Predicting Human Cost with Toronto Fire Incident Data

Introduction

Fires have plagued Toronto since its founding. Two great fires, in 1849 and 1904, destroyed large sections of the city. While Toronto's built environment has changed significantly since these calamities, and firefighting technologies have likewise improved, fires remain one of the city's foremost and least predictable public safety concerns. In 2017, there were approximately 1,753 fires in Toronto which caused roughly $77,320,995 in damage, temporarily or permanently displaced an estimated 17,837 people from their homes, and killed 145 civilians and 18 firefighters.

Is it possible to predict the fires with the greatest human cost? I will attempt to answer this question by leveraging machine learning algorithms and the City of Toronto's fire incident data. Though Toronto Fires Services and the Ontario Fire Marshal undoubtedly have many metrics for severity and risk, for this study I am interested in predicting fires with at least one human fatality (civilian or firefighter), or where at least one civilian was rescued by firefighters.

After data cleaning and univariate, bivariate and multivariate exploratory analyses, features will be selected after a comparison of three different methods: Information Gain, Recursive Feature Elimination, and LASSO. Following other recent machine learning studies in fire prevention, I will train and test models using the following algorithms: Decision Tree, Logistic Regression, Random Forest, AdaBoost, and an ensemble model of the last three. Decision Tree serves as a baseline model, while Logistic Regression, Random Forest and AdaBoost were among the best performing algorithms in studies that leveraged similar fire incident data from other locales. With k-fold cross-validation not being applicable to time-series data, I will instead use the "sliding window" time-partitioned approach to train and test the model.

Literature Review

As Moshashaei and Alizadeh's survey reveals, much of the work on fire risk prevention has targeted not urban areas, but wildfires and forest fires.[1] There have been few works pertaining to fire incident data in urban areas, though this area of study seems to recently be attracting researchers' attention with a series of papers published in 2018 and 2019 that leverage urban fire data.

Some American fire departments, mostly notably in New York and Atlanta, have leveraged fire incident data in the service of predicting which buildings were most at risk of serious fires. While New York has not disclosed details about its model[2], in Atlanta researchers used machine learning algorithms such as Support Vector Machine and Random Forest to great effect. Their SVM model predicted 71.36% of fires in their test data at a false positive rate of 20%, and was adopted by Atlanta's fire authorities as a potentially lifesaving tool.[3]

Walia et al. (2018) built on the Atlanta research, using fire incident data from Pittsburgh along with data about property features and property inspections and violations (e.g. sanitation, noise, gas leaks, etc.) to develop a predictive risk framework for commercial and residential properties.[4] The researchers used machine learning algorithms such as Logistic Regression, Ada Boost, Random Forest, XG Boost, and KDD16 Firebird, the model used in the aforementioned Atlanta study. Their commercial property model, which employs XG Boost, "accurately predicts nearly half of the fires, 70 times more effective than random guess, which would be correct 0.71% of the time, given the distribution of fire incidents."[5] Their residential property model, using Random Forest, achieved even better results, largely because of the more equal class balance in the fire incident data.

In the same vein, Dang, Cheng, Mann, Hawick and Li (2019) developed a fire prediction model for commercial properties in the Humberside region of the United Kingdom.[6] In addition to fire incident data, the researchers also used a range of data from other sources, such as commercial property data, business inspection data, and food hygiene ratings. They build models with four machine learning algorithms, including Adaptive Boosting, Extreme Gradient Boosting, Random Forest, and Multilayer Perceptron. The best performing model, which utilized AdaBoost, "has ability to identify the high-risk properties having above 70% chance to catch fire while the highest risk level of those currently used by HFRS has more than 4000 properties with the chance to catch fires at about 3.8%."[7]

Pirklbauer and Findling (2019) analyzed two years of fire data from Upper Austria joined with weather data to predict three major categories of fire department operation (fire, storm, and "person").[8] Pirklbauer and Findling utilized and compared machine learning algorithms such as k-Nearest Neighbours, Linear Discriminant Analysis, Decision Trees, Random Forest, and Support Vector Machine. Their SVM and RF models achieved accuracies of 61%, nearly double the accuracy of their best baseline model.

Wang, et al. (2019), using fire data from Zhengzhou, China, climate data, and data on e-commerce orders, also seek to forecast fire risk using a complex machine learning model called NeuroFire, which "utilizes GRU (Great Recurrent Unit) to learn temporal representations of urban data and integrate the temporal representations into fire risk sequences by CRF (Conditional Random Field)." [9] NeuroFire significantly outperformed 9 different baselines, including algorithms such as Logistic Regression, LASSO, Support Vector Machine, DeepST.

To the best of my knowledge, the only fire prediction study in a Canadian context comes from British Columbia, though this approach did not leverage machine learning techniques. Instead, this study was comprised of "a risk-based, data-driven framework for redesigning fire safety inspections," with risk scores being assigned to properties on the basis of "information about previous inspections performance, the responsible person in charge of the property, the property use, and the structure type."[10]

The final study that I want to spotlight is Anders Ohrn's two-part analysis of Toronto fire data in the online publication *Towards Data Science*.[11] While not an academic study, Ohrn's is the only published work that I am aware of that leverages the same data as my own study, though Ohrn's study does not propose a predictive model. In the second part of his study, Ohrn focuses on what he calls "extreme events." Crucially, he grapples with two problems that I have likewise had to navigate in this study. First, he notes that the worst fires by definition "are rare events and therefore in any finite data collection they are poorly sampled." Secondly, he articulates the complexities of quantifying severity in the context of the fire data. As neither Toronto Fire Services nor the Ontario Fire Marshal defines what constitutes a severe fire, it is necessarily for researchers interesting in exploring extreme fire incidents to establish their own "proxy metrics" of severity.[12] Ohrn elected to use the number of dispatched units during the course of the fire incident, though he notes at the end of his study that this definition biases fires of extended duration that require multiple units to come and go. I have opted for a different metric, and will focus this study on those fires where human lives are stake: fires where either a civilian or a firefighter dies, or fires where at least one person required rescue by the TFS.

Dataset

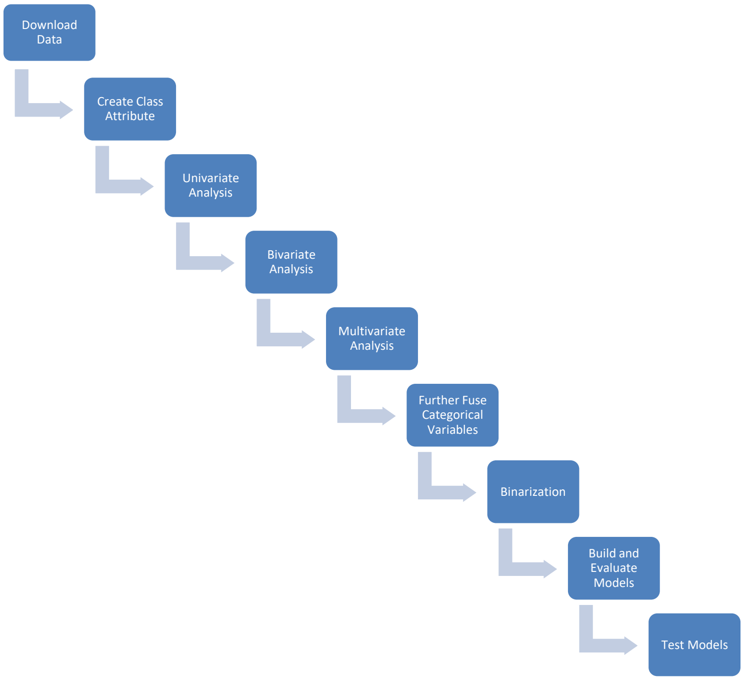
The dataset was procured from the City of Toronto's Open Data portal. It can be found here: https://open.toronto.ca/dataset/fire-services-incident-data/

