooks provided guidance for the various techniques, decisions and approaches used in this analysis. In the data mining category were two general data mining books authored by Daniel Larose and a machine learning book implemented in Python and authored by Willi Richert. In the text mining category were two text mining books authored by Sholom Weiss, et al., a text mining book authored by Roger Bilisoly, and a text mining book authored by Gary Miner, et al. On the natural language processing front was a book written by the authors of the Natural Language Toolkit, a popular Python NLP resource.

**Data Sources**

The main sources of data used in this project were publically available files downloaded from the U.S. Department of Labor’s Data Enforcement website, OSHA Enforcement Data page [*OSHA Enforcement Data*]. The OSHA data used in this project was last refreshed in early November 2013. The interested reader is encouraged to navigate to the site to learn more about the OSHA datasets, to query and browse accident case detail, and to download the actual data files used in this thesis, if desired. A dozen or so files were available for download. The files spanned four main categories: violations, inspections, accidents and metadata. This study focused on the three accident investigation datasets only. One additional publically available online resource used in this project was the SentiWordNet file. SentiWordNet is a lexical resource for opinion mining that assigns to each synset of WordNet three sentiment scores: positivity, negativity, objectivity [*SentiWordNet].*  SentiWordNet was leveraged to extract linguistic features from accident text summaries.

Table 1 below is a snapshot of output for a selected accident from OSHA’s online Accident Search tool. Although not a complete representation of all attributes available for each accident, the main types of data used in this analysis are present. Examples of structured data attributes that comprised the Structured feature set are the *Event Date* and categorical variables such as *Occupation* and *Nature of Injury*. Many other structured data fields existed in the raw accident investigation files beyond that which is shown here. The *Keywords* field included with each accident is a great example of semi-structured data.

Table 1: Example Accident Investigation Summary

The number of keywords and keyword categories vary considerably across accidents but there is still a pattern that one quickly intuits – each keyword associated with an accident is separated by commas. Transformation of the list of keywords needs to be performed before the keywords can be leveraged for modeling. The transformed keywords across accidents comprised the Keyword feature set. Each accident contains two unstructured data fields, the *Event Description* and *Text Summary*. The short phrase on the top right, the *Event Description*, contains a brief description of each accident. The center box, the *Text Summary*, contains a lengthier, more detailed description of each accident. More effort will be required to extract features from these unstructured text fields to use as predictors during the modeling phase. The *Degree of Injury* field represents the target variable. Note the unique accident and inspection identifier numbers on the left side corners. These identifiers were not used as modeling input variables but were retained on all records throughout the analysis and provided reference back to specific accidents.

**Online Project Repository**

All project datasets, Python code and IPython Notebooks are posted at a freely available public GitHub repository created especially for this project and will remain there indefinitely. The interested reader can view all project IPython Notebooks online via links provided below. A zip file of all project assets can be downloaded from the repository homepage with one simple click. Provided that a compatible Python distribution with the requisite packages listed in Appendix A is installed on a local computer, the interested reader can execute, modify and extend this project in its entirety. The Anaconda Scientific Python Distribution used for this project is freely available to download and comes preloaded with most of the requisite packages. Additional posts on the project wiki, accessible from the homepage, will provide guidance on execution of all project resources in the correct sequence.

*Project repository homepage*

<https://github.com/tswilk/ccsu-thesis-project-osha>

*IPython Notebook links*

<https://github.com/tswilk/ccsu-thesis-project-osha/wiki/Links-to-view-IPython-Notebooks-online>

*Anaconda Scientific Python Distribution*

<https://store.continuum.io/cshop/anaconda/>

**ANALYSIS**

**Structured Feature Set**

*Section IPython Notebook link:* [*osha\_01\_structured\_feature\_set.ipynb*](http://nbviewer.ipython.org/7535438)

The goal of this section was to transform the raw OSHA data files into a single feature set with one row per accident, or observation, and carefully selected features, or predictor variables, as columns, in preparation for subsequent modeling efforts. The predictor variables were fashioned out of the raw structured data attributes. Feature sets created in subsequent sections leveraged the raw semi-structured and unstructured accident attributes. Additional actions were taken to filter out accident observations and accident attributes not suitable for modeling, create new, more optimal features based on existing features, replace codes with meaningful labels and address missing values.

The files *osha\_accident.csv* and *osha\_accident\_injury.csv,* located on the project repository page, were downloaded from the OSHA website in early November 2013. The Accident file contained one row per accident and multiple attributes that pertained to the accident. The Injury file contained multiple rows per accident, one for each injured person involved in each accident, along with attributes of the injury. The manually created file, entitled *osha\_code\_map.csv*, also housed on the project repository, provided a mapping of codes to descriptions for fields relevant to this analysis and was used to replace codes present in the raw data fields with more legible and meaningful descriptions.

The raw Accident file comprised 103,771 accidents over the last 40 years. Each accident contained a field indicating fatality or non-fatality. This field was transformed into a binary indicator (1=fatal accident, 0 = non-fatal accident) and served as the target variable throughout the analysis. Figure 1 is a histogram of accidents by year with outcome overlay. Note the large jump in the number of accidents between 1989 and 1990, with no significant trend in the years leading up to 1989 and the years following 1990. Documentation on the OSHA website indicated that data for accidents occurring in 2008 and prior were fully represented in the online datasets but not so for accidents that occurred in 2009 and later. This explained the steep drop in total accidents for years 2009 to 2012, as the most current years were most incomplete. The normalized histogram of accidents by event year depicted in Figure 2 indicated a very slight downward trend in the proportion of fatal accidents. This trend may be due to a delay in the reporting of more serious, fatal accidents. Despite this potential systematic difference, a decision was made to partition data from the incomplete years of 2009 to 2012 into a validation holdout set and to use data from years 1990 to 2008 for training and testing classification models.

Figure 1: Histogram of Accidents by Year with Outcome Overlay

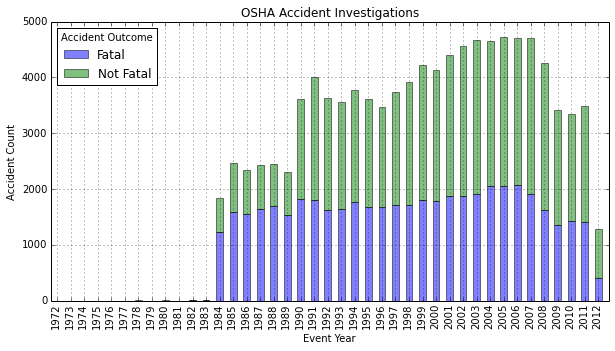
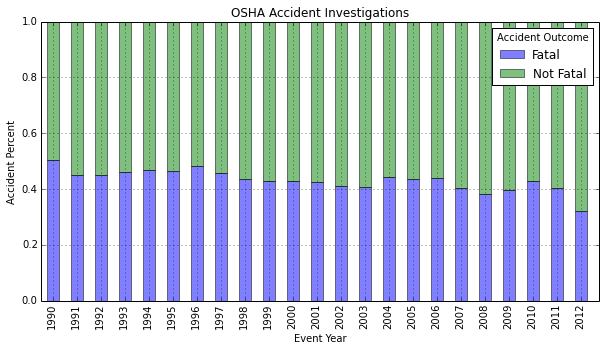


Figure 2: Normalized Histogram of Accidents by Year with Outcome Overlay



A few more data preparation steps were taken to refine the raw Accident data. A trinary variable with values of fatality, hospitalization and non-hospitalization was dropped as it was a proxy for the fatality indicator and the focus of this project was binary classification of accident outcomes as fatal or non-fatal. Time based variables *event year*, *event month*, *event weekday* and *event hour* were derived from the raw *event date* timestamp. SIC codes were replaced with SIC descriptions. Empty accident variables, unnecessary variables, and variables that did not contain information relative to the majority of accidents were omitted. This step excluded a set of variables pertaining to construction specific accidents that would likely be of interest for projects with a construction segment focus. Finally, the semi-structured and unstructured fields associated with each accident were temporarily omitted for purposes of the Structured feature set, but were utilized in the construction of text-based feature sets in later sections.

The raw Injury file comprised 130,410 injuries. Injury data associated with accidents out of scope were immediately filtered out. Table 2 provides statistics on the number of injuries per accident. A decision was made to retain accidents with single injuries and exclude accidents with two or more injuries. This action removed 7.8% of the total recorded accidents from 1990 to 2012. Note that 43% of all single-injury accidents were fatal. Not having to accommodate multiple injuries per accident simplified the construction of the Structured feature set, as single rows of accident data and injury data could be merged into one observation easily.

Table 2: Number of Injuries per Accident Distribution

|  |  |  |  |
| --- | --- | --- | --- |
| Number Injuries Per Accident | Accident Count | Portion Total Accidents | Average Fatalities |
| 1 | 82,840 | 92.2% | 0.43 |
| 2 | 4,130 | 4.6% | 1.08 |
| 3 | 1,356 | 1.5% | 1.36 |
| 4 | 578 | 0.6% | 1.68 |
| 5 | 282 | 0.3% | 1.74 |
| 6 | 168 | 0.2% | 1.75 |
| 7 | 122 | 0.1% | 2.35 |
| 8 | 70 | 0.1% | 2.06 |
| 9 | 62 | 0.1% | 2.47 |
| 10 | 67 | 0.1% | 2.39 |
| 11 | 30 | 0.0% | 2.57 |
| 12 to 153 | 190 | 0.2% | 6.14 |

A few more data preparation steps were taken to refine the raw Injury data. Injury variables that contained information relative to a minority of accidents, such as construction specific variables, a hazardous substance variable associated with a subset of accidents, and fall distance metrics, were omitted. Clearly, projects with a more narrow focus could likely benefit from inclusion of these variables. The *age* and *sex* variables were dropped as they contained unary values of zero and null, respectively. Eight injury attributes were retained and their coded values were mapped to descriptions. Finally, 939 observations were dropped due to missing codes, and therefore missing descriptions, for all injury attributes.

The next step was to merge the retained accident and single injury attributes into a single feature set with one record per accident. The resulting table of 81,874 single-injury accidents formed the basis of the Structured feature set after data preparation. A few more transformations were applied during the upcoming exploratory data analysis phase prior to modeling. Table 3 below depicts a sample record from the feature set.

All predictor variables that comprised the Structured feature set were categorical or binary in nature. For exploratory data analysis histograms, heatmaps, distribution charts and bar charts were generated to help gauge how useful the categorical variables might prove to be as predictors of accident outcome. The target variable was included as a graphical dimension to aid diagnosis. A review of all visualizations suggested that the majority of categorical variables and indicators would likely be useful as predictors of accident outcome.

