DOI: 10.15439/2014F502
ACSIS, Vol. 2

Robust Method of Sparse Feature Selection for Multi-Label Classification with Naive Bayes

Dymitr Ruta
Etisalat, British Telecom Innovation Centre
Khalifa University, Fatima F302, PO Box 127788
Abu Dhabi, UAE
Email: dymitr.ruta@kustar.ac.ae

***Abstract*—The explosive growth of big data poses a processing challenge for predictive systems in terms of both data size and its dimensionality. Generating features from text often leads to many thousands of sparse features rarely taking non-zero values. In this work we propose a very fast and robust feature selection method that is optimised with the Naive Bayes classifier. The method takes advantage of the sparse feature representation and uses diversified backward-forward greedy search to arrive with the highly competitive solution at the minimum processing time. It promotes the paradigm of shifting the complexity of predictive systems away from the model algorithm, but towards careful data preprocessing and filtering that allows to accomplish predictive big data tasks on a single processor despite billions of data examples nominally exposed for processing. This method was applied to the AAIA Data Mining Competition** 2014 **concerned with predicting human injuries as a result of fire incidents based on nearly 12000 risk factors extracted from thousands of fire incident reports and scored the second place with the predictive accuracy of 96*%*.**

I. Introduction

The unmanageable scale of big data comes in many forms symbolically paraphrased by *5Vs*: Volume, Velocity, Variety, Veracity and Value [1]. Huge volume defined by both the size or dimensionality of big data is one such "V" that particularly adversely affects computational complexity of the process of learning from data. The hype about big data may be therefore elusive while its possible value very difficult to extract. There are many examples reported in the literature that demonstrate both very powerful and very ineffective exploitations of large data sets for predictive tasks [4], [1], [1], [3].

Inspired by the pioneering work in [4], however, there is a widespread belief that the more data the better and the inability of exploiting it all is just a reflection of the predictor's weakness [3]. We argue, however, that a blind admission of all big data into the predictive modelling may be wrong or at least inefficient approach for some class of problems. Although certain cognitive tasks may indeed require billions of data points to reveal the full explanative power of the data [4], [5], our experience indicates that the majority of data problems can be explained by the relatively small data sample, which might be buried under the masses of big data. For these problems the availability of big data for predictive analytics widens the choice and the opportunity for both novel data exploitations and the improvement of predictive performance of the existing models.

As a result, the emerging paradigm of working with big data appears to be centred around careful data filtering, preprocessing, features generation and selection. Very often these procedures eliminate most of the original data leaving only essential evidence that retains almost complete explanative power [3]. What is more, the evidence reported in the machine learning literature indicates that given a typical supervised learning problem the key drivers for performance lie predominantly in the discriminative power and the choice of the data features rather than in the complexity of the predictive model [6], [7], [8], [9], [10]. All these points lead to a conclusion that when faced with the problem of learning from big data the main challenge and effort should be directed towards extracting or generating the key explanative data features while the actual learning and predictive performance could be delivered with relatively simple and robust learning model [10].

In line with this approach we have entered AAIA'2014 data mining competition with the intend to demonstrate how effective could be feature selection for supervised learning problems with very high dimensional data. We proposed a relatively easy and fast, greedy feature selection method that works particularly well with the large number of sparse features. In the competitive environment we will demonstrate that it delivers very high performance with a very simple predictor like Naive Bayes. We also propose much faster yet nearly equally robust feature selection method that eliminates completely the need of predictor application, and for that as we argue it is a very strong contender for real-time applications of predictive analytics on big data.