The dataset as initially downloaded contained 43 attributes. The class attribute was not a part of the dataset, and was instead derived from three attributes: Civilian Casualties, Firefighter Casualties, and Count of Persons Rescued. Of first these 44 attributes, 27 were categorical, 9 were numerical, 5 were datetime, and 3 contained character data. The dataset contains over 12,000 records, covering the period spanning from 2011 to 2017. This initial data together, together with the class variable, can be seen here: https://github.com/terrygitersos/CKME136/blob/master/Data/firedata_with_class.csv

The class attribute is quite imbalanced: only 8.1% of observations were characterized by high human risk, making this the minority class. This is a class imbalance of approximately 12:1: not quite a severe imbalance, but imbalanced enough to necessitate oversampling.

After the univariate and bivariate analyses were completed and extraneous attributes removed, there remained 33 attributes, including 30 categorical, two numeric, and one datetime. This dataset can be seen here: https://github.com/terrygitersos/CKME136/blob/master/Data/pre_binary.csv

Approach



The code for steps 1-6 can be found here: https://github.com/terrygitersos/CKME136/blob/master/data%20exploration.Rmd

Step 1: Download Data

More information about the data can be found above in the dataset description.

Step 2: Create class attribute

The method for this step is described above in the dataset description.

Step 3: Univariate Analysis

Analysis of Each Attribute

Each attribute was manually examined, one-by-one. In most cases a definition was provided, the attribute structure and summary data were scrutinized, and the distribution of the data was plotted. Data types were changed and datetime data was reformatted as necessary.

Reduce cardinality of categorical variables

Several categorical attributes were characterized by very high cardinality, reaching as high as 270 levels. To reduce the cardinality of these attributes, levels were grouped together using domain expertise and business logic. Several categorical variables had their cardinality reduced as per the groupings articulated in the Ontario Fire Marshal's Codes List, which can be found here: https://collections.ola.org/mon/2000/10298759.pdf. The levels in the Incident_Ward attribute, which consisted of the pre-2018 Toronto municipal ward numbers, were grouped together so as to recreate pre-2018 provincial electoral districts. Finally, the levels in the Initial_CAD_Event_Type attribute were combined as per the groupings articulated in a document provided to me by the City of Toronto's Kevin

Ku, which can be found here: https://github.com/terrygitersos/CKME136/blob/master/TFS%20Response%20Guidelines%20June%201st%202017.docx

Flag NA data as "Missing"

In 13 categorical attributes, approximately 27% of the observations were missing. Rather than risking data loss or the introduction of bias through data imputation, I chose to create an additional level, “Missing.” It was discovered during the multivariate analysis that most (but not all) of these missing observations were for fires that occurred in vehicles, so attributes pertaining to fire alarms, sprinkler systems, smoke detectors and housing structures appear truly to be not applicable in these cases. For the time being, I have left the data coded as “Missing.”

There were a small number of other missing observations scattered throughout the data. These were dealt with by imputing the median or mode, or, in the case of missing geographical data, by imputing data retrieved from Google Maps.

Create new temporal attributes

A total of nine new temporal attributes were created. Two of those attributes are numerical attributes containing ratio data: firefighting time, which measures the time (in seconds) between the firefighters’ arrival and the fire being declared under control; response time, which measures the time (in seconds) between the alarm being sounded and firefighters’ arrival on the scene.

Another five categorical attributes were extracted from the TFS_Alarm_Time attribute, isolating the year, month, day, weekday, and hour of the fire (using the 24-hour clock). From the newly created month and hour attributes, two additional temporal attributes were created: season (from month), which groups the data into levels of “winter,” “spring,” “summer” and “autumn”; and time_of_day (from hour), which groups the data into levels of “morning,” “afternoon” and “night.”

Removal of irrelevant and redundant attributes

Irrelevant and redundant attributes were removed at this stage. These included two ID fields; the attributes from which the class variable was derived; attributes containing values that are determined only after fires are extinguished and fully investigated, and thus have no relevance in a predictive model.

Treatment of Outliers

Numerical attributes were analyzed for outliers through boxplot visualizations and outlier tables. In almost all cases, the outliers were judged to be logical and accurate data (e.g. outliers in the firefighting_time attribute exist because some fires simply are more severe than the median and require more time and resources). Almost all outliers were retained for this reason.

A single row of data was deleted when one outlier in the response_time attribute was determined to be misrecorded data.

Step 4: Bivariate Analysis

Several pairs of attributes were analyzed and their relationships plotted. Especial attention was paid to geospatial and temporal analyses, e.g. plotting all fire incidents by longitude and latitude, total fires by ward and property use, number of fires by season and area of origin, number of fires by possible cause and day of the week, etc.

Removal of redundant attributes

More pruning of the dataset was undertaken at this point. Four out of five datetime attributes, made redundant after the creation of the nine new temporal attributes, were removed; the fifth, TFS_Alarm_Time, was retained as it may be required for time-partitioning the data. Latitude and longitude, which were retained only to map fire incidents in the bivariate analysis, were removed. A further geospatial attribute, Incident_Station_Area, was deemed redundant and removed.

Correlation analysis

A correlation analysis was conducted on the numerical variables in the dataset. Because none of these attributes were normally distributed and all contained outliers, the Spearman method was used. Features with strong correlations to one another were identified, and in all cases the attribute with the greater proportion of outliers was removed from the dataset.

Chi-square test of independence

Because the relationship between categorical variables cannot be calculated with a correlation coefficient, a chi-square test of independence was used to assess whether or not categorical variables were associated with the class variable. The null hypothesis in these tests was that the two variables are independent of one another, while the alternate hypothesis was that the two variables are associated with one another. Variables not proven to be independent of the class variable through the chi-squared test were not removed; it is expected that they will be filtered out in the feature selection process.

Step 5: Multivariate Analysis

k-Prototypes Clustering

Because of their reliance on concepts such as means and Euclidian distance, many of the most popular and omnipresent clustering algorithms such as k-means clustering were unsuitable for mostly categorical data. Instead, the k-Prototypes clustering algorithm, which accounts for both categorical and continuous variables, was used.

The “elbow method” was employed to assess the optimal number of clusters. It was determined that five clusters best suited the data.

Much of the missing data and the vehicle fires were grouped together in cluster 2. As alluded to above, further analysis of this cluster revealed that many (but not all) of this missing data belonged to instances of vehicle fires: for most fire incidents that start in a vehicle, information about fire alarms, the structure level of origin, the smoke alarm, and the sprinkler system is truly not applicable.

Cluster 1 grouped together traits that suggest relatively low-impact but still complex fire incidents that were efficiently contained. The extent of the fire is confined to the object of origin; fire and smoke alarm systems were present and operational; all persons were evacuated upon hearing the alarms. Interesting, the most common level of origin were the Upper Floors, and the most common ward was Toronto Centre-Rosedale: this, along with the presence of rare amenities such as sprinkler systems and inconnected smoke alarms suggests that many of the fires in cluster 1 may have occurred in office or apartment buildings.