Table 3: Sample Record from the Structured Feature Set After Data Preparation

|  |  |  |
| --- | --- | --- |
| Full Name | Variable Name | Typical Variable Value |
| Accident ID Number | summary\_nr | 508937 |
| Fatality Indicator | fatality\_ind | 1 |
| Event Timestamp | event\_ts | 8/11/97 10:30 |
| Event Year | event\_year | 1997 |
| Event Month | event\_month | 8 |
| Event Weekday | event\_weekday | 0 |
| Event Hour | event\_hour | 10 |
| SIC Description | sic\_desc | Amusement and Recreation Services |
| Inspection Number | rel\_insp\_nr | 301494159 |
| Nature of Injury | nature\_of\_inj | OTHER |
| Part of Body | part\_of\_body | LUNG |
| Source of Injury | src\_of\_injury | WATER |
| Event Type | event\_type | CARD-VASC/RESP FAIL. |
| Environmental Factor | evn\_factor | OTHER |
| Human Factor | hum\_factor | PERCEPTION MALFUNC,TASK-ENVIR. |
| Occupation Code | occ\_code | Occupation not reported |
| Task Assigned | task\_assigned | TASK OTHER THAN REGULARLY ASSIGNED |
| OSHA Detail URL | osha\_detail\_url | [Accident Detail Link](https://www.osha.gov/pls/imis/establishment.inspection_detail?id=301494159) |

Figures 3 and 4 depict heatmaps for two selected categorical variables. The number of fatalities by accident year and categorical value are represented by color, with darker colors indicating higher fatality counts. Similar visualizations for the other categorical variables can be viewed within the IPython Notebook accompanying this section. Injuries with an *Event Type* of ‘Struck By’ are the top cause of fatalities, followed by ‘Fall (From Elevation)’ and ‘Caught In Or Between’. The remaining event types vary by the number of fatalities over time. The last five event types appear to rarely result in fatality, if at all. It is interesting that the greatest number of fatalities for *Source of Injury* was attributed to an ‘Other’ category. Accidents involving motor vehicles were the next greatest source of fatalities.

Figure 3: *Event Type* Heatmap

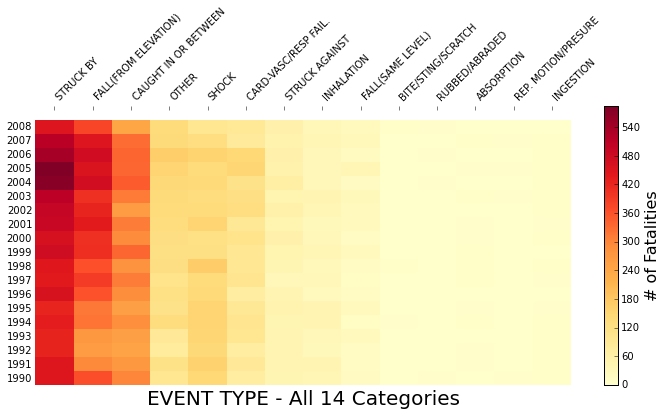
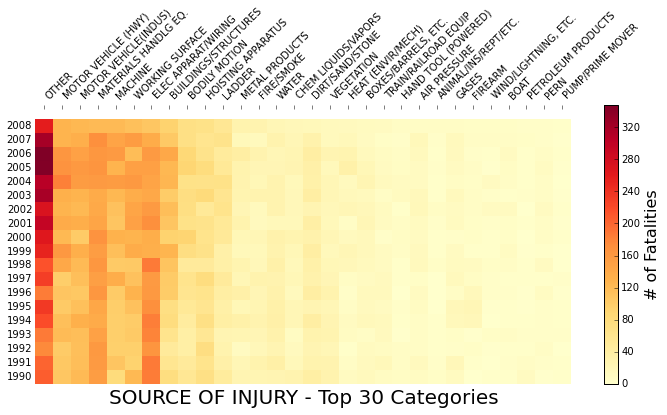


Figure 4: *Source of Injury* Heatmap



Figures 5 and 6 depict bar charts of the number of accidents across all years with outcome overlay for two selected categorical variables. A normalized distribution of accidents by outcome accompanies each bar chart to help discern the proportion of fatal and non-fatal outcomes by category. The top ten categories are shown separately with remaining categories aggregated into a single category. The proportion of fatalities varied significantly across categories for both the *Nature of Injury* and *Part of Body* variables. These variables were expected to aid classification models to discriminate between accident outcomes. Again, similar visualizations for the other categorical variables can be viewed within the IPython Notebook accompanying this section.

Figure 5: *Nature of Injury* Distribution Chart



Figure 6: *Part of Body* Distribution Chart

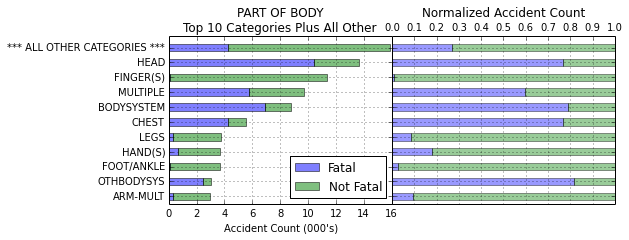


Figure 7 depicts histograms of the time variables *year, month, weekday* and *hour.* A normalized histogram with accident outcome overlay accompanies each plot. Although the number of accidents peaks during the summer and early fall, the proportion of accidents that result in fatality is fairly constant throughout the year. The number of accidents and proportion of fatal accidents is level during the weekdays with the fewest accidents occurring on Friday. Although the number of accidents on weekend days is less than half of accidents that occur on weekdays the proportion of fatalities given an accident occurred is slightly greater. As would be expected most accidents occur during daytime working hours with a lull during the noontime lunch hours. The fewest accidents occur during the early morning hours with slightly more occurring during the late evening hours. The spike in accidents at midnight is due to a large portion of accidents with accurate dates but without accompanying time information. A decision was made to not adjust for this feature.

Additional actions were taken as a result of exploratory data analysis to prepare the Structured feature set for modeling. The *year* and *month* variables were dropped. The *weekday* variable was transformed into a binary *weekend* *indicator* (1=weekend, 0=not weekend). The two-state categorical variable *task assigned* was transformed into a binary *regular task* *indicator* (1=regular task, 0=not regular task). Figure 8 depicts plots of the binary indicator values for the two newly transformed variables by average fatality.

Figure 7: Time Variable Histograms with Accident Outcome Overlay

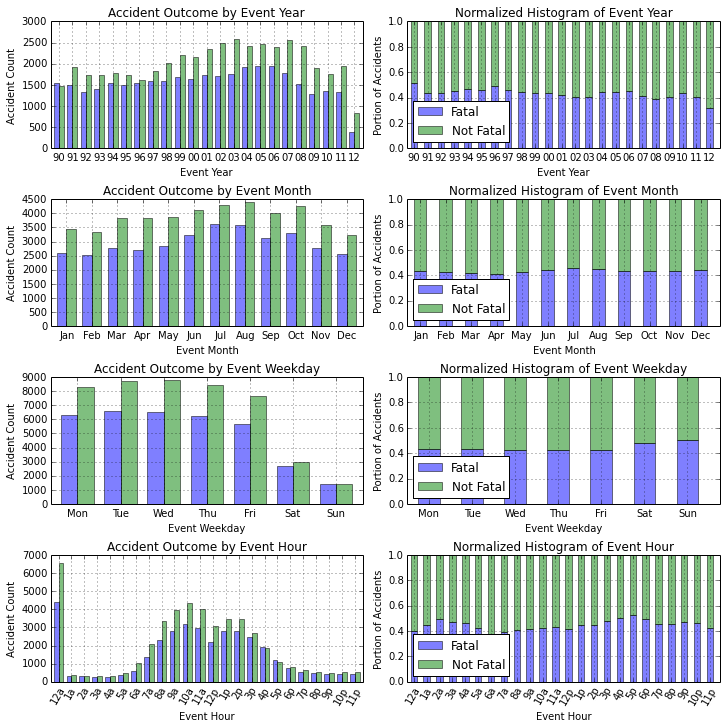
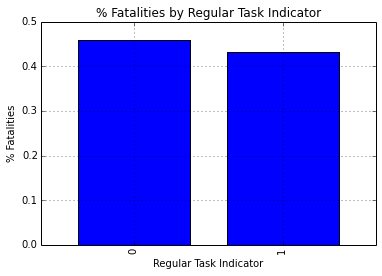
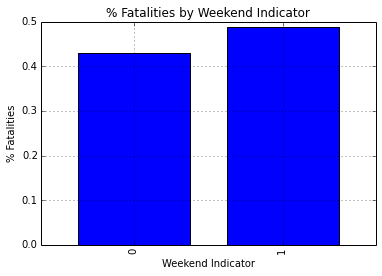


Figure 8: Binary Indicator Variables



The final Structured feature set for modeling contained 263 predictor variables spanning 81,874 accident observations. The predictor variables were comprised of the nine categorical variables converted into 261 binary indicators and the two binary variables *weekend indicator* and *regular task indicator*. Transforming each categorical feature with *m* possible values into *m* binary features, with only one active, was necessary as the classification algorithms used in this project required continuous input variables. The categorical variables are listed in Table 4 along with the number of distinct values that equated to the number of binary indicators created for each variable. Each observation contained a binary target variable indicating whether the accident resulted in a fatality or not. This was the target variable. Classification models will attempt to predict whether or not a fatality occurred given the set of predictor variables provided for each accident.

Table 4: Categorical Variables Transformed Into Flag Indicators for Modeling

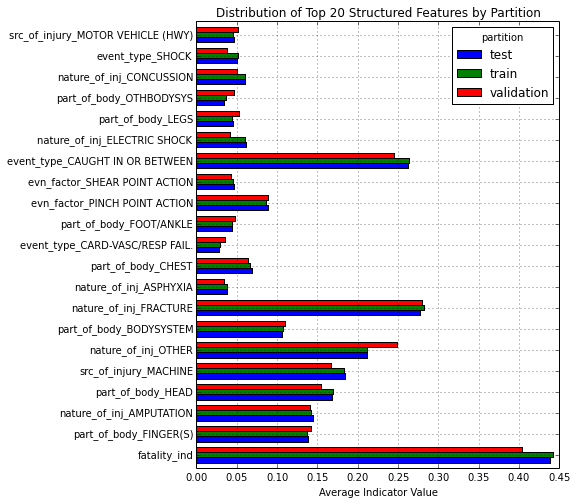
|  |  |
| --- | --- |
| Categorical Variable | Distinct Values |
| Part of Body | 31 |
| Environmental Factor | 18 |
| Event Type | 14 |
| Event Hour | 24 |
| Human Factor | 20 |
| Nature of Injury | 22 |
| SIC Group | 84 |
| Source of Injury | 48 |
|  |  |

At this point data preparation and EDA was performed on the Structured feature set and the structured data predictors and target variable were set and stable. Before modeling accident outcomes with the Structured feature set, partitioning the feature set into a train, test and validation set was indicated. Due to the information-rich nature of the feature set, and to guard against model over-fitting, a 30% / 70% train and test set partition was selected. Recall that accidents from years 2009 to 2012 were earmarked to serve as a holdout validation set. After setting the validation set accidents aside, the train and test partitions comprised 71,101 accidents in total. Accidents were randomly assigned to train and test partitions based on the 30% / 70% split. The data was split relatively evenly with respect to the target variable as indicated by the mean percentage of fatal accidents across partitions in Table 5. To ensure that distributions of variables across the train and test set were similar, the top 20 most important features were selected by computing ANOVA F-value statistics between each binary predictor variable and the binary target variable. Figure 9 depicts a similar distribution of the top 20 features and fatality indicator across the train, test and validation partitions as measured by each feature’s average indicator value.

Table 5: Distribution of Accidents into Train, Test and Validation Partitions

|  |  |  |
| --- | --- | --- |
| Partition | Accident Count | Percent Fatalities |
| Train | 21,331 | 44.2% |
| Test | 49,770 | 43.8% |
| Validation | 10,773 | 40.4% |
| Overall | 81,874 | 43.5% |

Figure 9: Similar Distribution of Top Features Across Train and Test Partitions



Five classification algorithms were employed in this project for model training of all feature sets. These algorithms were selected for their speed and for their availability within the *scikit-learn* machine learning library. For simplicity and for consistency across all feature sets, the *scikit-learn* default model configurations for each classification model were selected. Table 6 lists the five classification algorithms, the short name used to refer to each classifier throughout the remainder of this analysis, and a brief description.

