SHORT COMMUNICATION
63rd STAPP CAR CRASH CONFERENCE

SC19-01

Machine Learning Based Model for Predicting Head Injury Criterion (HIC)

Vikas Hasija
Bowhead (Systems & Technology)

Erik G. Takhounts
National Highway Traffic Safety Administration (NHTSA)

ABSTRACT – The objective of this study is to develop a machine learning based predictive model from the available crash test data and use it for predicting injury metrics. In this study, a model was developed for predicting the head injury criterion, HIC_{15}, using pre-test features (vehicle, test, occupant and restraint related). This problem was solved as a classification task, in which HIC_{15} with a threshold of 700 was divided into three classes i.e. low, medium and high. Crash test data was collected from the NHTSA database and was split into training and test datasets. Predictive models were developed from the training dataset using cross-validation while the test dataset was only used at the final step to evaluate the chosen predictive model. A logistic regression based predictive model was chosen as it demonstrated minimal overfitting and gave the highest F1 score (0.81) on the validation dataset. This chosen model gave a F1 score of 0.82 on the test (new/unseen) dataset.

INTRODUCTION

NHTSA has commissioned various types of research, New Car Assessment Program (NCAP), and compliance crash tests over the years. Information on these crash tests has been recorded in the NHTSA vehicle crash test database, which contains both pre-test and post-test information on the vehicle, occupant i.e. Anthropomorphic Test Device (ATD) along with the type of test, restraints and the injury metric values measured on the ATD. These data, in the past, have been analyzed extensively to investigate correlations between different variables, understand the injury trends, etc. but has never been used for predictive modeling. Considering both the cost of conducting a physical crash test and budget limitations on the number of tests that NHTSA can carry out, it would be helpful if injury metrics for the new vehicles could be predicted to help NHTSA decide which vehicles to assess under the agency's crash test program. Thus, the objective of this study is to a) develop a machine learning based predictive model for predicting injury metric using only the pre-test information from the available crash test data, and b) evaluate the performance/predictability of the model by using it to predict injury metric on new/unseen data. In this study, a machine learning based predictive model was developed for the Head Injury Criterion (HIC_{15}) as most data were available for this injury metric.

METHODS

Data

For developing the predictive model for HIC_{15}, crash test data from Oblique, NCAP, and Federal Motor Vehicle Safety Standard (FMVSS) 208 tests were collected from the NHTSA vehicle crash test database for the years 1989 to 2018. These tests were further filtered to include only the driver for a total of 1600 tests. For each of these tests, pre-test features related to the vehicle, test, occupant (ATD) and restraints were collected along with the injury metric value (Table A1, Appendix A).

Predictive Model Development

A classification based predictive model was developed in this study. Open source python based machine learning library Scikit-Learn (Pedregosa et al, 2011) was used. HIC_{15} was divided into three classes, i.e. low (0-400, class1), medium (401-700, class2), and high (>700, class3). The dataset was split into training (70%) and test (30%) datasets. The composition of each dataset was 82% low class, 15% medium class and 3% high class. Since the dataset is class imbalanced, F1 score (Scikit-learn user guide, 2019, section 3.3) was used as the evaluation metric.

Of the twenty-five pre-test features, one was categorical-ordinal, eleven were categorical-nominal, and thirteen were continuous (Table A1, Appendix A). Since machine learning algorithms only work with numerical data, feature transformation was carried out for the categorical features in the training dataset. All the categorical-nominal features were one-hot encoded (Albon, 2018a), which is essentially a representation of categorical-nominal features as binary vectors. Model year which was treated as a categorical-ordinal variable was integer encoded (Albon, 2018a). All continuous features were normalized to be on the same scale.

Address correspondence to Vikas Hasija, 1200 New Jersey Ave. SE, Washington, DC. Electronic mail: vikas.hasija.ctr@dot.gov

Following feature transformation, feature selection was carried out to find the most important features that affect the target variable (HIC_{15}). A univariate feature selection scheme (Brownlee, 2018a) was used for this purpose.

Before using advanced machine learning algorithms, a simple baseline classifier (Albon, 2018b) was evaluated. The baseline classifier makes predictions using simple rules and does not learn anything from the data. This classifier is useful as a simple baseline to compare with other advanced classifiers. In this study, the baseline classifier that predicts the "most frequent class" was set up and its F1 score was evaluated. This score was compared with the F1 score of advanced machine learning algorithms to make sure that these advanced algorithms were scoring higher thus indicating that these algorithms were learning from the data.