Clusters 4 and 5 have very similar profiles. The biggest difference is the season and time of day: cluster 4 grouped together summer fires that occurred at night, while cluster 5 grouped together spring fires that occurred during the afternoon.

Cluster 3 was limited to six records, much too small a sample size for any valuable insight to emerge.

Step 6: Fuse Categorical Variables

The cardinality of categorical variables was reduced significantly in the univariate analysis (step 3), but further fusing of levels was required, to prevent machine learning model sensitivity to sparse categories. Though in some cases (e.g. Incident_Ward), it was possible to further reduce cardinality employing business logic, for the vast majority of attributes sparse categories with the cumulative frequency of the bottom 15% of the data were combined together into a single category called "OTHER."

Step 7: Binarization

One-hot encoding

Following the univariate, bivariate and multivariate analyses, the data was binarized to make it useable for machine learning algorithms that don't accept categorical variables, as well as to ensure that the models will not misinterpret the ordering of the levels.

Correlation Analysis

Having transformed the data through fusion and binarization, a further round of correlation analysis was necessary to detect and eliminate multicollinearity. Using the findCorrelation function in R's caret package, pairwise correlations with a coefficient of over 0.75 or under -0.75 were identified and removed. The majority of the highly correlated attributes identified were those previous discussed in Steps 3 and 5 which were coded "Missing."

Following this step there are 96 attributes in that dataset. That binarized dataset can be seen here: https://github.com/terrygitersos/CKME136/blob/master/Data/binarized_final.csv

Step 8: Build and Evaluate Models

The methodology described in this section describes that which achieved the best training and test results with the baseline model. Other strategies were tested including, but not limited to, using unbinarized instead of binary data; partitioning the data to different sized training and test sets; using different numbers of features, including the entire data set; resampling the training data prior to cross-validation, both prior to and immediately after feature selection; employing larger and smaller cross-validation folds; removing mathematical outliers from the response_time attribute; centering, scaling, and reducing the dimensionality of the data through Principal Component Analysis. Many, but not all, of these trials have been documented here: https://github.com/terrygitersos/CKME136/blob/master/baseline%20tests.Rmd. The Final Code and Results can be found here: https://github.com/terrygitersos/CKME136/blob/master/Final%20Results%20and%20Code.Rmd.

Data Partitioning

The data was first sorted chronologically using the TFS_Alarm_Time attribute, then partitioned into a training set containing the oldest 80% of the data (10148 records), and a test set containing the most recent 20% of the data (2498 records).

Feature Selection

Three different feature selection algorithms were used, one from each of the three general families of feature selectors.

From the filter family, the Information Gain algorithm, which selects the attributes that maximize information gain and minimize entropy. From the wrapper family, the Recursive Feature Elimination method, where a model is fitted with all the attributes and the algorithm removes the weakest features one by one. From the embedded family, the LASSO algorithm, a type of linear regression that shrinks the regression coefficients toward zero, a process through which attributes are eliminated from the model.

There was some overlap in the features selected by the three algorithms as being most important. Four features appeared in the top ten of all three feature selectors, while a further four appeared in the top ten of two feature selectors.

Table 1 : Underlined features appeared in the top ten of all three feature selectors, while italicized features appeared in the top ten of two feature selectors.

#	Information Gain	Recursive Feature Elimination	LASSO
1	Extent of Fire: Other	Area of Origin: Functional Area	Area of Origin: Functional Area
2	Area of Origin: Functional Area	Extent of Fire: Confined to object of origin	Extent of Fire: Other
3	Property Use: Residential	Extent of Fire: Other	Exposures: Yes
4	Building Status: Other	*Smoke Alarm Impact on Evacuation: Other*	Property Use: Residential
5	Sprinkler System Presence: No Sprinkler System	Building Status: Other	*Material First Ignited: Undetermined*
6	*Initial CAD Event Type: Other*	*Material First Ignited: Undetermined*	*Smoke Spread: Confined to part of room/area of origin*
7	Property Use: Vehicles	Property Use: Residential	Smoke Spread: Spread to other floors, confined to building

8	Level of Origin: Upper Floors	*Smoke Spread: Confined to part of room/area of origin*	*Smoke Alarm Impact on Evacuation: Other*
9	Property Use: Miscellaneous	*Initial CAD Event: Other*	Building Status: Other
10	Ignition Source: Other Electrical/Mechanical	Area of Origin: Storage Area	Possible Cause: Misuse of Material First Ignited

The Recursive Feature Elimination rfe() function in R's caret package stores information indicating how many features should be optimally included in a model, according to a designated evaluation metric. For this measurement, only accuracy and Cohen's kappa can be selected for classification models. Because accuracy can be extremely misleading in the evaluation of imbalanced datasets, I selected Cohen's kappa as the evaluation metric, and tuned the algorithm to consider subsets ranging between one and 20 features (rfe() automatically considers the full data set as a whole). The optimal number of features returned was 20.