Table 6: Classification Models



Two types of combination models were generated during the modeling phase of each feature set based on the aggregated test results of the five base classification models:

* The voting models counted the number of fatal accident predictions from each of the five base models for each accident test case and predicted fatality if the count of fatal predictions was equal to or greater than a set number of models. The voting model with threshold of 4+, for example, predicted fatality if 4 or 5 of the models predicted fatality, and predicted non-fatality otherwise. To identify the optimal threshold, five models were generated with thresholds 1+, 2+, 3+, 4+ and 5.
* The mean response probability models (MRP) capitalized on the confidence measures, between 0 and 1, which each model delivered with its predictions. The MRP model with threshold 40, for example, predicted fatality if the average confidence in fatal predictions for a given accident test case from each of the five classification models was greater than or equal to 0.40, and predicted non-fatality otherwise. To identify the optimal threshold, eleven models were generated with MRP thresholds of 25 to 75 incremented by 5.

Each of the five classification algorithms was fit to the train partition data of the Structured feature set and evaluated on the test partition data. Table 7 lists the eight evaluation metrics computed, along with their definition. Model evaluation statistics for the five base classification models and top performing combination models are summarized in Table 8.

Table 7: Classification Model Evaluation Metrics

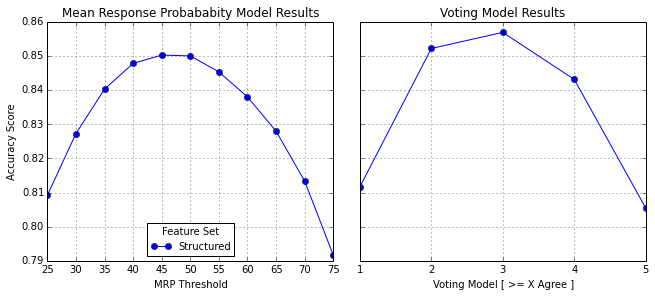


Table 8: Base Model and Combination Model Results

****

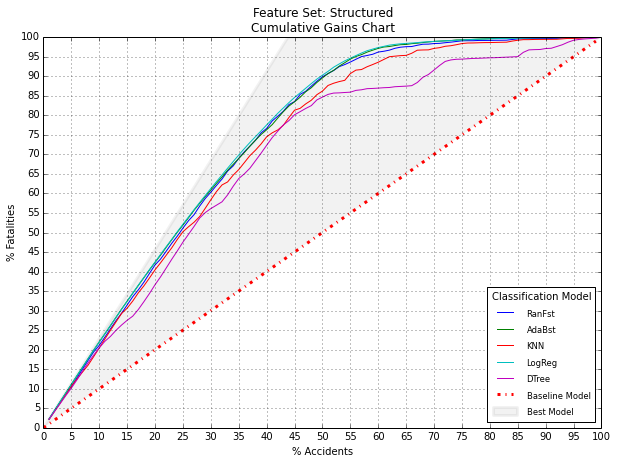
The Logistic Regression model was the top performer with a classification accuracy score of 85.3%. This model was selected as the baseline structured data only model from which to measure the performance of models trained on text-based feature sets in upcoming sections. Note that the top MRP model came close and that the 3+ voting model earned a slight 0.4% improvement in accuracy. Accuracy scores at different thresholds for the combination models are plotted in Figure 10.

Figure 10: Combination Model Results



A cumulative gains chart for each of the five classification models is depicted in Figure 11. A cumulative gains chart is a graphical method for assessing model performance of one or more classification models. An explanation is in order here. The baseline model randomly selects test cases and predicts fatality each time such that x% of actual fatal accidents would, on average, be captured after sampling x% of the total accidents. To create the gains curve for each model, the model’s confidence in predictions of fatal outcome for each test case is ranked from high to low and the cumulative number of correct fatal predictions divided by the cumulative number of total predictions made is calculated at increasing percentages of overall accidents. Accidents that comprised the test partition resulted in fatality 43.8% of the time. The best model makes all accurate predictions such that 100% of fatal accidents are captured from 43.8% of all accidents scored. The two top performing models, Logistic Regression and Random Forest, captured about 90% of all fatal accidents in the first 50% of accidents scored.

Figure 11: Base Classification Model Cumulative Gains Chart



Figures 12 and 13 are plots of feature importance for the top 20 features of the Random Forest and Logistic Regression models, respectively. Importance for logistic regression input variables was computed from the regression coefficients provided by the model for each variable. Greater positive coefficient values indicated stronger influence on fatal outcome and greater negative values indicated stronger influence on non-fatal outcome. Calculation of feature importance of random forest input variables was more esoteric. *Scikit-learn* returns an array of feature importance for input variables fit to a random forest classifier. The importance of each feature is roughly the average rank, or depth of a feature from the root node, calculated from multiple randomized decision trees utilized in the random forest classifier.

Figure 12: Random Forest Classification Model Feature Importance

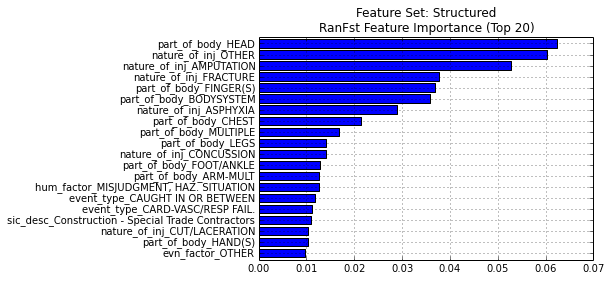
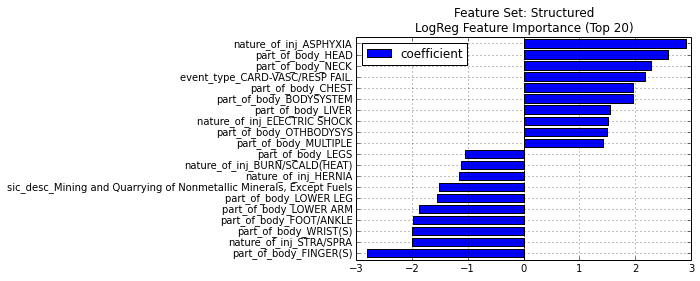


Figure 13: Logistic Regression Classification Model Feature Importance



**Unstructured Text Preprocessing**

*Section IPython Notebook link:* [*osha\_02\_unstructured\_preprocessing.ipynb*](http://nbviewer.ipython.org/7535439)

The OSHA accident abstract data contained multiple rows of text per accident. The raw abstract file, entitled *osha\_accident\_abstract.csv,* is located on the project repository. The first text pre-processing task was to combine these multiple lines of text into one text field per accident. Table 9 depicts a sample accident’s abstract records before and after transformation.

The second text pre-processing task involved removal of words from the Keywords, Event Description and Text Summary fields that were both highly correlated with fatality and of similar semantic meaning. Allowing these words to remain in the text would provide the text-based feature sets with a distinct advantage over the Structured feature set. Models initially generated with these words included achieved accuracy scores between 98% and 99%. Table 10 lists the two most frequent stemmed tokens from the Event Description field that were highly correlated with fatality and of similar semantic meaning. Over 96.6% of the time that the stemmed tokens *kill* and *die* occurred in the description the accident outcome was fatal.

Table 9: Merging Multiple Text Lines into a Single Text Field Per Accident



Table 10: Top Two Stemmed Words from Event Description Highly Correlated with Fatality

|  |  |  |  |
| --- | --- | --- | --- |
| Stemmed Token | # Occurrences in Non-Fatal Accident Descriptions | # Occurrences in Fatal Accident Descriptions | Percentage Fatal Accident Occurrence |
| kill | 101 | 20,344 | 99.5% |
| die | 260 | 7,427 | 96.6% |

On the other side of the spectrum words that were a variant of *hospital* resulted in fatalities 14% of the time, and were likewise removed. Variants of the word *employee* were also removed as they occurred in a majority of accidents and were essentially stopwords for purposes of this project. Table 11 lists all word variants that were removed from the unstructured text fields. Table 12 depicts a sample accident’s text fields before and after word removal. Although the resulting text is not always grammatically correct the application of subsequent text mining techniques used in this project were expected to tolerate these modifications without issue.

Table 11: Stemmed Word Proxies for Fatality and Corresponding Word Variants Removed

Table 12: Sample Accident Text Fields Before and After Modification



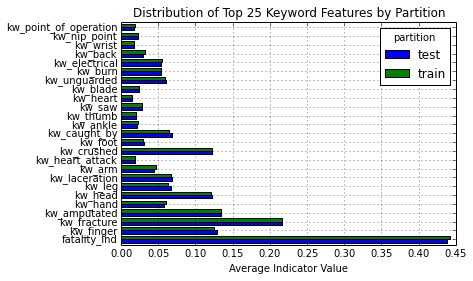
**Keyword Feature Set**

*Section IPython Notebook link:* [*osha\_03\_keyword\_feature\_set.ipynb*](http://nbviewer.ipython.org/7535441)

Keywords associated with each accident were provided as a single text field separated by commas in the raw accident data file. The pre-processing task here was quite simple as the data was already semi-structured. The feature set created from the keywords data was a binary matrix with accidents as rows and the set of 1,300+ keywords extracted from all accidents in the train partition as columns. Iterating through each accident a value of 1 was assigned to each keyword column associated with the accident and 0 was assigned for all other keyword columns. To ensure that distributions of keyword occurrences were similar across the train and test set the top 25 most important features were selected by computing ANOVA F-value statistics between each binary keyword indicator and the binary target variable. Figure 14 depicts a similar distribution of the top 25 features and fatality indicator across the train and test partitions as measured by each keyword’s average occurrence.

A K-Means cluster model with four clusters was fit to the train partition and evaluated on the test partition. Figure 15 demonstrates that the distribution of clusters across the train and test set was remarkably similar. Mean value distributions for each of the top 25 keywords and the fatality indicator were plotted by partition and cluster to facilitate inspection of the results. Inspection of the results from Figures B.1 and B.2 of Appendix B indicated that the K-Means cluster membership would likely prove to be an important predictor for the accident outcome classification models. In preparation for modeling, the cluster membership field was converted into flag indicators and added to the Keyword feature set.

Figure 14: Distribution of Top 25 Keywords by Partition



Each of the five classification algorithms was trained on the train partition data of the Keyword feature set and evaluated on the test partition data. Model evaluation statistics for the base and combination models are summarized in Table 13.

Figure 15: K-Means Cluster Membership and Counts by Partition

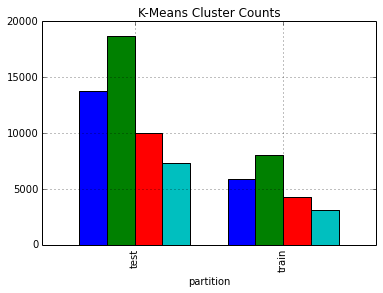


Table 13: Base Model and Combination Model Results



The Logistic Regression model was again the top performer with a classification accuracy score of 86.2%, a 0.9% improvement in accuracy from the baseline structured data only model. The combination models did not provide additional lift for this feature set. A cumulative gains chart for each of the five classification models is depicted in Figure 16. Accuracy scores at different thresholds for the combination models are plotted in Figure 17. Figures 18 and 19 are plots of feature importance for the top 20 features of the Random Forest and Logistic Regression models, respectively.