Various advanced machine learning algorithms were trained to find the algorithm that gave the best predictive model. The machine learning algorithms used in this study were Logistic Regression (Brownlee, 2018b; Scikit-learn user guide, 2019, section 1.1), Support vector machines (Brownlee, 2018c; Scikit-learn user guide, 2019, section 1.4), and Decision Tree (Breiman, 1984; Scikit-learn user guide, 2019, section 1.10). In addition, ensemble methods namely Random forest (Breiman, 2001; Scikit-learn user guide, 2019, section 1.11), and Extreme Gradient Boosting (Chen et al, 2016) were also used.

Each of these algorithms has hyperparameters, which are parameters external to the model and cannot be learned from the data. These hyperparameters must be tuned to find the best model. For hyperparameters tuning, training was carried out with cross-validation, which involves splitting the training dataset further into two sets namely training set and validation set. The training set is used to train the model. This trained model is then evaluated on the validation set to find the best hyperparameters.

Best predictive model selection
The model that gave the highest F1 score on the validation dataset with least overfitting was selected as the best predictive model.

Model evaluation on test (new/unseen) dataset
The selected predictive model was evaluated on the test dataset. Before evaluation, feature transformation (one-hot and integer encoding, normalization) was performed on the test dataset and the same important features that were selected from the training dataset using feature selection scheme were selected from the test dataset. F1 score was then computed along with the classification report and confusion matrix (Scikit-learn user guide, 2019, section 3.3) to evaluate the model's performance on this test (new/unseen) dataset.

Model Stability
Stability of a learning algorithm refers to how sensitive the output of the model is to changes in the training dataset. A learning algorithm is stable if the learned model does not change much when training data is changed. In this study, the selected predictive model was evaluated for stability by changing the random seed responsible for splitting the data into training and test datasets. Twenty-five different random seeds were evaluated and F1 score was recorded each time on the test dataset to judge the stability.

RESULTS

Feature Selection
Feature selection was carried out to find the minimum number of features that gave maximum performance. Table 1 shows a set of eight features that was used in this study.

Table 1. Selected Features

1.	Model year	5.	Frontal airbag
2.	Closing speed	6.	Occupant size
3.	NCAP test	7.	HH (Appendix A, Figure A1)
4.	Oblique test	8.	HD (Appendix A, Figure A1)

Predictive Models
The baseline classifier that was set up to predict the "most frequent" class gave a weighted F1 score of 0.74 for both the training and the test datasets. Advanced machine learning algorithms were trained with cross-validation using the selected features. The training and validation weighted F1 scores are presented in Table 2. Each model scores better than the baseline classifier showing that the models are learning from the data.

Table 2. Training and validation F1 scores

Machine Learning Algorithm	Training F1 score	Validation F1 score
Logistic Regression	0.81	0.81
Support vector classifier	0.80	0.79
Decision tree	0.81	0.78
Random Forest	0.82	0.79
Extreme Gradient Boosting	0.82	0.79

Best predictive model selection
The logistic regression model showed the highest F1 score on the validation dataset and the least overfitting. Hence, it was selected as the best predictive model.

Model evaluation on test (new/unseen) dataset
The selected logistic regression model was evaluated on the test dataset. A weighted F1 score of 0.82 was obtained on the test dataset. Table A2 (Appendix A) shows the classification report and confusion matrix for the logistic regression model on the training and test datasets.

Model Stability
The selected logistic regression model was evaluated for stability. The mean F1 score on the test dataset was found to be 0.803 with a standard deviation of 0.009.

DISCUSSION

The objective of this study was to explore the possibility of training machine learning algorithms on available crash test data (pre-test features) to learn patterns from the data and use this learned model to predict injury metric (HIC_{15}) for new/unseen data. It can be observed from the results that machine learning algorithms were able to learn and find patterns from the data. This is clear from the F1 score comparison between the baseline classifier (0.74) and logistic regression classifier (0.81) on the training dataset. The baseline classifier is a simple model that does not learn anything and thus gives zero F1 score for all classes except class1 ('most frequent class'). However, the logistic regression model was able to learn from the data about all three classes as demonstrated by the confusion matrix and F1 scores in the classification report (Table A2, Appendix A).

The logistic regression model learns but not equally well about the three classes. There are 1600 data points which are not uniformly distributed among the three classes. There is more data for class 1 than class 2 or class 3 and as such the learning algorithm was able to learn more about class 1 than class 2 or class 3. This resulted in the ability of the learning model to identify more relevant instances of class 1 (89%) than class 2 (48%) or class 3 (53%) in the test (new/unseen) dataset (Table A2, Appendix A). For the same reason, higher precision was obtained for class 1 (91%) than class 2 (45%) or class 3 (42%) (Table A2, Appendix A) on the test (new/unseen) dataset.