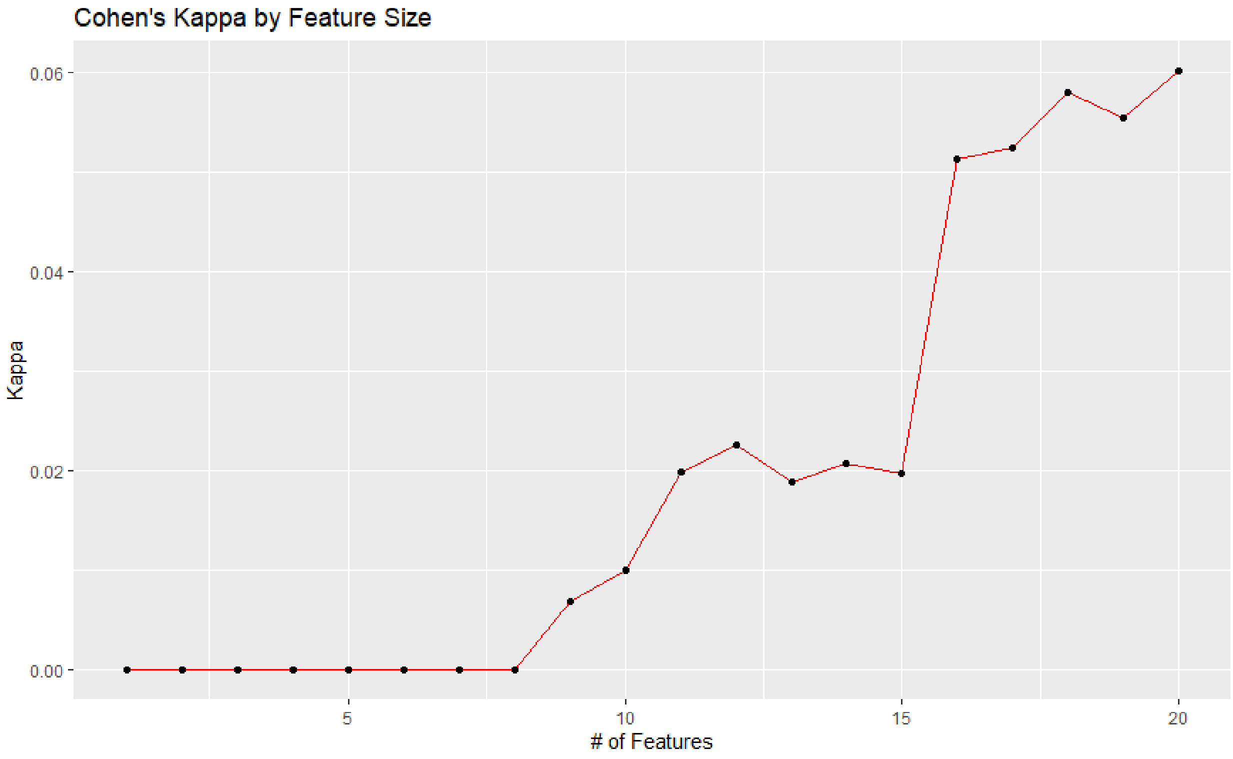


Figure 1

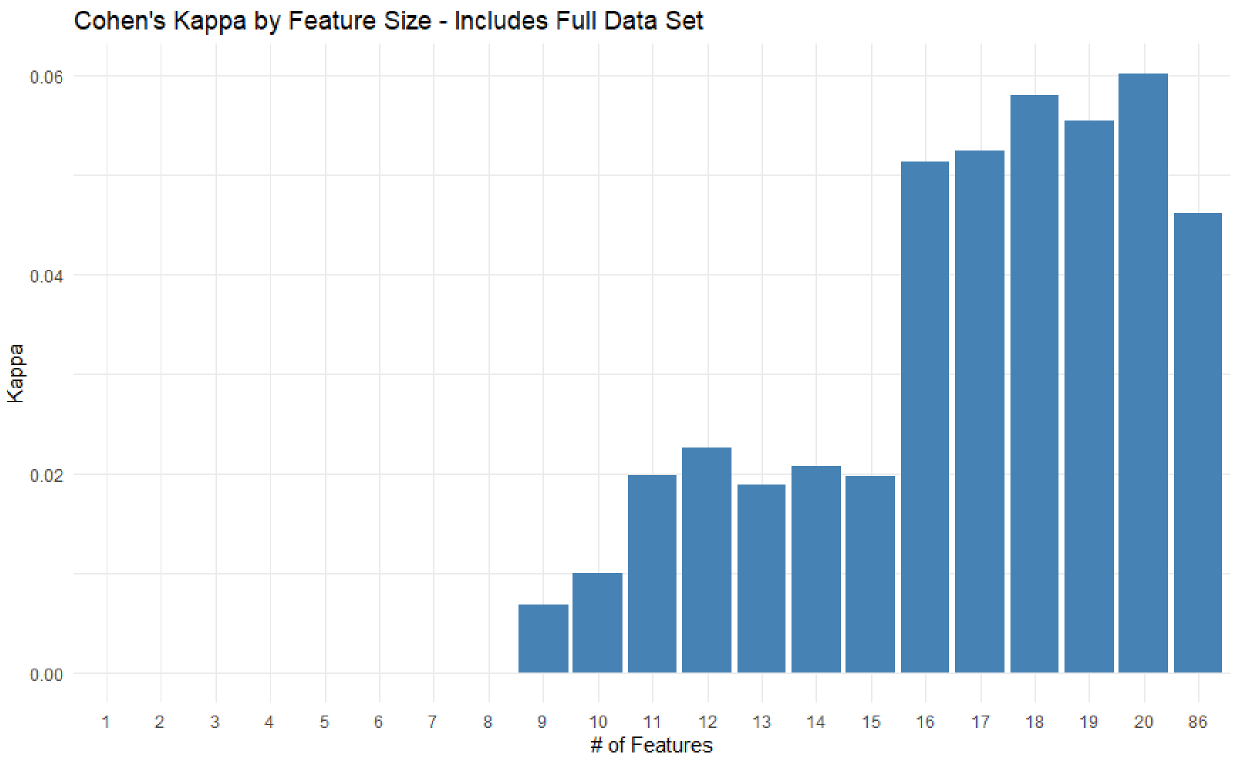


Figure 2

Generally, the kappa and the number of features are positively correlated and thus increment upwards together, so the model improves as more and more variables are selected. Additionally, the difference between the optimal number of features, and using all the features in the data set, is relatively slight. This suggests that the data cannot be easily generalized as good models must, which is a big impediment to building a simple and viable model with it.

The intention was to select a blended set of features by applying a points system to the output of the three feature selection methods, where the top-ranking attribute for a given feature selector received 10 points, the second top ranking attribute received 9 points, etc. But because the optimal number of features is so high this points system is moot. Instead, all 18 attributes that appeared in the top ten of any of the three feature selectors were selected.

Model Selection

I am guided here both by the updated benchmarking study by Lessman et al.,[13] as well as the fire risk prevention literature discussed above in the literature review. Lessman et al. recommend using one individual classifier, one homogeneous ensemble classifier, and one heterogeneous ensemble classifier. Following Lessman, I selected two individual classifiers, two homogenous ensemble classifiers, and one heterogeneous ensemble classifier that combines the other three classifiers. The models are as follows:

1. Decision Tree (baseline model)
2. Logistic Regression. This is the individual classifier that proved strongest in Lessman's study, and was also commonly employed in the fire risk prevention studies that I consulted.

3. Random Forest. This homogenous ensemble classifier was recommended by Lessman et al., and was also was commonly used in the fire risk prevention studies that I consulted. It was the best performing model for residential data in Walia's study of Pittsburgh fire data.
4. AdaBoost. Another homogenous ensemble classifier, AdaBoost is a boosted decision tree, a brand of classifier that tested strongly in Lessman's study. It is also very commonly used in the fire risk prevention studies that I consulted, and was the most effective model in Dang, Cheng, Mann, Hawick and Li's study of Humberside fire data.
5. Bagged (or bootstrap aggregated) hill-climbing ensemble classifier of models 2 through 4, using the caretStack function in R's caret package.

Resampling: Time Series Cross-Validation

In regular cross-validation, it is entirely possible for a model to be trained on the final 90% of a data set and validated on the first 10% of the data. In the context of time series data, this means that a model could attempt to predict events in past with data from the future. To preserve the chronological integrity of the data, the "sliding window" variety of time series cross-validation was used to train and validate the models. Each fold is comprised of 8,753 chronological records, with approximately the first 80% of the data (7002 records) in the fold allotted to training and approximately the final 20% in the fold (1751 records) used for validation. The folds are 155 records apart: the first fold starts at record number 1, the second fold starts at record number 156, etc.

Subsampling for Class Imbalance: SMOTE

Instead of subsampling the entire training set before or after feature selection, approaches that can skew validation results and lead to overfitting, the data was subsampled during cross-validation within each fold. The oversampling method used is the Synthetic Minority Oversampling Technique (SMOTE). SMOTE is a hybrid method of subsampling, where new data points are synthesized in the minority class while the majority class is randomly down-sampled. The default SMOTE setting in the trainControl function in R's caret package was used, which doubled the number of records in the minority class, while removing double the number of new records created from the majority class (e.g. if 100 new records are created in the minority class, 200 records are randomly removed from the majority class).

Evaluation Metrics

Two evaluation metrics are primarily used in this analysis: Cohen's kappa, a metric that describes the model's overall accuracy while taking into account the expected accuracy; and the harmonic mean (F-score or F-value), a measure combining precision and recall that describes the model's ability to predict the positive class (in this case, fires resulting in deaths and near-deaths). F-score was selected instead of just precision or just recall, because both of those measurements were judged to be equally important in this analysis: while we are certainly interested in correctly predicting as high a proportion of fires resulting in deaths and near-deaths as possible (recall), in a real world situation it would equally important to avoid false alarms (precision).