Figure 16: Base Classification Model Cumulative Gains Chart

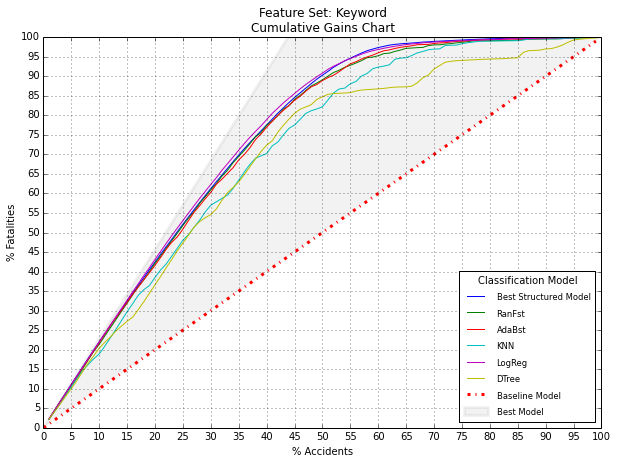


Figure 17: Combination Model Results

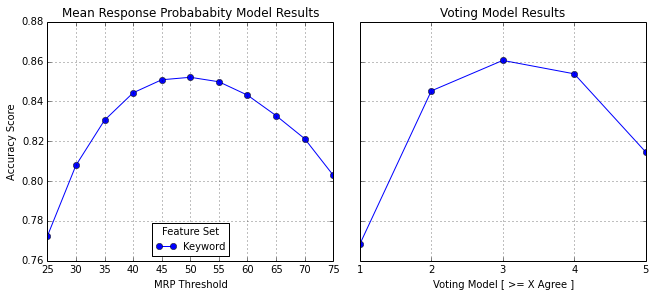


Figure 18: Random Forest Classification Model Feature Importance

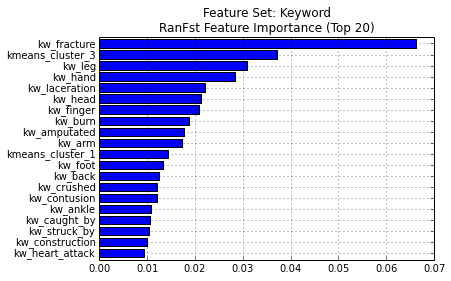
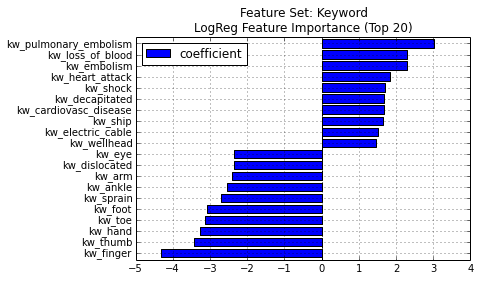


Figure 19: Logistic Regression Classification Model Feature Importance



**Linguistic Feature Set**

*Section IPython Notebook link:* [*osha\_04\_linguistic\_feature\_set.ipynb*](http://nbviewer.ipython.org/7535442)

This feature set was created by application of simple natural language processing techniques to extract linguistic features present within the unstructured accident summary data. The rationale was that linguistic features embedded within accident summaries might vary across accident outcomes in a manner that would enable classifiers to better discern between fatal and non-fatal accidents. With this goal in mind various features were extracted and assembled into a single feature set for subsequent modeling. The linguistic features might not prove to be strong predictors of accident outcome on their own but a few of them might provide an edge when combined with features from other feature sets at later stages of this project.

Features based on the main parts of speech (POS) from language used in the accident summaries were extracted. The rationale for inclusion was that a greater proportion of nouns might be used in less serious accident summaries and a greater proportion of adjectives and verbs might be used in more serious accident summaries. The POS tagger from the NLTK package was employed to extract counts of nouns, adjectives, verbs and adverbs from each accident summary. Another possibility was that fatal accidents might on average be lengthier, more descriptive and more lexically diverse. With this possibility in mind features such as summary text length, number of tokens and number of unique tokens were extracted from each accident summary. As the *sex* variable from the raw injury data was empty, counts of female prepositions and male prepositions were also extracted and fashioned into features.

The final two linguistic features were measures of average positive sentiment and average negative sentiment contained in each accident summary. Measures of sentiment were derived as features based on the reasoning that severe accidents might exhibit greater negative sentiment than less severe accidents and therefore assist models to better discriminate between accident outcomes. The idea, code base and implementation used to extract these particular features came from Richert’s excellent book *Building Machine Learning Systems with Python* (pg. 138). In fact, the book was inspiration for most of this project. SentiWordNet is a publically available resource file built on WordNet that assigns the majority of English words a positive and negative value and takes into consideration the part-of-speech of each word. The overall average negative value and overall average positive value of each word from the accident summaries comprised the two features. Figure 20 is a plot of average negative sentiment versus average positive sentiment for each accident in the train and test set with accident outcome overlay. Although the results were stable across partition, the measures of sentiment were not expected to be strong predictors of accident outcome. These measures were included as modeling inputs nonetheless.

Figure 20: Average Positive and Negative Sentiment Scatterplot with Accident Outcome Overlay

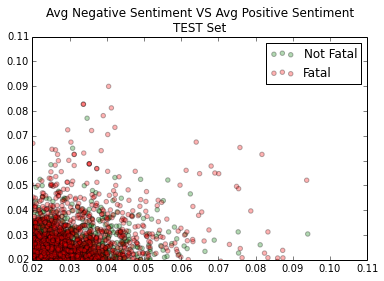
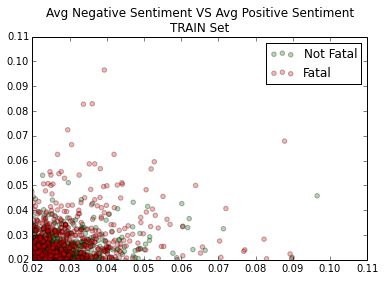


Table 14 lists the linguistic features extracted from all accidents in the train and test partitions along with their mean values. As all of the linguistic features were numerical and clearly correlated, principal components analysis was indicated. A PCA model was fit to the scaled predictor variables of the train set and evaluated on the test set. Figure 21 depicts a scree plot of the principal components. Figure 22 is a plot of the first and second principal components with accident outcome overlay. Similar to measures of sentiment, the principal components derived from this feature set were not expected to be strong predictors of accident outcome. The final Linguistic feature set for modeling was created with the first 10 principal components. The original variables were omitted.

Table 14: Linguistic Feature Set Statistics



Figure 21: Principal Components Scree Plot

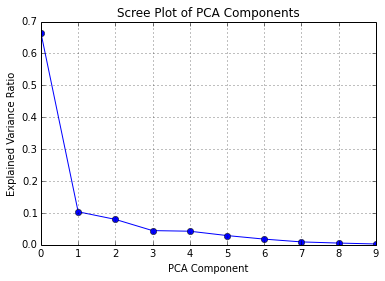
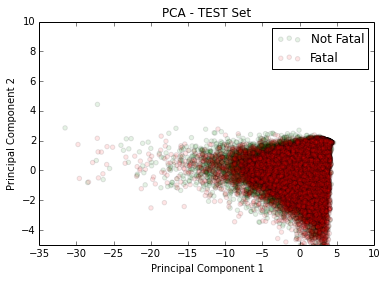
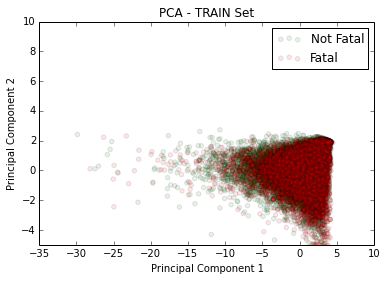


Figure 22: Scatterplot of Top Two Principal Components with Accident Outcome Overlay



Each of the five classification algorithms was trained on the train partition data of the Linguistic feature set and evaluated on the test partition data. Model evaluation statistics for the base and combination models are summarized in Table 15.

The Logistic Regression model was again the top performer with a classification accuracy score of 64.0%, a dismal 21.3% drop in accuracy from the baseline structured data only model. The combination models did not provide additional lift for this feature set. A cumulative gains chart for each of the five classification models is depicted in Figure 24. Accuracy scores at different thresholds for the combination models are plotted in Figure 23. Figure 25 is a plot of feature importance for the top 20 features of the Random Forest model.

Table 15: Base Model and Combination Model Results



Figure 23: Combination Model Results

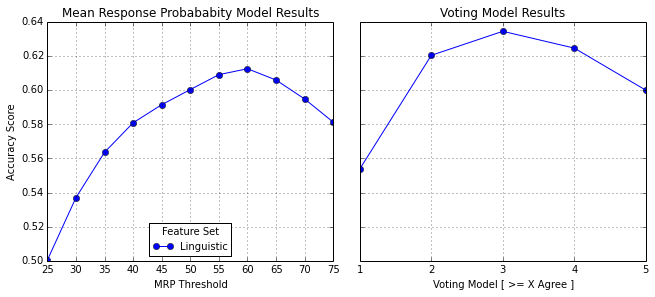
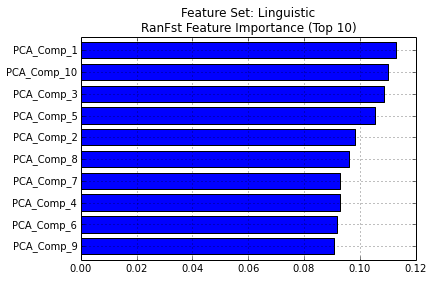


Figure 24: Base Classification Model Cumulative Gains Chart



Figure 25: Random Forest Classification Model Feature Importance



**Topic Feature Set**

*Section IPython Notebook link:* [*osha\_05\_topic\_feature\_set.ipynb*](http://nbviewer.ipython.org/7535447)

For this feature set a Python implementation of Latent Dirichlet Allocation from the Gensim package was used to develop a topic model based on the accident summary content of each accident in the training set. Unlike unsupervised clustering where each observation is assigned to one cluster, LDA assigns each text observation to a small set of groups, or topics. Furthermore, the degree of membership in each topic is weighted, with a large weight indicating strong membership, and vice versa. LDA does not require a priori knowledge of topics and configures topics based on a supplied parameter.

The topic model is a sparse model where many topics define the overall corpus but only a few are assigned to each text. The topics are multinomial distributions over words, with each topic giving each word in the corpus a probability. Words with higher probability are more associated with certain topics than words with lower probability (Richert, pg. 79). Topic composition can be summarized by the most highly weighted words associated with it. Accident summaries can be compared in topic space by creating an accident-topic matrix with each accident as a row, each topic as a column, and topic weights as cell values. Two accident summaries are similar if they refer to the same topics. For purposes of this project, the desired effect was that some topics would be more associated with fatal accidents than non-fatal accidents and this difference would enable classification algorithms to better discriminate between the two. As the number of topics was far less than the number of words that comprised the corpus of all accident summaries another benefit of the topic space model was dimensionality reduction.