II. Task Description

AAIA Data Mining Competition 2014 was concerned with extracting the risk factors and attributes of fire incidents that would allow the most accurate prediction of human injuries or casualties as a result of these incidents. The total of 11852 features extracted from 50000 fire incident reports were presented as input data and the objective of the competition was to select a subset of features that would achieve the best predictive accuracy of detecting simultaneously the following 3 binary class target outputs with the Naive Bayes model:

- serious injury or death of one of the firefighters or members of the rescue team
- children were among injured people
- civilians were among hurt/injured people

978-83-60810-58-3/$25.00 © 2014, IEEE

Additional constraint enforced by the competition was the format of the solution and its assessment. The format of the solution was enforced to be organised within 10 feature subsets of at least 3 features each and the performance metric was set to be the area under the curve (AUC) of the receiver-operator curve (ROC) obtained from averaging the outputs from Naive Bayes classifier ensembles across all 3 target variables. The performance metric additionally incudes the penalty term that penalises for using many features in the solution as in the following:

$$score(s) = max\left\{0, \frac{1}{3}\sum_{i=1}^{3} AUC_i(s) - \left(\frac{|s|-30}{1000}\right)^2\right\} \quad (1)$$

Note that the size penalty term reduces to 0 when all 10 selected feature subsets have exactly 3 features. The problem is challenging due to the fact that the input data is huge, high dimensional and sparse in nature, while the class target values are highly imbalanced. Further difficulty is that the evaluation considers the average performance of de-facto 3 distinct classification problems sharing the same features. A successful feature selection method needs to find the compromise in maximising the average performance of all the 3 models at the same time.

III. Feature Elimination

Given the very large feature dimensionality and the sparse nature of the input data the first natural step is to eliminate redundant features that have no chance of contributing to the performance of the target prediction. The approach taken was that given the feature sparsity and huge imbalance of the target class variables, the features which have all non-zero values occurring only at negative class outputs have completely zero predictive power in isolation or in combination with other features. Denoting by $X^{[N\times M]}$ the matrix of input data and by $Y^{[N\times 3]}$ the matrix of corresponding class outputs we can safely eliminate redundant features by applying the following simple filtering expressed in Matlab formulation:

```
F = find(sum(X(any(Y, 2), :))>0);    (2)
```

This simple filtering resulted in elimination of 1931 (16.3%) redundant features. An interesting observation is that if the three target class variables were to be predicted and assessed separately, the above filtering would have resulted in much deeper reductions of: 6418 (54.1%), 6174 (52.1%), and 2146 (18.1%), respectively. What is more, separating predictive tasks would further allow to identify feature redundancy through containment. Namely, we can further eliminate a feature *A* whose non-zero intersection with the positive target class (true positives) is fully contained by other feature *B*, while its non-zero intersection with the negative class (false positives) fully contains feature *B*. What it means is that prediction with feature *A* would always be less accurate than with feature B which is guaranteed to make more positive predictions (true positives) at a lower costs (false positives). Such further elimination through feature containment would achieve the reductions of 9754 (82.3%), 9313 (78.6%), and 5005 (42.2%) respectively. It suggests that it might be much more efficient to model all three predictive tasks independently rather than force them to share the same feature subset of input data.

Constrained by the evaluation criterion defined by eq. 1 the feature set to work with had to be left with the only lightly reduced size - down to 9921 features, to avoid the loss of information.

IV. Naive Bayes Classifier

Naive Bayes (NB) is a simple yet very effective and fast probabilistic classification method that naively assumes that features are conditionally independent given the class value [11]. Given a binary classification problem with n features F_i the NB model tries to give the estimate of the posterior class probability given feature observations: $p(C, F_1, ..., F_n)$. From the chain rule applied to conditional probability definition the searched likelihood becomes:

$$p(C, F_1, ..., F_n) = p(C)p(F_1|C)p(F_2|C, F_1)... \\ ...p(F_n|C, F_1, F_2, ..., F_{n-1}) \quad (3)$$

which after applying the naive assumption of conditional feature independence simplifies to:

$$p(C, F_1, ..., F_n) = \frac{1}{Z}p(C)\prod_{i=1}^{n} p(F_i|C) \quad (4)$$

where Z is a constant scaling factor that is fixed for known feature variables.

Since most of the features are binary or categorical we are dealing with the multinomial distribution here, and constructing a posterior class likelihood is just a matter of calculating a product of class conditional probabilities of specific feature values observed for every input data instance. This process is critical and most often repeated when evaluating classification performance hence it is reasonable to speed it up by precalculating the class conditional likelihoods of all feature values empirically from the training data. Calculating the posterior would be then reduced to taking the relevant class conditional feature probabilities from the lookup table and multiplying them together or adding log likelihoods in odder to avoid the precision loss for small numbers.