All algorithms used in this study are nonlinear except logistic regression. Logistic regression, being a linear model, has fewer degrees of freedom compared to other algorithms used in this study and is thus less prone to overfitting. Complex models tend to overfit, which can be controlled by tuning the hyperparameters/degrees of freedom of the algorithm. Hyperparameter tuning was conducted but the nonlinear algorithms still showed slightly more overfitting compared to logistic regression.

CONCLUSION

The predictive model developed in this study showed promising results and is only limited by the amount of data. Model predictability was excellent for the low HIC_{15} class as it had the most available data. For the medium and high HIC_{15} classes, the model predictability may be further improved if more data is generated/obtained for these classes.

REFERENCES

Albon, C. (2018a) Machine Learning with Python Cookbook, pp. 81. O'Reilly Media, Inc.

Albon, C. (2018b) Machine Learning with Python Cookbook, pp. 184. O'Reilly Media, Inc.

Breiman L, Friedman J, Olshen R, and Stone C. (1984) Classification and Regression Trees, Wadsworth, Belmont, CA.

Breiman L. (2001) Random Forests, Machine Learning, 45(1), 5-32.

Brownlee, J. (2018a) Machine Learning Mastery with Python. pp. 53. Machine Learing Mastery.

Brownlee J. (2018b) Master Machine Learning Algorithms, Pages 52-61, Machine Learning Mastery.

Brownlee J. (2018c) Master Machine Learning Algorithms, Pages 116-125, Machine Learning Mastery.

Chen T., Guestrin, C. (2016) XGBoost : A Scalable Tree Boosting System. https://arxiv.org/abs/1603.02754

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

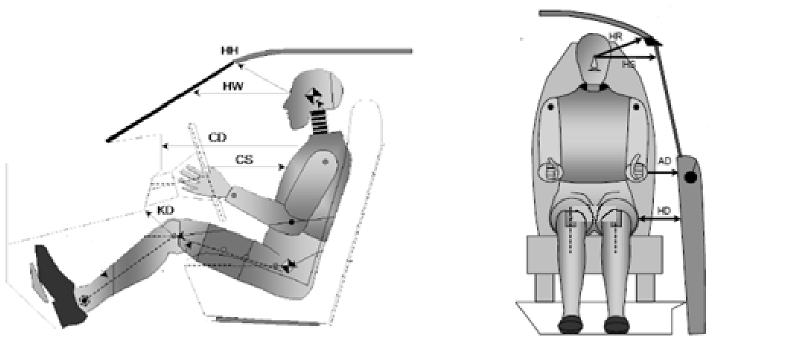
Scikit-learn User Guide (2019).

Appendix A

Table A1. Test information collected

Test related	**Vehicle related**	**ATD related**	**Restraint related (Used/Not used)**	**Injury Metric**
Test type (CN) (NCAP, Oblique, 208)	Weight (CS)	Occupant type (CN) (THOR, H-III)	Seatbelt (CN)	HIC_{15}
Closing speed (CS)	Length (CS)	Occupant size (CN) (50th, 5th)	Frontal airbag (CN)	
PDOF (CN)	Width (CS)	Seat Position (CN) (Center, Rearward, Forward)	Side curtain airbag (CN)	
	Body type (CN) (Sedan, Coupe, etc.)	Clearance measurements (CS) (Appendix A, Figure A1)	Knee airbag (CN)	
	Model year (CO)		Torso/pelvis airbag (CN)	

*Feature Types: CN: Categorical-Nominal, CS: Continuous, CO: Categorical-Ordinal



CD:	Chest to Dash
CS:	Chest to Steering wheel hub
HW:	Head to Windshield
HH:	Head to Header
KD:	Knee to Dash
AD:	Arm to Door
HD:	H-Point to Door
HR:	Head to Side Header
HS:	Head to Side Window

Figure A1. Clearance measurements

Table A2. Logistic regression predictive model results

Dataset	**Classification report**					**Confusion matrix**			
Training	class	precision	recall	f1-score	support	class	1	2	3
	1	0.90	0.89	0.89	915	1	**811**	93	11
	2	0.42	0.44	0.43	171	2	80	**75**	16
	3	0.37	0.47	0.42	34	3	7	11	**16**
	weighted avg	**0.81**	**0.81**	**0.81**	**1120**				
Test	class	precision	recall	f1-score	support	class	1	2	3
	1	0.91	0.89	0.90	392	1	**347**	40	5
	2	0.45	0.48	0.46	73	2	32	**35**	6
	3	0.42	0.53	0.47	15	3	4	3	**8**
	weighted avg	**0.82**	**0.82**	**0.82**	**480**				