Prior to fitting a topic model with 200 topics to the train partition data the accident summary text was pre-processed. NLTK was employed to exclude stopwords, stem words, remove punctuation, and retain words with all alphabetical characters only. A histogram of the number of resulting topics per accident is depicted in Figure 26. A large percentage of accidents had 6 to 8 topics a piece with no accident having more than 35 topics. Figure 27 depicts the proportion of accident outcomes for the first 25 topics. Based on the variance in proportions of fatality exhibited across the first 25 topics the Topic model feature set was expected to add predictive power during the modeling stage.

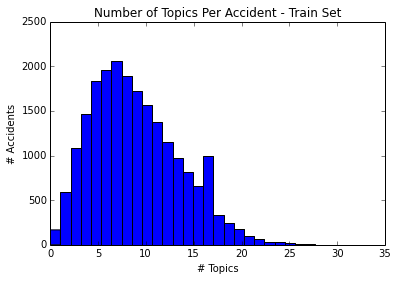
Figure 26: Histogram of the Number of Topics Per Accident

Table 16 depicts an example accident’s summary along with the top 7 topic memberships and weights. The overall topic composition for the top 7 topics is included to aid interpretation. Not all content words from the accident summary are matched with the “highest probability” words that comprise the topic and vice versa. But one can see why the accident was assigned to most of the topics nonetheless. What might seem like a weakness is actually a strength, as accidents that allude to similar concepts with different wording can converge on the same topics and allow models to detect similarities across text from separate observations in cases where the bag-of-words model would fail.

Figure 27: Proportion of Fatal Accidents for the First 25 Topics

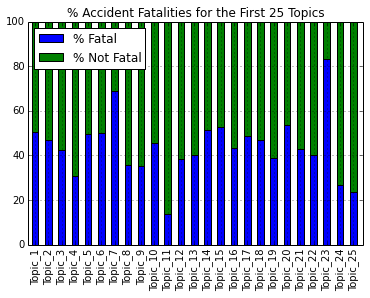


Table 16: Example Accident Topic Membership and Weight 

Each of the five classification algorithms was trained on the train partition data of the Topic feature set and evaluated on the test partition data. Model evaluation statistics for the base and combination models are summarized in Table 17.

The Logistic Regression model was again the top performer with a classification accuracy score of 70.9%, a significant drop in accuracy from the baseline structured data only model. The 3+ voting model earned a slight 0.3% improvement in accuracy over the top performing Topic base model. The MRP models did not provide additional lift for this feature set. A cumulative gains chart for each of the five classification models is depicted in Figure 28. Accuracy scores at different thresholds for the combination models are plotted in Figure 29. Figures 30 and 31 are plots of feature importance for the top 20 features of the Random Forest and Logistic Regression models, respectively.

Table 17: Base Model and Combination Model Results



Figure 28: Base Classification Model Cumulative Gains Chart

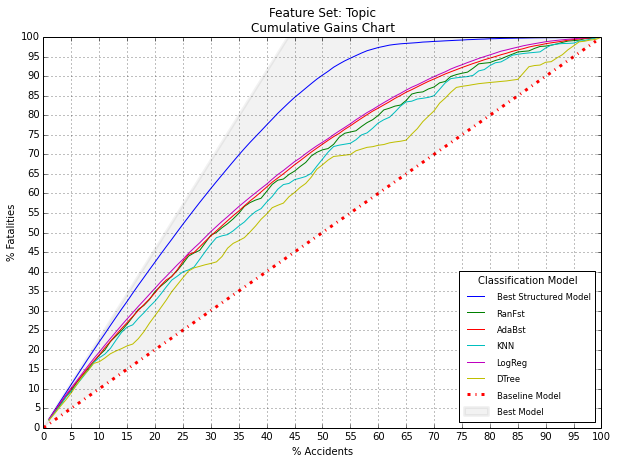


Figure 29: Combination Model Results

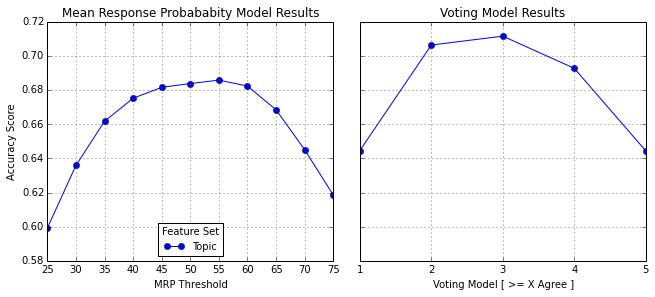


Figure 30: Random Forest Classification Model Feature Importance

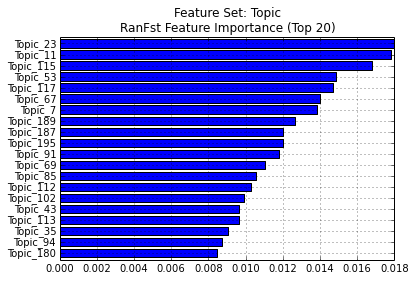
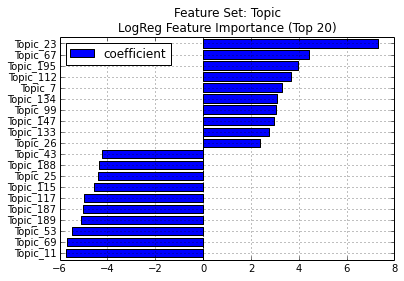


Figure 31: Logistic Regression Classification Model Feature Importance



**Description SVD Feature Set**

*Section IPython Notebook link:* [*osha\_06\_svd\_description\_feature\_set.ipynb*](http://nbviewer.ipython.org/7535448)

Four feature sets were created by applying singular value decomposition (SVD) to vectorized matrices of terms contained in the Event Description field. Because additional steps were taken to apply TF-IDF transformation to the vectorized document-term matrices the technique in this context is known as Latent Semantic Analysis. According to Wikipedia, “Latent semantic analysis (LSA) is a technique in natural language processing, in particular in vectorial semantics, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. LSA assumes that words that are close in meaning will occur in similar pieces of text.”

There are various approaches for pre-processing raw text prior to vectorization. Table 18 below summarizes the four different configurations experimented with in this project. The four feature sets created from application of SVD to each of these vectorized document-term matrices were used during modeling to allow results to pass judgment on the optimal configuration.

Table 18: Vectorization Parameters and Resulting SVD Feature Sets



Singular value decomposition is a dimensionality reduction technique and also a feature extraction technique. Like principal components analysis, SVD constructs a linear combination of variables, a small number of which contain the majority of all information resident in the original variables. When used as an intermediate step prior to predictive modeling it is customary to retain many components, as the models themselves will discern which components are useful and which are not. Like PCA, the components are uncorrelated so multicollinearity is not an issue. An arbitrary decision was made to retain the first 50 components. As indicated in Table 18 the largest document-term vector prior to application of SVD contained 3,567 features. A matrix of features this size is unmanageable, but becomes much more manageable and informative after application of SVD.

SVD can be thought of as a process that uncovers “latent dimensions of meaning” in the linear combinations of input variables. Each document, an accident description in this case, is assigned a score, or weight, with each component. The document score for each component denotes that documents influence on the component. Weights range from negative to neutral to positive, indicating the sign of the relationship. The anticipated effect here is that some components will be associated with a higher likelihood of fatality and that classification algorithms will be able to better discriminate between accident outcomes based on the document-component scores.

Figures E.1 and E.2 in Appendix E are scatterplots of the top four components containing the majority of information resident in the underlying document-term matrices. Note that the plots are remarkably similar across the train and test partitions, especially as the SVD model was fit to the train partition data and subsequently used to transform the test partition data into component space. The overlay of accident outcome, with red indicating fatality, on each component-to-component scatterplot, helps show why the SVD feature sets are expected to be strong predictors of accident outcome. If all the documents’ scores were clustered together around the point of origin, or if document scores from accidents with different outcomes were plotted together away from the point of origin, the components would not be expected to be strong predictors of accident outcome. Further analysis of document clusters farthest away from the point of origin may uncover additional insight, and could be an interesting area of further research.

Each of the five classification algorithms was trained on the train partition data of the Description SVD feature set and evaluated on the test partition data. Of the four feature sets created, the *desc\_stem\_n1* feature set was selected to go forward with. Model evaluation statistics for the base and combination models are summarized in Table 19.

Table 19: Base Model and Combination Model Results



The Logistic Regression model was once again the top performer with a classification accuracy score of 90.0%, a 4.7% improvement in accuracy from the baseline structured data only model. The best MRP combination model provided an additional lift of 0.6% for this feature set. The best voting combination model provided an additional lift of 0.8%. A cumulative gains chart for each of the five classification models is depicted in Figure 32. Accuracy scores at different thresholds for the combination models are plotted in Figure 33. Figures 34 and 35 are plots of feature importance for the top 20 features of the Random Forest and Logistic Regression models, respectively.

Figure 32: Base Classification Model Cumulative Gains Chart

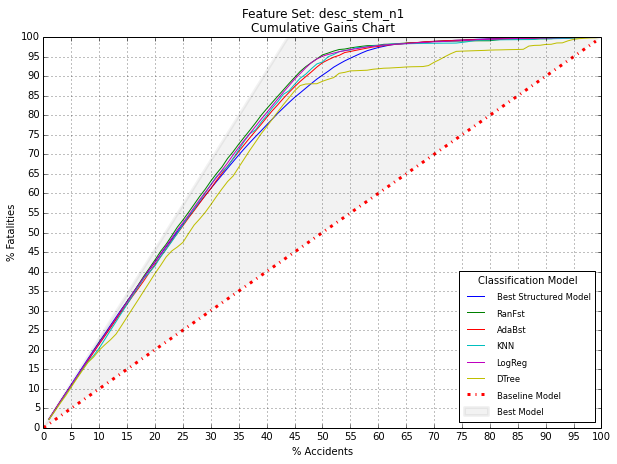


Figure 33: Combination Model Results

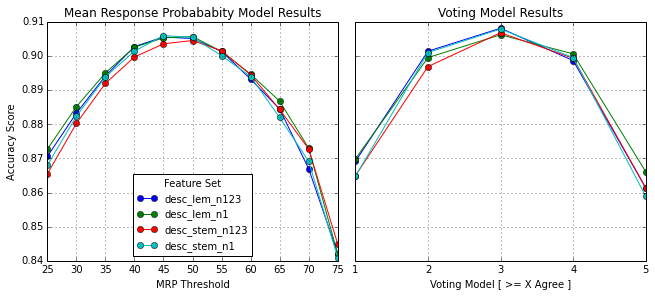


Figure 34: Random Forest Classification Model Feature Importance

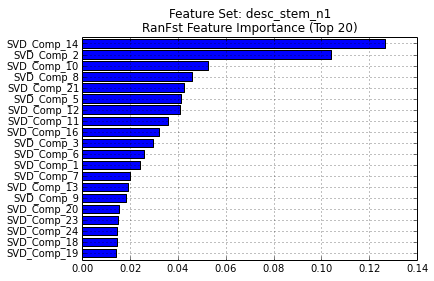
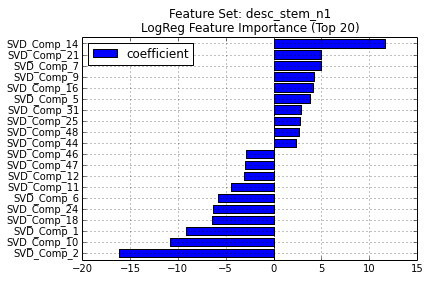


Figure 35: Logistic Regression Classification Model Feature Importance



**Summary SVD Feature Set**

*Section IPython Notebook link:* [*osha\_07\_svd\_summary\_feature\_set.ipynb*](http://nbviewer.ipython.org/7535453)

Similar to the process followed in the previous section for the Event Description field, applying singular value decomposition to vectorized matrices of terms contained in the Summary Text field created four feature sets. Table 20 below summarizes the four different configurations experimented with in this project. The four feature sets created from application of SVD to each of these vectorized document-term matrices were used during modeling to allow results to pass judgment on the optimal configuration.