Since the competition performance metric was an AUC of the ROC curve, the class posteriors are all that is required for the score calculation.

V. Sparse Feature Selection Strategies

Given the training input data of 50000 examples composed of nearly 10000 sparse features (after filtering) and a well defined and fast predictive performance metric defined in eq. 1 the objective was to extract 10 subsets of features that would maximise the expected predictive performance on the unseen testing set. Our preliminary investigations revealed that separating a validation set out of training data appears to be a good method for comparing the generalisation robustness of the strategies. On the other hand setting any data aside for the validation reduces the evidence that the predictive model is learnt on and hence may not give the best performance on the testing set. The approach that was finally taken was to use the actual performance feedback to decide whether a validation set improves the predictive performance on the testing set.

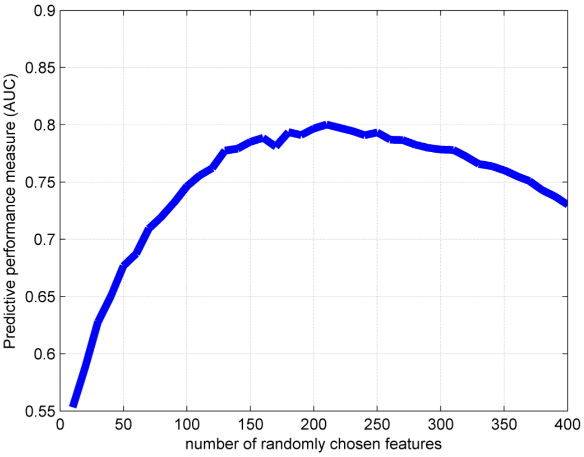

Fig. 1. Random subset method performance curve

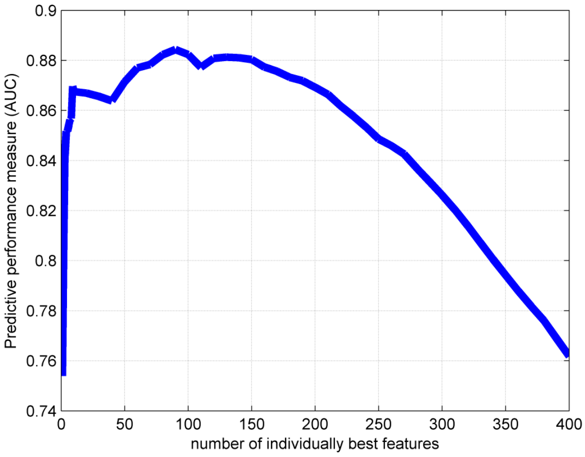

Fig. 2. Incremental single best performance curve

A. Random Feature Subsets (RFS)

Random feature subset method appears very naive for large feature space problems, however it was quickly developed to provide some intuition around the predictive value of the data and to set some baseline predictive performance levels. It was also useful for establishing the impact of the performance metric penalty term provided by eq. 1 and through some experiments draw an estimate of what might be the optimal size of the feature set that would maximise the competition performance criterion.

The performance of random feature subset selection method was evaluated using random features set sizes increasing from 10 to 500 at a step of 10 and obtaining corresponding performance measure inline with eq. 1. It has been repeated 50 times and the results averaged to build stable performance estimates for increasing number of random features included in the model. The resulting performance curve is presented in fig. 1.

As it can be seen from the figure, the performance of the random feature subset method is expectedly quite poor and peaks for roughly 200 features included in the model.

B. Incremental Single Best (ISB)

The random feature subset method performs quite poorly but it does not require any computation effort related to feature selection. Incremental single best method presented here goes a step further and shifts the balance towards improving the predictive performance at the relatively small prior computational cost of evaluating all individual features performance. Since each feature is evaluated in isolation, no conditional feature dependencies are considered, and the model build simply follows the greedy strategy of sequentially adding individually best available features until their combined predictive performance stops increasing. Fig. 2 illustrates the performance curve of such incremental single best selection strategy for the feature set sizes set from 1 to 500. Clearly the performance curve very quickly climbs to a much higher levels above 0.88 comparing to the random subset method and it peaks for about 90 individually best features included in the model.