Training Decision Tree, Linear Regression, Random Forest and AdaBoost

After the baseline Decision Tree, Linear Regression, Random Forest and AdaBoost models were trained, the kappa values and F-scores from each individual fold were compared against each other using

Friedman tests. For both kappa and F-value the null hypothesis was rejected, meaning that at least one of the models has statistically different results (better or worse) than the others. A further round of Friedman testing was conducted excluding the baseline Decision Tree model. Again, the null hypothesis was rejected, meaning that at least one of Logistic Regression, Random Forest, or AdaBoost is statistically different than the others.

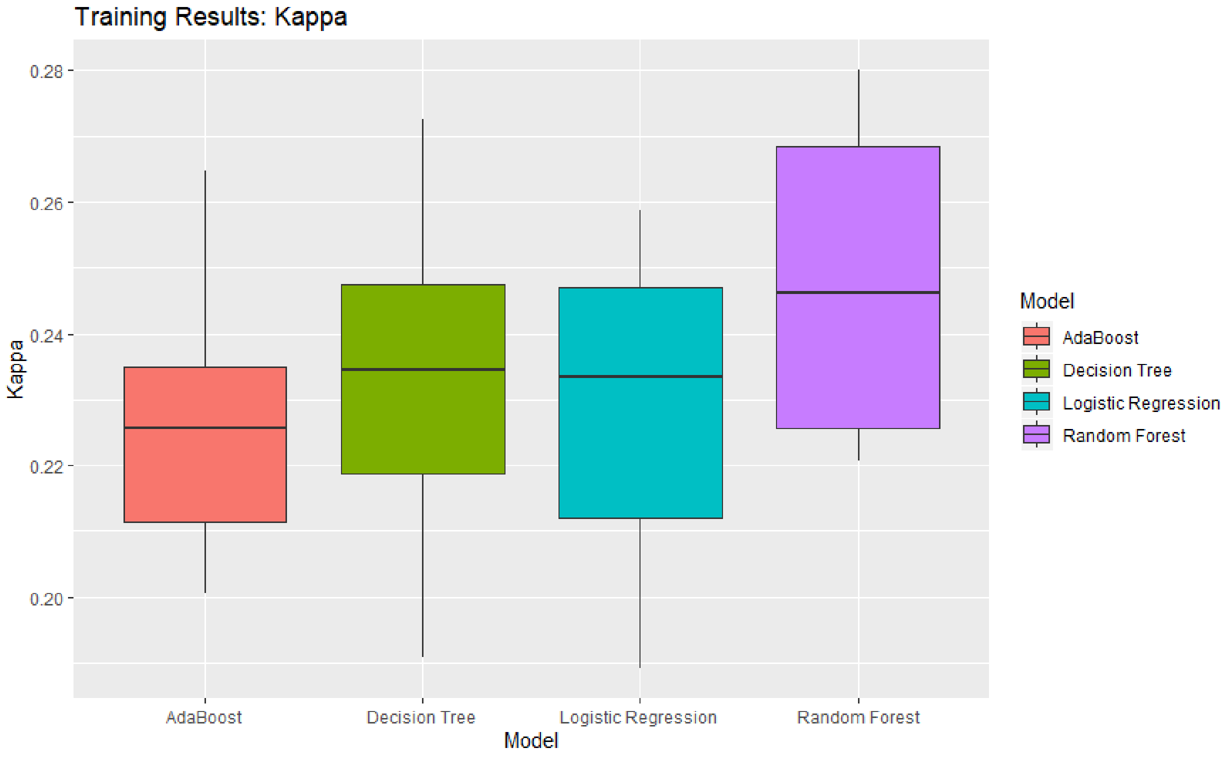


Figure 3

The kappa values are quite low overall, indicating that the model is struggling to accurately make predictions. Of the four models tested, the Random Forest model appears to be the best by this metric, by a very slight margin. It has the highest median kappa, while achieving the lowest minimum kappa and highest maximum kappa. Its spread is comparable to the other models.

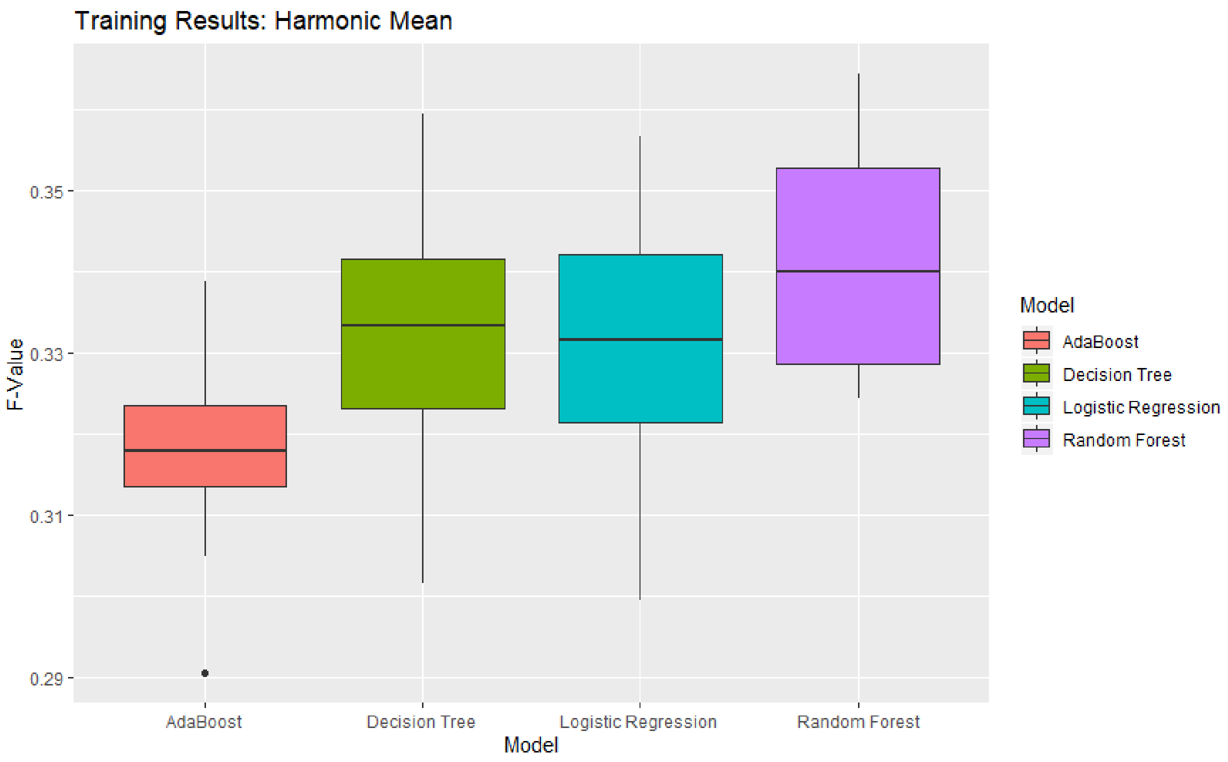


Figure 4

The F-values tell a similar story as kappa: the values are quite low overall, suggesting again that the models are struggling to accurately make predictions. Once again, Random Forest appears to be the strongest model by a slight margin: it returned the lowest minimum and the highest maximum values, and the highest median value. It is also the most consistent model from one vantage point, with the lowest spread between minimum F-value and maximum F-value of any model.