Table 20: Vectorization Parameters and Resulting SVD Feature Sets

****

Figures F.1 and F.2 in Appendix F are scatterplots of the top four components containing the majority of information resident in the underlying document-term matrices. Note that, like the Description SVD, the Summary SVD plots are remarkably similar across the train and test partitions. However, the document score plots are more blurred and not as distinctive. This is likely due to the much larger variety of words and concepts contained in the summary text. These Summary SVD feature sets are expected to be weaker predictors of accident outcome then those from the Description SVD feature sets, but still predictive nonetheless. There are clear clusters of documents with similar accident outcomes in concept-to-concept space removed from the origin. Concepts contained in the Summary SVD features might complement the Description SVD features during the upcoming combined feature set modeling phase and provide a distinctive edge.

Each of the five classification algorithms was trained on the train partition data of the Summary SVD feature set and evaluated on the test partition data. Of the four feature sets created, the *summ\_stem\_n1* feature set was selected to go forward with. Model evaluation statistics for the base and combination models are summarized in Table 21.

Table 21: Base Model and Combination Model Results



The Logistic Regression model was once again the top performer with classification accuracy score of 83.7%, a slight 1.6% drop in accuracy from the baseline structured data only model. The combination models did not provide additional lift for this feature set. A cumulative gains chart for each of the five classification models is depicted in Figure 36. Accuracy scores at different thresholds for the combination models are plotted in Figure 37. Figures 38 and 39 are plots of feature importance for the top 20 features of the Random Forest and Logistic Regression models, respectively.

Figure 36: Base Classification Model Cumulative Gains Chart

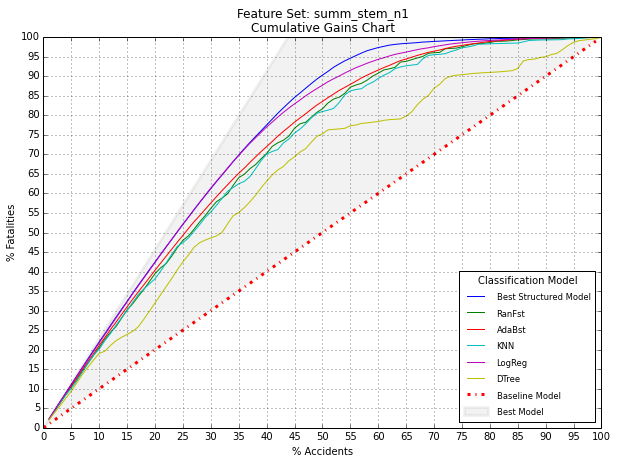


Figure 37: Combination Model Results

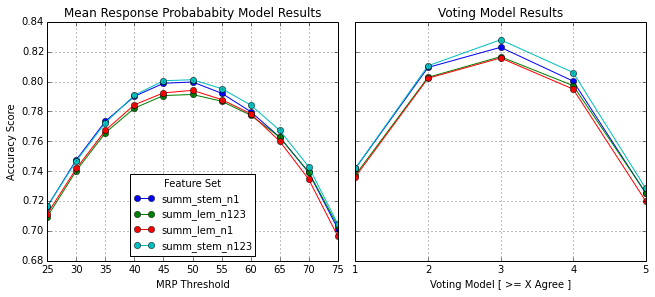


Figure 38: Random Forest Classification Model Feature Importance

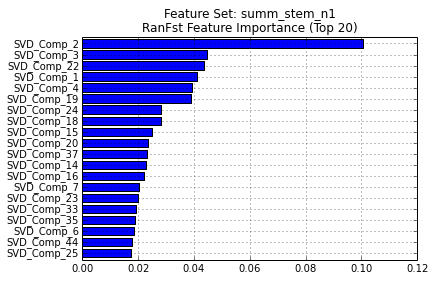


Figure 39: Logistic Regression Classification Model Feature Importance



**Combined Feature Set**

*Section IPython Notebook link:* [*osha\_08\_combined\_feature\_set.ipynb*](http://nbviewer.ipython.org/7535457)

In this final section combined feature sets comprised of top predictors selected across the various text-based and structured data feature sets were created and used as input to the five base classification models. The goal was to create synergies by joining together the best-of-predictors into combined feature sets and achieve potentially greater gains in classification accuracy. Combined feature sets were created based on various predictors selected from the Structured, Keyword, Linguistic, Topic, Description SVD and Summary SVD feature sets. Two feature selection methods and two methods of applying the feature selection method to the underlying feature sets were employed. Table 22 lists the 28 combined feature sets and the logic used in their construction.

Each of the five classification algorithms was trained on the train partition data of each combined feature set and evaluated on the test partition data. Five classifiers trained on 28 feature sets resulted in 140 distinct classification models. Model evaluation statistics for the base and combination models from selected top performing combined feature sets are summarized in Table 23. The Logistic Regression model was once again the top performer with classification accuracy score of 94.0%, a 4% gain in accuracy from the top performing standalone feature set model, and a 8.7% gain in accuracy from the baseline structured data only model. The combination models did not provide additional lift. A cumulative gains chart for each of the five classification models of the top performing combined feature set is depicted in Figure 41. Accuracy scores at different thresholds for the selected combination models are plotted in Figure 40. Figures 42, 43 and 44 are plots of feature importance for the top 20 features of the top performing combined model and two other combined models.

Table 22: Configuring the Combined Feature Sets



Table 23: Base Model and Combination Model Results



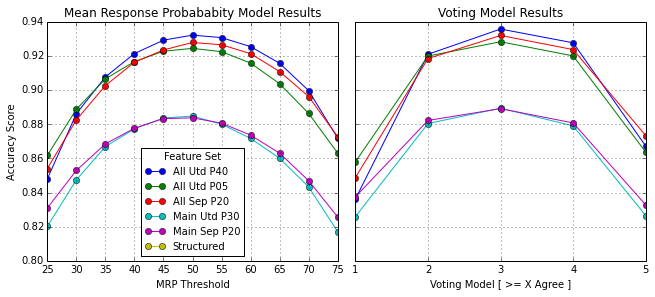
Figure 40: Combination Model Results

Figure 41: Base Classification Model Cumulative Gains Chart

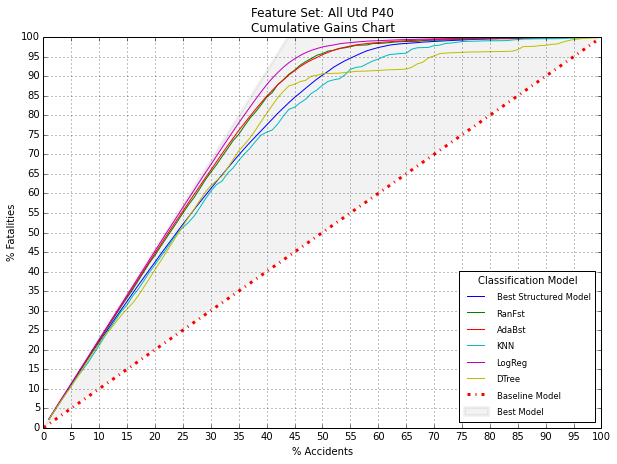


Figure 42: Classification Model Feature Importance of Top Performing Combined Model

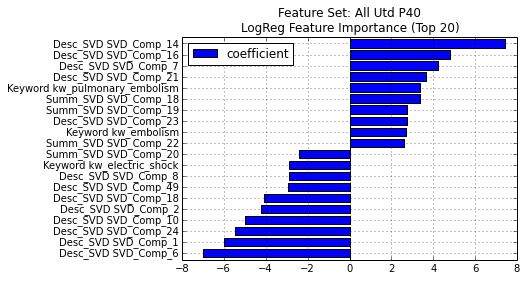


Figure 43: Logistic Regression Feature Importance of a Selected Combined Model

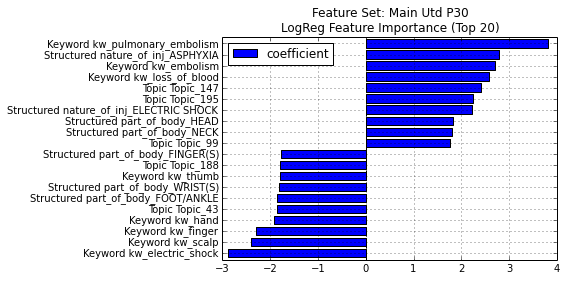
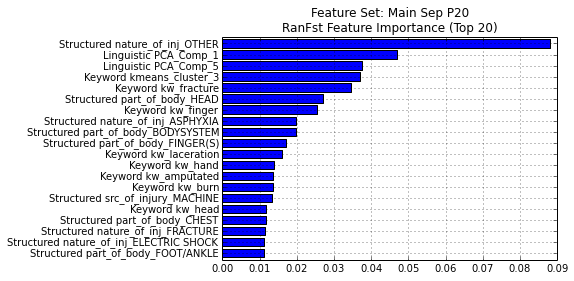


Figure 44: Random Forest Feature Importance of a Selected Combined Model



**CONCLUSION**

This thesis demonstrated that features mined from text-based attributes captured concepts and information not present in the structured data attributes and that this infusion of new information enabled classification algorithms to better discriminate between target variable outcomes, thereby improving model accuracy. Feature sets created from a medley of statistics-based and linguistics-based text mining techniques resulted in a measurable improvement in classification model accuracy in some cases beyond that of models trained on structured data attributes only. This thesis also demonstrated that combining multiple predictive models trained on the same features set obtained better performance than was obtained from any of the constituent models independently. Finally, combined feature sets composed of top predictors selected across the various text-based and structured data feature sets provided the greatest lift to classification model accuracy.

**FURTHER RESEARCH**

This analysis placed equal importance on the classification of accident outcomes and evaluated models based on their overall accuracy rate. The misclassification cost ratio is the ratio of false positive misclassification costs to false negative misclassification costs, which was implicitly one for this project. However, classifying a true-death as non-death should be more costly. Future work could investigate the effect of asymmetric misclassification costs on classifier results. Correct application of misclassification costs would likely decrease false negative outcomes at the expense of increased false positive outcomes, an acceptable tradeoff.

Recall that the raw data contained a trinary variable with injury outcomes of fatal, hospitalized and non-hospitalized. This trichotomous variable was omitted in favor of a dichotomous target variable with values of fatal or non-fatal. Development of a trinary classification model based on this trinary target, with application of trinary evaluation measures for model goodness, such as sensitivity, is another area ripe for future work.