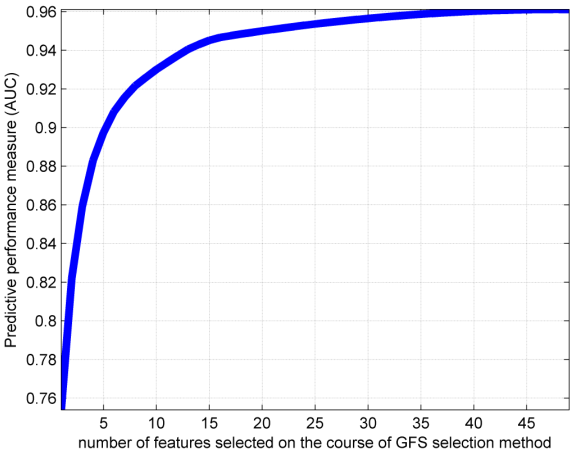

Fig. 3. Greedy forward search performance curve

C. Greedy Forward Search (GFS)

The next feature selection strategy to consider is a traditional greedy forward search. This method would start from the same individually best feature. At each round it then checks the performance change of adding all remaining features one at the time to finally add a feature that results in maximum possible improvement of predictive performance. This search already introduces a significant processing cost as its computational complexity based in the number of performance evaluations is $O(N \times n)$ where N is the total number of available features and n stands for the number of features selected for the model.

The performance curve of the GFS is presented in fig. 4.

It might be surprising that it beats the performance of all previously presented selection methods with just 4 first features. The performance curve peaks with 49 features. The reported validation performance in excess of 0.96 was comparable to the training performance and on its own climbs to a competitive level. The advantage of GFS method is a rapid performance growth with just a few features yet the computational complexity heading towards quadratic order becomes a real issue here and caused the complete search to take few days on the standard PC. Further drawback of this method is that beyond few features the risk of falling into local shallow maxima grows really fast and affects the method ability to find robust solutions.

D. Diversified Greedy Backward-Forward Search (DGBFS)

Greedy forward search introduced in the previous section demonstrated really good potential for high predictive performance that is however hindered by the problem of local maxima trap. The proposed diversified DGBFS method tries to exploit the strengths of the forward search method while improving its flexibility to get out of local maxima traps, increasing the exposure to the diversity of the whole feature set and significantly improving the speed of the search.

The method starts from the same greedy forward search but rather than adding only the single feature that maximally improves the performance for every feature set scanning round it keeps adding all the features that improve the currently best performance. As a result a single forward scanning round could add hundreds of features instead of just one. What happens then is a backward search, in a sense that all features selected so far are attempted to be removed from the set and such removal is granted if it causes the performance improvement. Such backward search adds vital ability of the method to refine its earlier greedy choices by exploring latter additions that do not maximise the performance gain but lead to better longer term solutions.

The forward and backward scanning rounds follow each other in a sequence until for both not a single addition or removal is able to improve the performance. Since this method is dependent on the order of features presented for the scanning, feature indices are randomly permutated before each scanning round such that the whole feature space is equally exposed to the chances of being selected.

The complete performance curve across many rounds of additions and removals is visualised in fig. 4.

Forward moving sections represent the performance progression during forward search and backward sections reflect the corresponding performance gains during backward search. Notable is a big overshoot of the size of the selected feature set during the first forward search. This was the effect of the initial ease of improving the performance through additions. In fact most of the newly added features were later removed in the subsequent backward search since they were added not because they were very robust but because they were just better than random features initially populating the selected feature set.

The presented DGBFS feature selection method achieved the top expected performance of over 0.97 and was selected to generate solutions for the AAIA'14 Data Mining Competition. Both the initial feedback and the final assessment positioned its solution on a second place in the competition trailing just a fraction of a per cent behind the top wining solution.