By this metric, AdaBoost is the weakest model. Decision Tree and Logistic Regression appear once again to be very similar, with comparable maximum, minimum, and median F-scores.

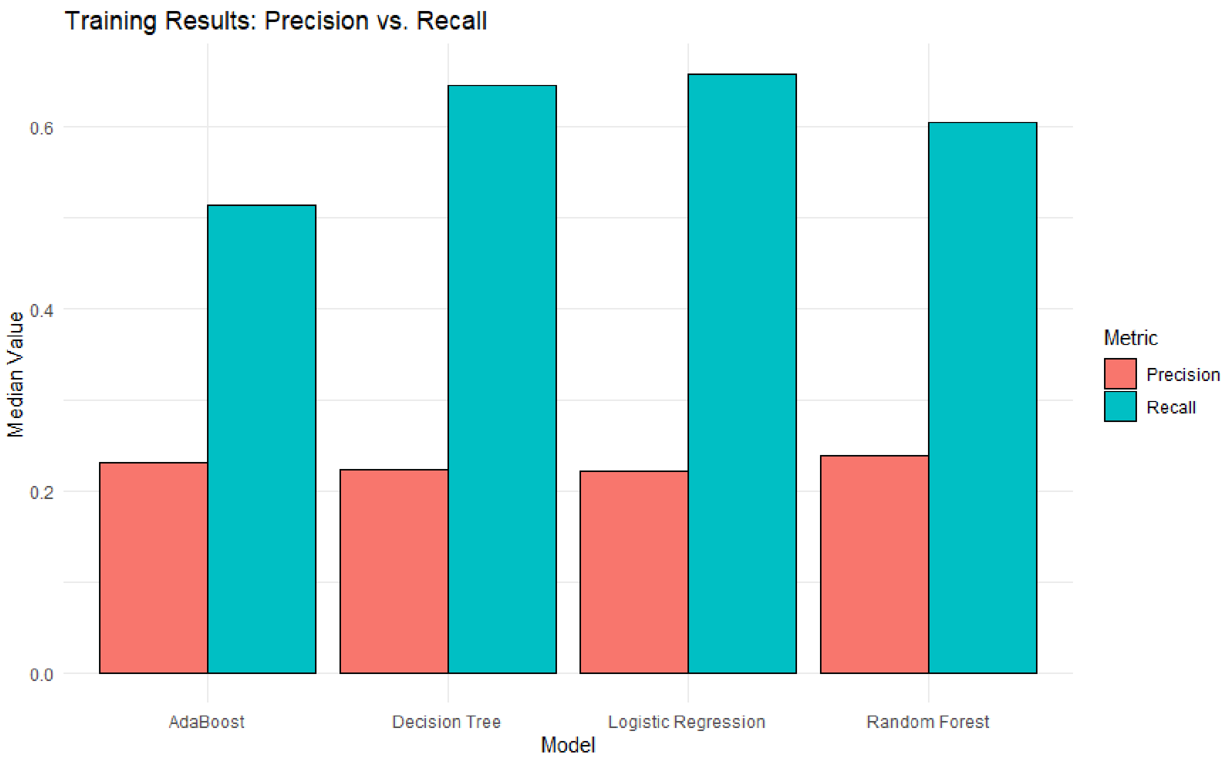


Figure 5

The reason for the weak training results is type I errors (false positives, or false alarms). While the models are reasonably adept at identifying fires resulting in deaths or near deaths for what they are (as evidenced by the recall values), they are over-predicting the minority class which results in type I errors and very weak precision values. All models are acting similarly in this respect, with some small variances. Random Forest, for example, has slightly lower median recall than Decision Tree or Logistic Regression, but incrementally higher median precision.

The last action taken in training these four models was to assess their correlation. And indeed, AdaBoost aside, the models are strongly correlated to one another. This suggests that Decision Tree, Logistic Regression and Random Forest have similar strengths and weaknesses and are predicting much of the same data well and much of the same data poorly.

Ensemble Model

As described earlier in this report, following Lessman et al., I opted for a bagged (or bootstrap aggregated) hill-climbing ensemble classifier of the Logistic Regression, Random Forest and AdaBoost models, using the caretStack function in R's caret package. I selected Random Forest as the "meta-model" in this stacking process, as it appeared to be the best performing of my four models.

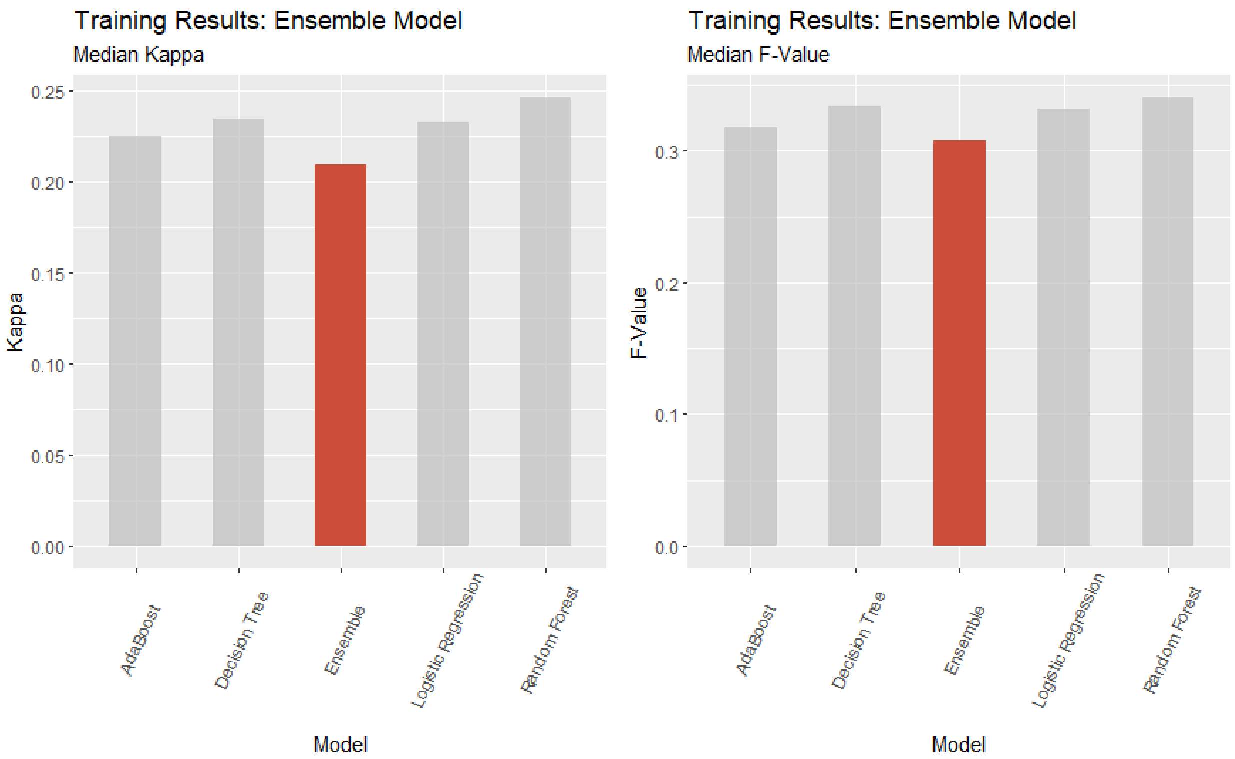


Figure 6

As can be seen in Figure 6, the median kappa and F-value of the ensemble model is lower than any of the four models that went into its ensemble. This is unexpected. The expectation was that the ensemble model would have all the combined strengths of its constituent models, but instead appears to have picked up their weaknesses in equal measure.