Assumptions and decisions made during the data preparation phase dictated the quality of feature sets used to train classification models and exerted considerable influence on predictive performance. The breadth and depth of this project reduced feature extraction and selection, two critical components of any predictive modeling project, to arbitrary decisions at certain steps in the analytical pipeline. Examples are the selection of 50 singular value decomposition components and four K-Means clusters, text pre-processing procedures, simplified sentiment computations, exclusion of structured accident attributes and default classification algorithm parameters. The only way to thoroughly test whether assumptions and decisions made throughout this analysis were optimal would be to evaluate the impact of each unique path of decisions against some common metric, such as model accuracy on the test set. Of course this is not feasible, as the number of paths to evaluate would grow exponentially with each decision made. With time, dedication and a flexible framework, running iterations of the entire project from start to end along multiple decision paths would help determine optimal thresholds and make important decisions less arbitrary and more scientific.

This author believes that the Python scientific computing environment is an ideal framework for programming such iterations over complex analytical pipelines, and that the publically available project code, and the powerful open source Python capabilities that this project is built upon, offer an ideal foundation from which to expand and manage analytical complexity of this nature.

Furthermore, default classification model configurations were used in this study as the modeling framework and feature engineering tasks were complex enough. Time spent experimenting with different parameters and discovering optimal configurations may be worth the effort. In a similar manner the five classification models used in this study were selected in part for their speed. Experimenting with different classifiers, including those with longer run times, may improve results beyond that which was achieved in this project.

**REFERENCES**

Bilisoly, Roger. *Practical Text Mining with Perl*. Hoboken, NJ: Wiley, 2008. Print.

Bird, Steven, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. Beijing: O'Reilly, 2009. Print.

Esuli, A.; Sebastiani, F., SentiWordNet: A publicly available lexical resource for opinion mining Proceedings of LREC-06, 5th Conference on Language Resources and Evaluation, Genova, IT, 2006.

Fernando Pérez, Brian E. Granger, *IPython: A System for Interactive Scientific Computing*, Computing in Science and Engineering, vol. 9, no. 3, pp. 21-29, May/June 2007, doi:10.1109/MCSE.2007.53. URL: http://ipython.org.

Harrington, Peter. (2012). *Machine Learning In Action*. Shelter Island, New York: Manning Publications, 2012. Print.

Larose, Daniel T. *Data Mining: Methods and Models*. Hoboken, New Jersey,: John Wiley and Sons, 2006. Print.

Larose, Daniel T. *Discovering Knowledge in Data an Introduction to Data Mining*. Hoboken, NJ: Wiley-Interscience, 2005. Print.

McKinney, Wes. (2013). *Python for Data Analysis.* Sebastopol, CA: O’Rielly Media, Inc., 2012. Print.

Miner, Gary. *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. Waltham, MA: Academic, 2012. Print.

Occupational Safety and Health Administration. (2013). *All About OSHA*. Retrieved from

<https://www.osha.gov/Publications/all_about_OSHA.pdf>.

Pedregosa et al. Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011.

Richert, W., & Coelho, L. P. (2013). Building Machine Learning Systems with Python. Birmingham: Packt Publishing.

G. van Rossum et al., Python Language Website, [http://www.python.org/](http://www.python.org/" \t "_blank).

United States Department of Labor. (2013). *OSHA Enforcement Data* [Data files]. Retrieved from <http://ogesdw.dol.gov/data_catalogs>.

Weiss, Sholom M., Nitin Indurkhya, and Tong Zhang. *Fundamentals of Predictive Text Mining*. London: Springer-Verlag, 2010. Print.

Weiss, S. M. *Text Mining Predictive Methods for Analyzing Unstructured Information.* New York: Springer, 2005. Print.

**APPENDIX A: Python Scientific Computing Resources and Packages**

|  |  |  |  |
| --- | --- | --- | --- |
| Resource | | URL | Description |
| Python | Programming Language | http://www.python.org | Powerful dynamic programming language that is used in a wide variety of application domains |
| IPython / IPython Notebook | Interactive Computing | <http://ipython.org>  <http://ipython.org/notebook> | Web-based interactive computational environment that combines code execution, text, mathematics, plots and rich media into a single document. |
| Pandas | Python for Data Analysis | <http://pandas.pydata.org> | High-performance, easy-to-use data structures and data analysis tools for the Python programming language. |
| Scikit-Learn | Machine Learning in Python | <http://scikit-learn.org/stable/> | Simple and efficient tools for data mining and data analysis; Built on NumPy, SciPy, and Matplotlib. |
| NLTK | Natural Language Toolkit | <http://nltk.org> | Leading platform for building Python programs to work with human language data. |
| Gensim | Topic Modeling | <http://radimrehurek.com/gensim/> | Scalable statistical semantics and analysis of documents for semantic structure. |
| Matplotlib | Visualization and Graphing | <http://matplotlib.org> | Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms |
| SciPy | Mathematical Algorithms and Functions | <http://www.scipy.org> | Python-based ecosystem of open-source software for mathematics, science, and engineering |
| NumPy | Base N-Dimensional Arrays | <http://www.numpy.org> | Fundamental package for scientific computing with Python |

**APPENDIX B: Keyword Feature Set Additional Figures**

Figure B.1

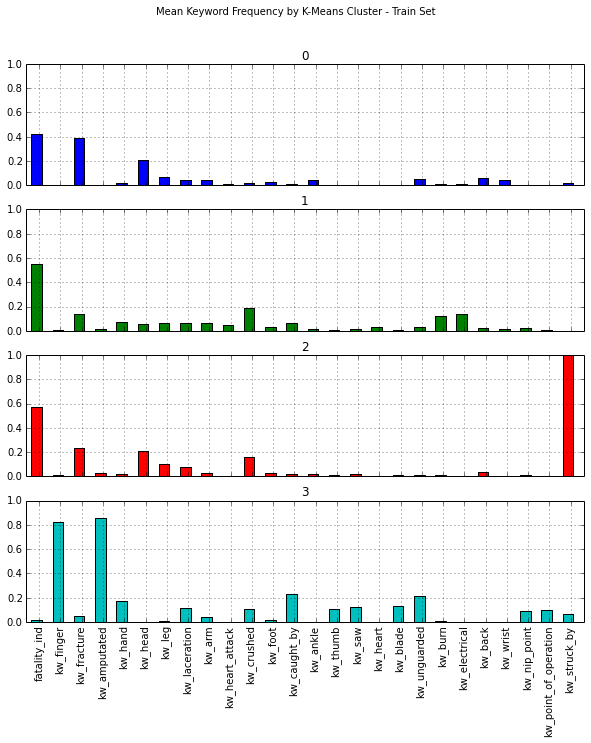
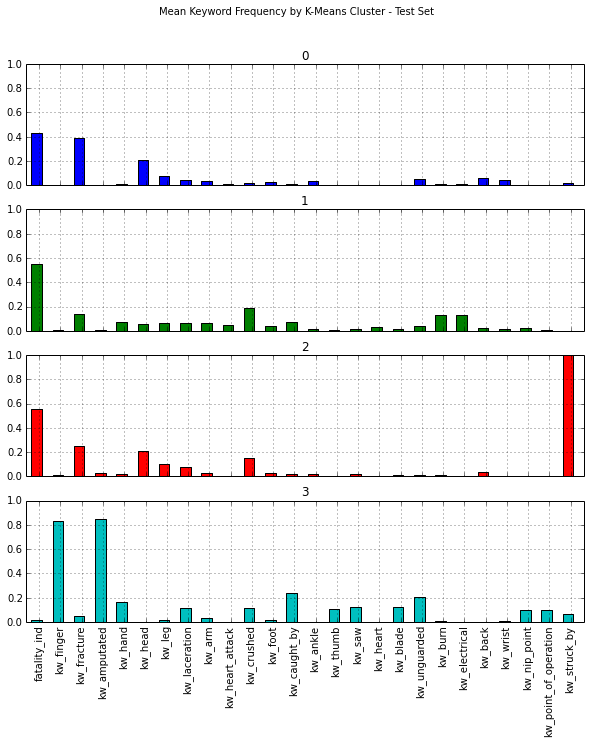
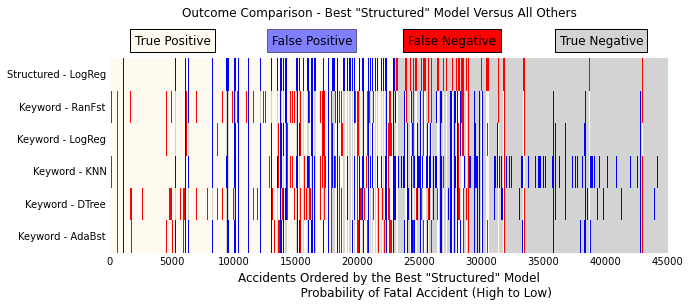
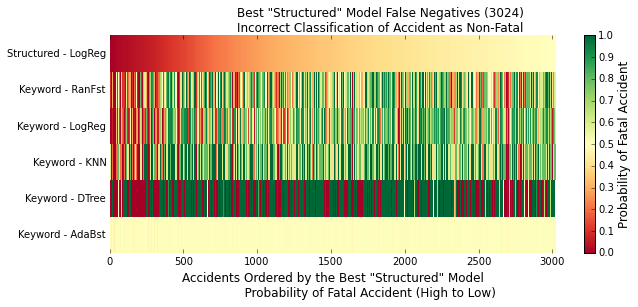
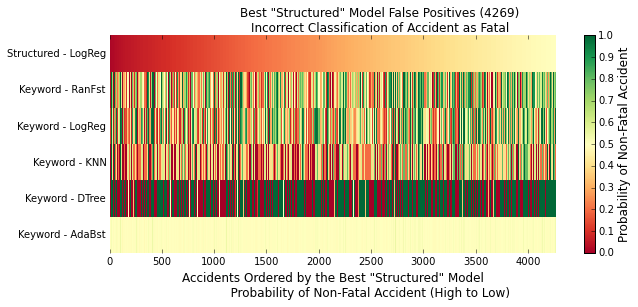


Figure B.2

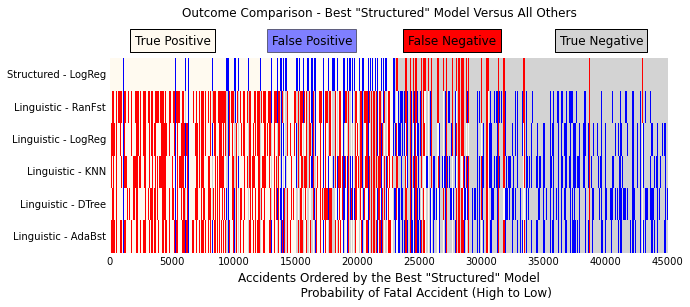


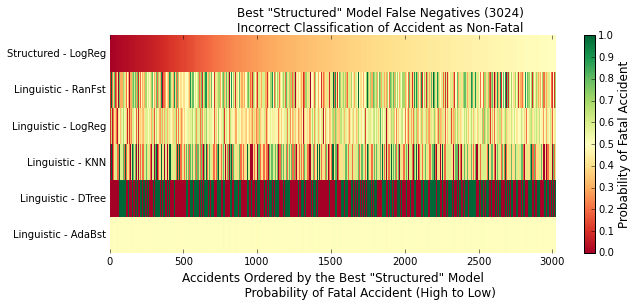


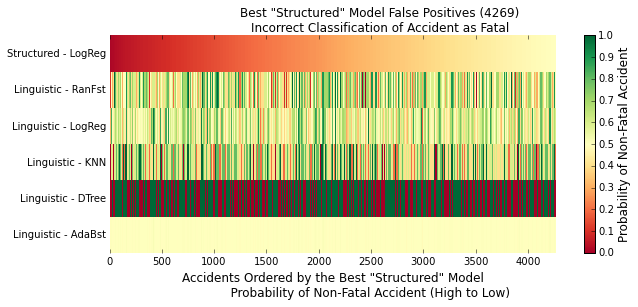




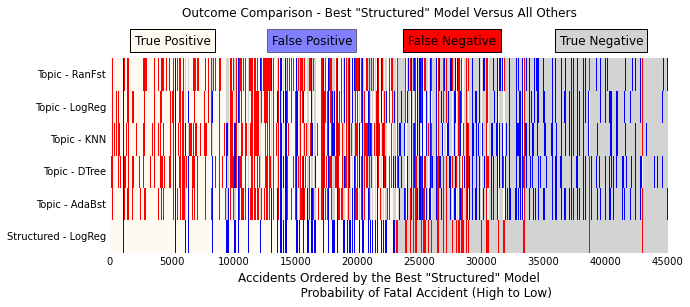
**APPENDIX C: Linguistic Feature Set Additional Figures**