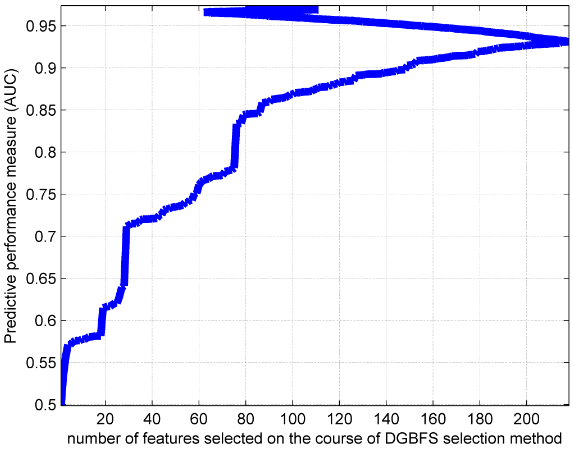

Fig. 4. Diversified greedy backward forward search performance curve

E. Fast Cumulative Sparse Feature Count Search (FCSFCS)

The greedy feature selection method presented in the previous section results in the best predictive performance as defined by eq. 1 that uses optimised Naive Bayes classifier to generate posterior class likelihoods. However, even such simple classifier additionally optimised for processing speed still absorbs non-negligible processing time and, due to the nature of Naive Bayes implementation, may require temporary expansion from the sparse to full representation making it impossible to evaluate the predictive performance with very many features. These issues significantly adversely affect the model scalability and might render its application impossible for larger scale problems, especially in the real-time operation.

To address this issue a further significant simplification is introduced which models the Naive Bayes posterior by just a simple sum of binarised features. We assumed that all the sparse features can be converted into a binary representation that indicate simply a presence of non-zero value. In case of the opposite enumeration of the features, binarisation should be preceded by the value conversion such that binarised "1/true" is always assigned to the sparse class i.e. unlikely set of feature values that has the highest joint probability with the positive target class. Once such binarisation is completed the posterior probability of the positive target class given the features can be simply modelled by the sum of positive binarised feature values which is equivalent to the voting count of true features for each input example. Such sum on binary features is extremely fast to calculate, is fully compliant with sparse feature representation and can swiftly evaluate the models with extremely large feature subsets. What is more it is actually performing very well as a classification method just slightly trailing the Naive Bayes classifier.

The performance of such method has been explored due to its very attractive properties of scalability and speed crucial for applications of huge high-dimensional data for predictive analytics purposes. Using this method allowed to explore normally prohibitive search strategies of greedy backward search (starting from the whole set) and multiple greedy ensemble search that now we managed to carry out in a matter of minutes

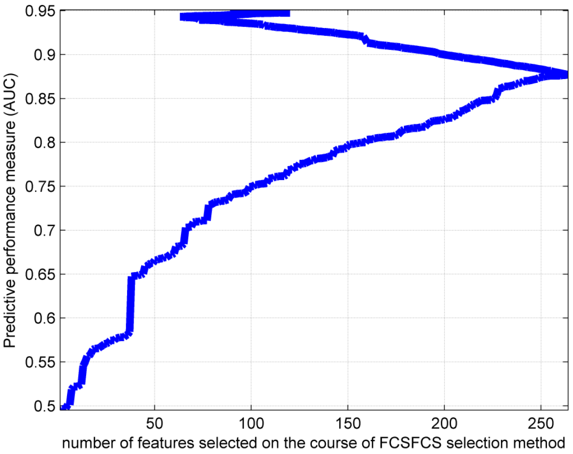

Fig. 5. Fast cumulative sparse feature count performance curve

on a standard PC.

Fig. 5 presents the performance curve achieved by the FCSFCS method using the same penalised AUC of ROC performance measure defined in eq. 1. Although it is just over 1% behind the top DGBFS method, given its simplicity and swift processing taking just minutes it really is a very good and fast feature selection proposition method or in fact a complete very shallow yet robust predictor.