Step 9: Test Models

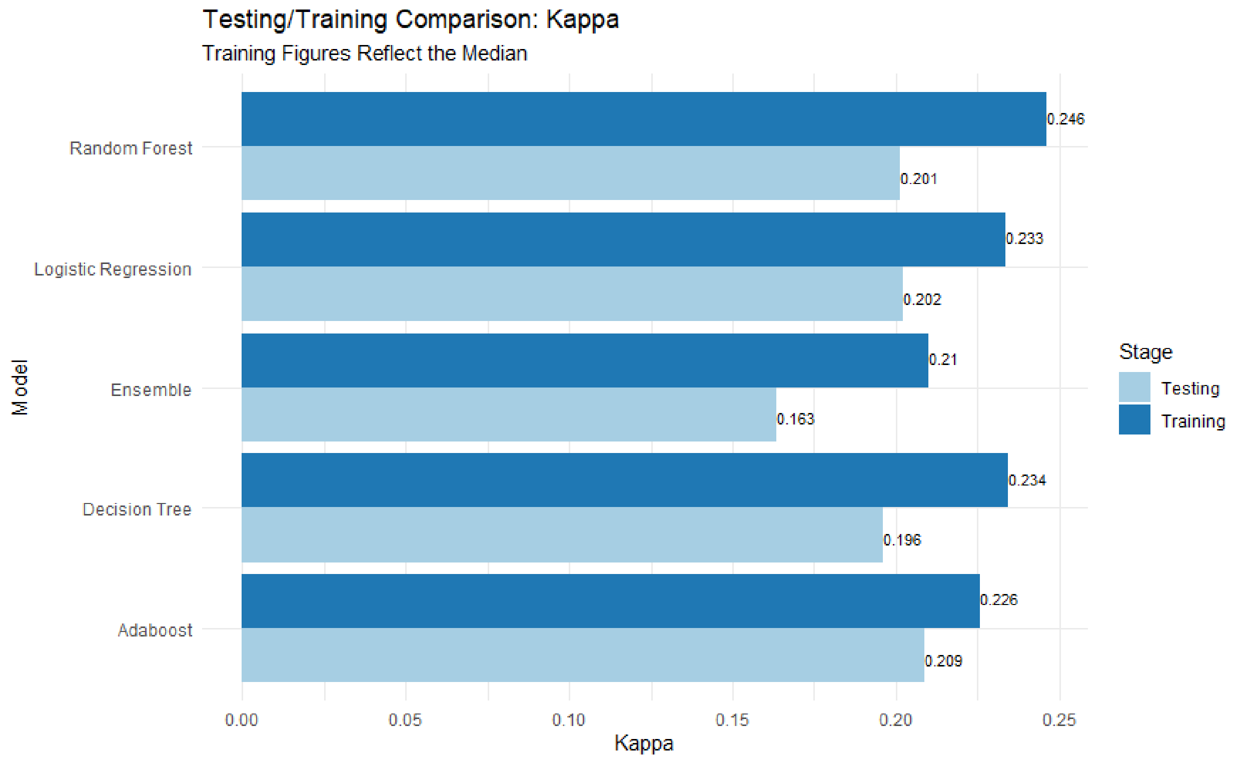


Figure 7

The models were characterized by test kappa values that were over 13% lower than the median training kappa value with the notable exception of AdaBoost, whose test Kappa is only 7.5% lower than its median training kappa. AdaBoost had the highest test kappa value, though the difference in kappa between AdaBoost, Random Forest, Logistic Regression and Decision Tree is quite small. The Ensemble model fared by far the worst of the five models with a kappa of only 0.163, approximately 22% worse than its median training kappa.

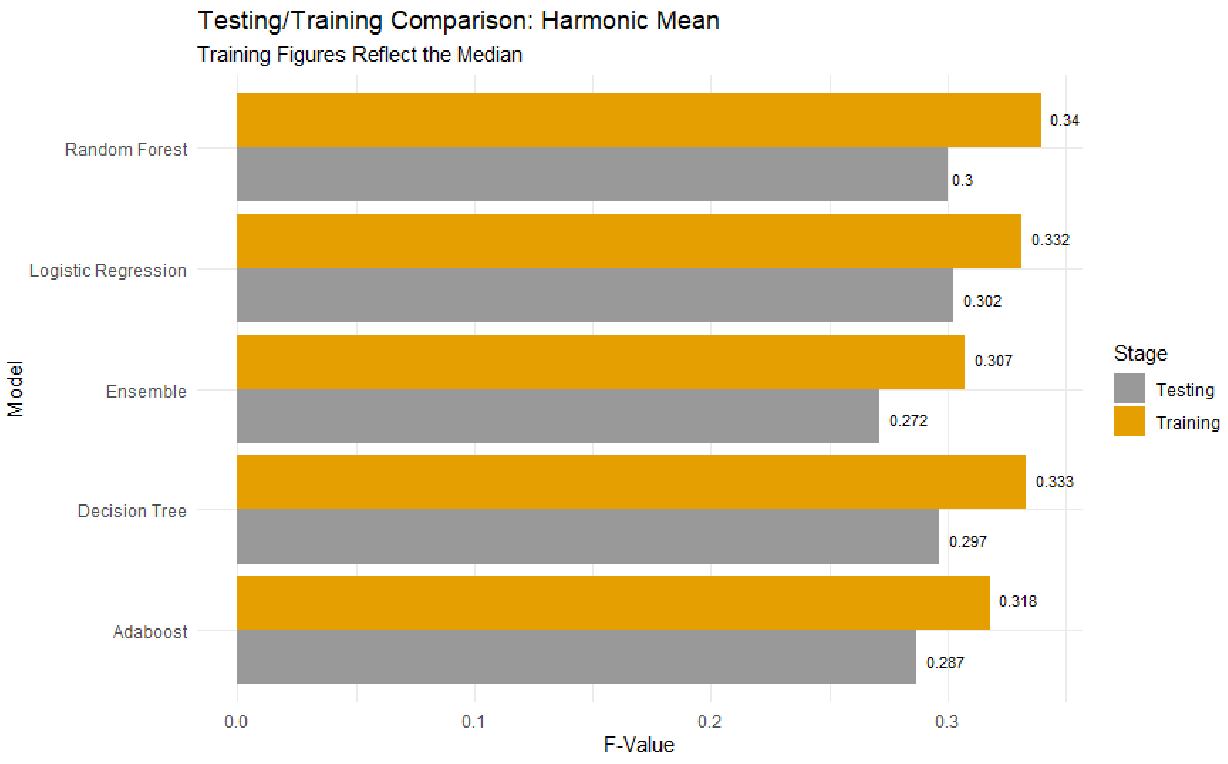


Figure 8

As with kappa, all models performed quite poorly and returned a worse F-value in testing than their median training f-value (between 9% and 12% worse). By this metric, Logistic Regression performed best, though the difference between it and Random Forest is a razor-thin 0.02. The Ensemble Model once again fared the worst of the five models.