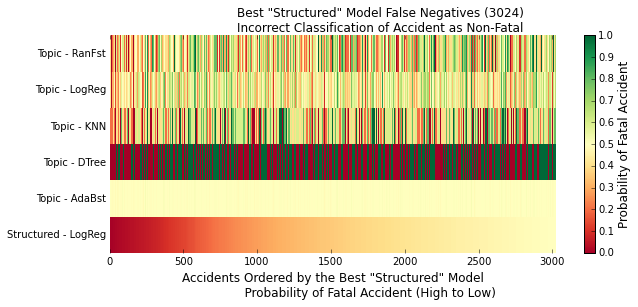


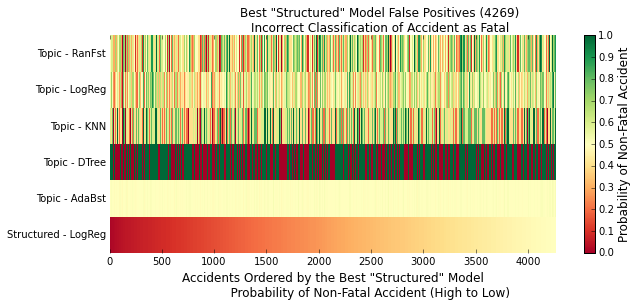




**APPENDIX D: Topic Feature Set Additional Figures**







**APPENDIX E: Description SVD Feature Set Additional Figures**

Figure E.1

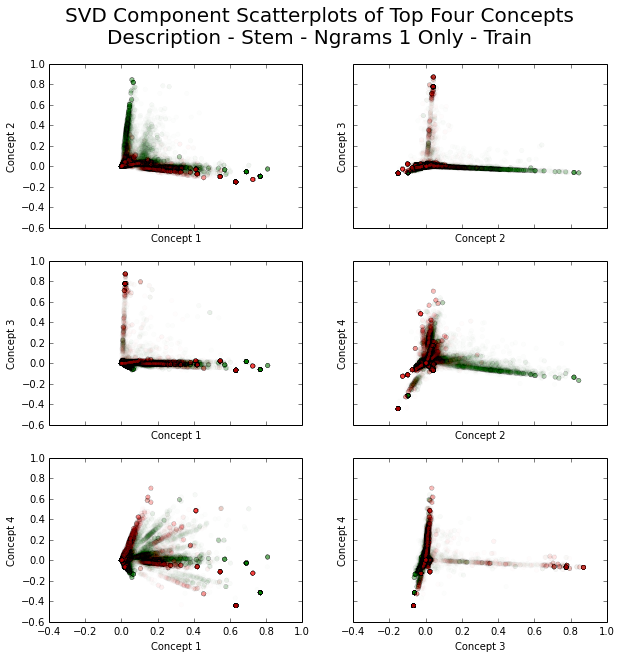
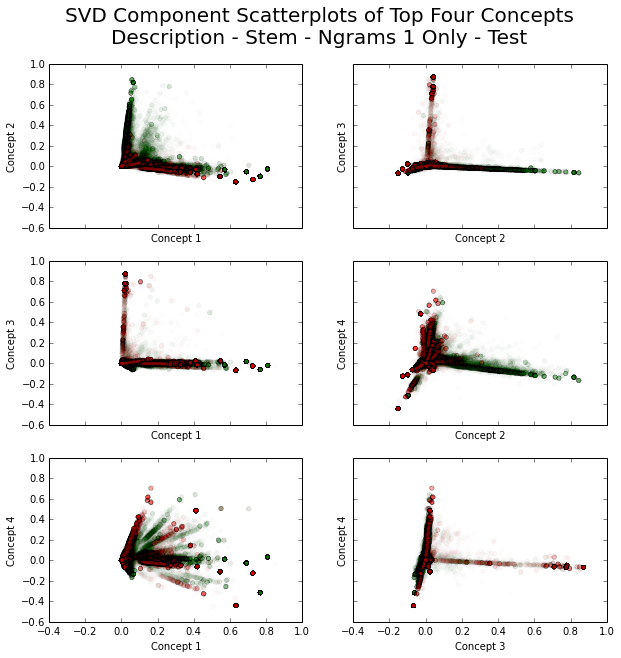
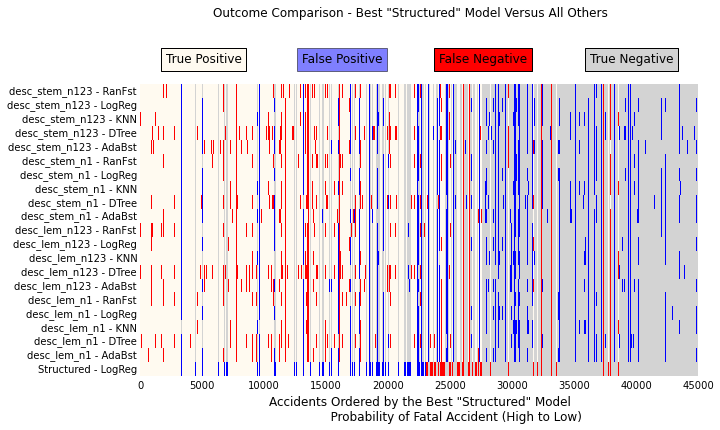
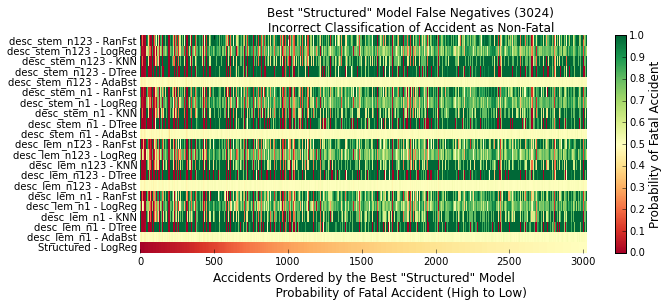
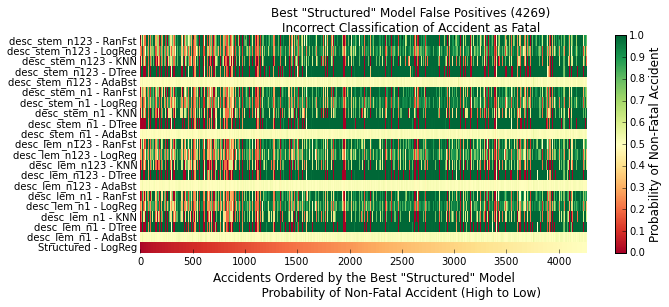


Figure E.2







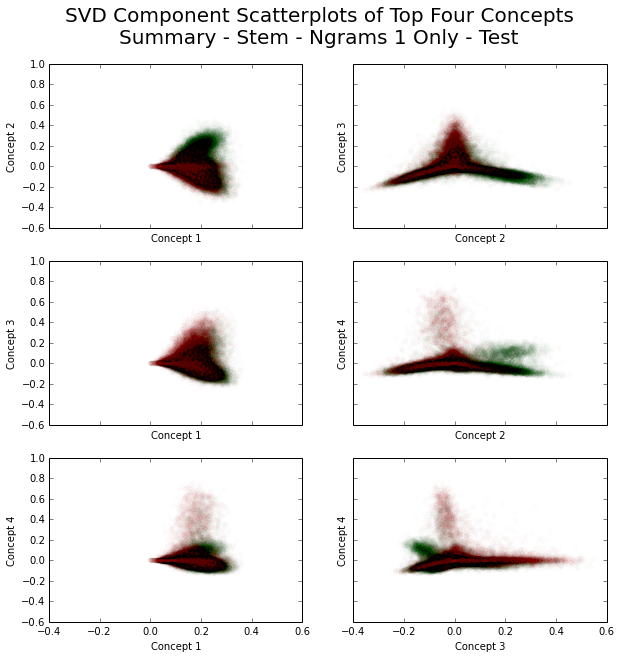


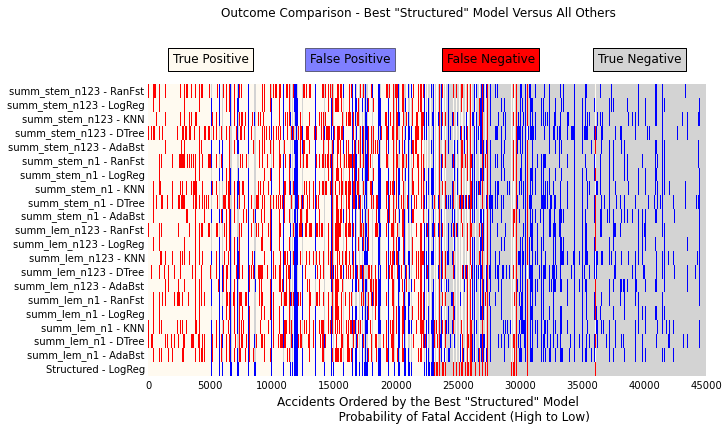
**APPENDIX F: Summary SVD Feature Set Additional Figures**

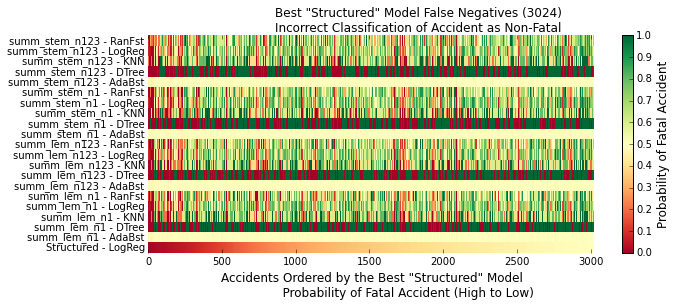
Figure F.1

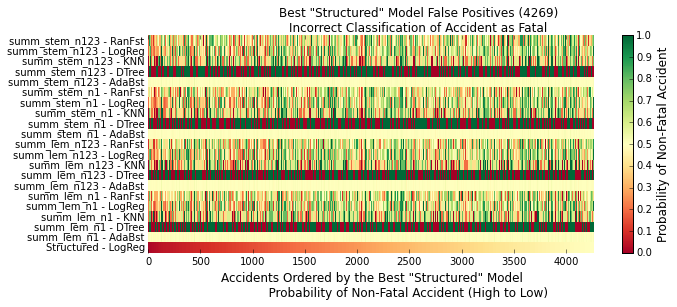


Figure F.2









**APPENDIX G: Combined Feature Set Additional Figures**