F. *Individual vs Ensemble Models*

For the evaluation purposes a clarification is required as to the way feature subsets were evaluated. The competition performance criterion defined in eq. 1 clearly enforces the construction of the ensemble of 10 feature subsets with at least 3 features in each subset. The question of whether to construct the ensemble of the subsets of features or use just a single flat subset in fact reflects a long standing dilemma of individual vs ensemble learning. One motivation for ensemble learning is that it is much more efficient and faster to build multiple models with different parts of the feature subset if the computational complexity of the learning process exceeds linear order. This is however not the case for Naive Bayes classifier that given N examples of M-dimensional features is linearly complex in both $O(NM)$. There are also many examples reported in the literature, how a combination of weak learners each built on a small subset of evidence outperforms the single predictor trained on the complete evidence [12]. This effect of ensemble robustness through synergic complementarity is well known and reported on in ensemble learning methods like boosting [13], [12], where the performance gain through combination of weak learners is probably the most exposed.

What we have seen in the context of predominantly greedy feature selection methods is that building the ensemble of feature subsets is much more prone to overfitting. In fact we have built the ensemble versions of the DGBFS and FCSFCS methods where greedy additions or removals were done in turns for all ensemble subsets and the methods terminated when it was not possible to improve the performance for neither addition nor removal from any of the ensemble feature subsets. For all such experiments we have observed a consistent pattern of training performance improved by more than 1% but the validation and testing consistently down by almost 2%. We have also observed a pattern of about 10% to 20% increase in the total number of features selected with the ensemble evaluation method. A possible explanation is that with 10 different feature subsets the ensemble search has many more degrees of freedom and appears to find many new ways to better fit the training data with more features despite the penalty term. In the validation or testing phase, those many unstable coincidences of feature values turn out to be just random noise while the penalty term hits back with the guaranteed decrease of predictive performance.

Therefore throughout the competition the single flat feature subset representation was used and then to meet the solution requirement of being represented in a form of exactly 10 feature subsets a simple yet robust feature distribution method was used. This method exploited the property of the greedy search models which tend to provide the solution in a form of items ordered inline with their quality or contribution to the group performance. What it means is that the features added first and "surviving" in the solution subset tend to be the best while items added last are likely to be individually the worst. To distribute the predictive power of features evenly among the ensemble subsets taking from the top (best) to bottom (worst) the ensemble subsets were appended in turns until all selected features were distributed. As a result the ensemble was composed of different feature subsets that shared similar predictive power and the size difference between the least and most populous subset was at most 1 feature.

VI. Summary of Experiments

The experiments followed the typical competition journey of trying initially simple models, reflecting on the results, and gradually adding more and more complexity in a search for performance improvement. There were many more feature selection methods tested beyond the one reported above. Among the most significant were feature selection with decision trees reported in [10] and the acclaimed fast binary feature selection with conditional mutual information reported in [8]. None of these alternatives came close to the performance achieved by our top DGBFS method.

Fig. 6 illustrates the comparison of performance curves corresponding to different feature selection methods investigated in the paper.

What is striking is how efficient greedy forward search initially was. With just a few features it achieved really impressive performance. However this effect is achieved at the price of really slow processing and in the longer run suboptimal performance caused by the traps of falling into local maximum. The sparse feature voting method was by far the fastest as it effectively eliminated the Naive Bayes classifier, yet still managed to deliver very high predictive performance. The diversified greedy backward forward method performed relatively fast as it absorbed many suboptimal but good features and stabilised with a very robust solution after just few forward and backward rounds. It had reported the best predictive performance and was submitted as proposed solution to the AAIA'14 data mining competition.