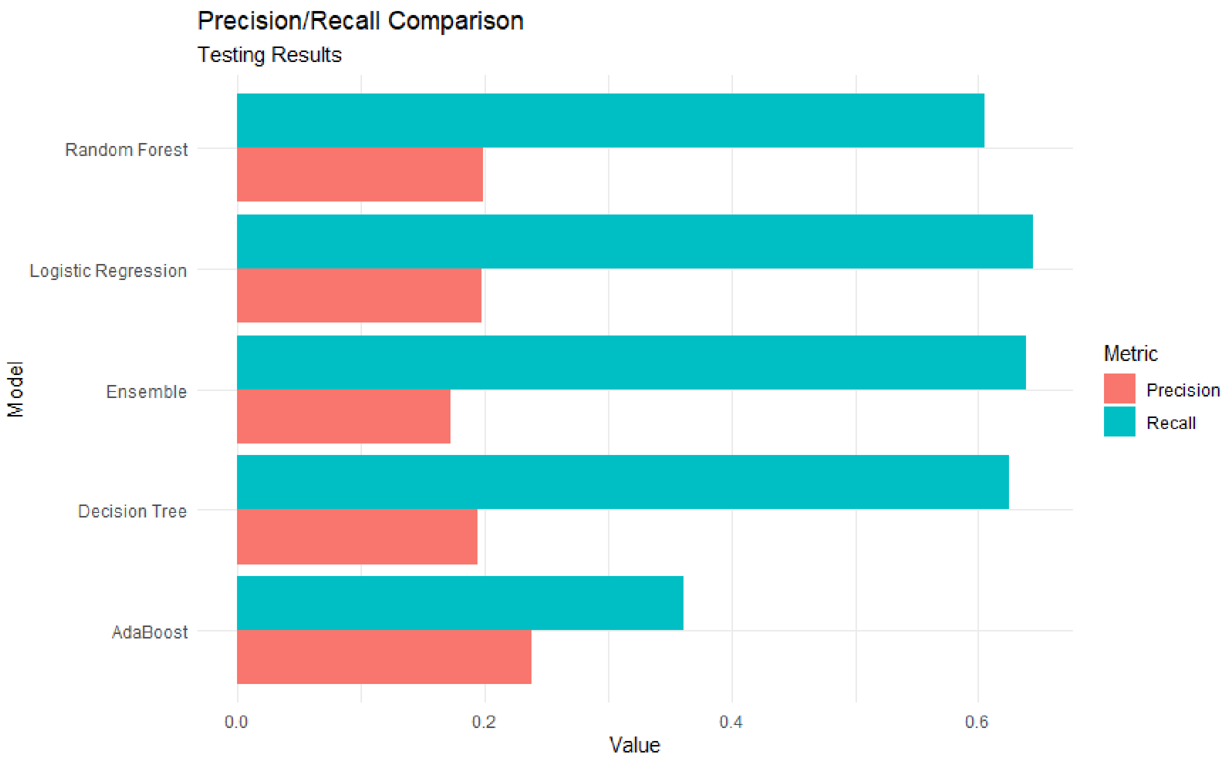


Figure 9

As was observed in training the models, the weak results are due to a preponderance of type I errors; as during training, the models all significantly over-predict the minority class. Interestingly, AdaBoost comes closest to convergence in its precision and recall, but both values are quite low resulting in the low harmonic mean seen in Figure 8.

Conclusions

In conclusion, none of the models trained and tested classify the data nearly well enough to be deployed in a real-world capacity, especially for a possible life-and-death situation such as a fire incident. The models, though differing slightly in effectiveness, behave similarly overall and share the same shortcomings. While reasonably capable of recognizing a fire resulting in death or near-death (precision), the models grossly over-predicted the number of those fires, resulting in very poor precision. It is highly unlikely that further tweaks or tuning changes to these models will result in enough improvement to transform the models into viable classifiers.

These results, as well as the information discussed earlier about the optimal number of features gleaned from the Recursive Feature Selection algorithm, strongly suggest that it likely will not be possible to build a viable classifier for this specific research question with this particular data set. More and different data would be required for starters to potentially train classifiers to predict fires resulting in deaths or rescues, though researchers would be advised to select a different proxy metric for risk when analyzing this data.

[1] Parisa Moshashaei and Seyed Shamseddin Alizadeh. 2017. Fire Risk Assessment: A Systematic Review of the Methodology and Functional Areas, In *Iranian Journal of Health, Safety and Environment* 4, 1 (2017), 654-669

[2] Eddie Copeland. 2015. *Big Data in the Big Apple*. Capital City Foundation (2015). http://capitalcityfoundation.london/wp-content/uploads/2015/06/Big-Data-in-the-Big-Apple.pdf

[3] Michael Madaio, Shang-Tse Chen, Oliver L Haimson, Wenwen Zhang, Xiang Cheng, Matthew Hinds-Aldrich, Duen Horng Chau, and Bistra Dilkina. 2016. Firebird: Predicting Fire Risk and Prioritizing Fire Inspections in Atlanta. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 185–194 (2016).

[4] Bhavkaran Singh Walia, Qianyi Hu, Jeffrey Chen, Fangyan Chen, Jessica Lee, Nathan Kuo, Palak Narang, Jason Batts, Geoffrey Arnold, and Michael Madaio. 2018. A Dynamic Pipeline for Spatio-Temporal Fire Risk Prediction. In *KDD '18*. ACM, 764–773 (2018).

[5] Singh et al., pp. 768.

[6] Trung Thanh Dang, Yongqiang Cheng, Joanne Mann, Ken Hawick, Qingde Li. 2019. Fire Risk Prediction Using Multi-Source Data: A case study in Humberside area. In *Proceedings of the 25th International Conference on Automation and Computing (ICAC)*, Lancaster, United Kingdom, 1-6 (2019)

[7] Dang et al., pp. 5

[8] Kevin Pirklbauer and Rainhard Dieter Findling. 2019. Predicting the Category of Fire Department Operations. In *The 21st International Conference on Information Integration and Web-based Applications & Services (iiWAS2019), December 2–4, 2019, Munich, Germany*. ACM, 5 pages (2019). https://doi.org/10.1145/3366030.3366113

[9] Qianru Wang, Junbo Zhang, Bin Guo, Zexia Hao, Yifang Zhou, Junkai Sun, Zhiwen Yu, Yu Zheng. 2019. CityGuard: Citywide Fire Risk Forecasting Using A Machine Learning Approach. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 4, Article 156 (2019)

[10] Joseph Clare, Len Garis, Darryl Plecas, and Charles Jennings. 2012. Reduced frequency and severity of residential fires following delivery of fire prevention education by on-duty fire fighters: Cluster randomized controlled study. In *Journal of safety research* 43, 2 (2012), 123–128.

[11] Anders Ohrn. 2019. Toronto on Fire in Data, Part 1. In *Beyond Data Science* (2019). https://towardsdatascience.com/toronto-on-fire-in-data-part-1-484435eca880

[12] Anders Ohrn. 2019. Toronto on Fire in Data, Part 2. In *Beyond Data Science* (2019). https://towardsdatascience.com/toronto-on-fire-in-data-part-2-33e150b8e45d

[13] Stefan Lessman, Hsin-Vonn Seow, Bart Baesans, and Lyn C. Thomas. 2013. Benchmarking state-of-the-art classification algorithms for credit scoring: A ten-year update. In *Credit Research Centre, Conference Archive.* (2013). https://crc.business-school.ed.ac.uk/wp-content/uploads/sites/55/2017/02/Benchmarking-State-of-the-Art-Classification-Algorithms-for-Credit-Scoring-Lessmann-Seow-Baesens-and-Thomas.pdf