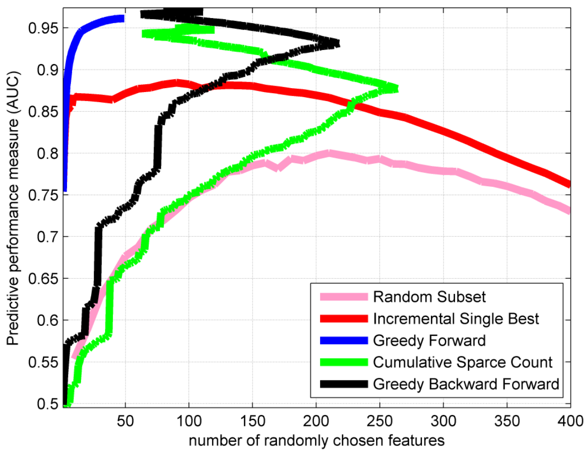

Fig. 6. Feature selection performance curves comparisons

VII. Conclusions

In this work we illustrated relatively simple yet very robust and generic feature selection method that appears to be particularly suitable for handling very large data sets of high-dimensional sparse features. The method employs highly diversified backward-forward search that is relatively fast yet allows to achieve deep features complementarity and very high and stable predictive performance. We have drafted the journey leading to the development of the top model and included informative and comparative examples of other feature selection models some of which could be good candidates for specific predictive requirements. Specifically we have shown that greedy forward search could be a very good model for a very limited number of features, while the binarised features voting method due to its extremely high speed looks to be particularly suitable for live, dynamic predictive system applications with the real-time requirement.

All of the presented feature selection models were considered for an entry in the AAIA'2014 Data Mining competition. Since the predictive performance is the only metric for the competition, the top performing model of diversified greedy backward-forward search has been applied to the data and its solution submitted as our entry in the competition. This solution scored the second place with the tested predictive performance in excess of 96%, just a quarter of the per cent behind the top scored solution. This achievement proves how important is the feature selection step and how much it can reduce the useful input data to provide huge improvements in predictive performance. In the end the model selected only 79 features from the total pool of nearly 12000 thereby elimination more than 99% of data.

The presented model could be used to better understand and prevent various accidents and complex hazardous situations. It really is a good example of how predictive analytics turned big data into small data to potentially save many lives.

References

[1] B. Franks. *Taming The Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics,* Wiley, Hoboken, NJ; 2012.

[2] V. Mayer-Schonberger and K. Cukier. *Big Data: A Revolution That Will Transform How We Live, Work, and Think,* John Murray Poblishers, London; 2013.

[3] T. Davenport. *Big Data at Work: Dispelling the Myths, Uncovering the Opportunities,* Harvard Business Review Press, Boston; 2014.

[4] M. Banko and E. Brill. "Scaling to Very Very Large Corpora for Natural Language Disambiguation," In Proceedings. of the 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001), pp 26–33, 2001.

[5] Y. Bengio. "Learning Deep Architectures for AI," Foundations and Trends in Machine Learning 2(1): 1–127, 2009.

[6] E. Diaz-Aviles, W. Nejdl, L. Drumond and L. Schmidt-Thieme. "Towards real-time collaborative filtering for big fast data," In Proceedings of the 22nd International Conference on World Wide Web companion 2013, pp 779–780, 2013.

[7] L. Yu and H. Liu. "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution," In Proceedings of the 20th International Conference on Machine Learning, pp 856–863, 2003.

[8] F. Fleuret and I. Guyon. "Fast Binary Feature Selection with Conditional Mutual Information," Journal of Machine Learning Research 5: 1531–1555, 2004

[9] H. Liu and Lei Yu. "Toward Integrating Feature Selection Algorithms for Classification and Clustering," IEEE Transactions on Knowledge and Data Engineering 17(4): 491–502, 2005.

[10] C. Ratanamahatana and D. Gunopulos. "Feature Selection for the Naive Bayes Classifier Using Decision Trees," Applied Artificial Intelligence 17: 475–487, 2003.

[11] T. Mitchell. "Generative and discriminative classifiers: naive bayes and logistic regression," in Machine Learning, McGraw Hill, 2010.

[12] J.H. Friedman. "Greedy Function Approximation: A Gradient Boosting Machine," Annals of Statistics 29: 1189–1232, 2000.

[13] Z. Zhi-Hua. *Ensemble Methods: Foundations and Algorithms,* Chapman & Hall / CRC Press, Boca Raton, FL; 2012.