Automobile Crashes: Life or Death

Ian Colvin
Computer Science
California State University,
Sacramento
Sacramento, CA USA
nai.colvin@gmail.com

Minquan Li
Computer Science
California State University,
Sacramento
Sacramento, CA USA
minquan.li@outlook.com

James Tath
Computer Science
California State University,
Sacramento
Sacramento, CA USA
Jameschantath@csus.edu

ABSTRACT

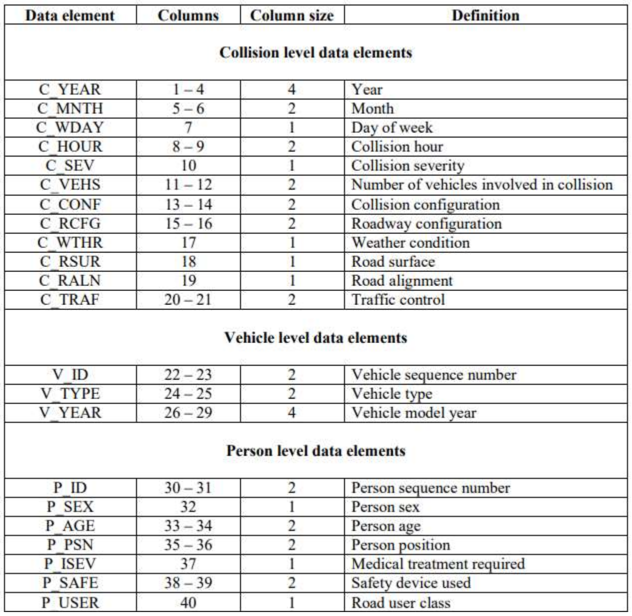
There are nearly 1.25 million people who die in automobile crashes pear year, which is an average of 3,287 deaths a day. According to ASIRT [1], automobile crashes are listed at the ninth highest leading cause of deaths and account for 2.2% of all deaths globally. Automobile crash fatalities are certainly a tragic event that is very difficult to predict, and are highly spontaneous that can highly affect one's life permanently. With the best to our abilities, we will work to figure out the highest contributing factors of fatalities, and try to emphasize these features to help prevent future fatalities by teaching others what to be more wary about. To solve the problem, we used a fully connected neural network, and a convolutional neural network to help train a dataset of crashes with a training set of 223960 and testing set of 74654. Then with further regularization of the dataset, we were able to find out the highest leading features that leads to fatalities. Our solution performed a 0.86 F1-score from the CNN model, and when compared to a paper that did a study on the same topic landed with a 0.86 F1-score from a CNN model. The area we outperformed their approach is that we took on a focus to find out which specific features were the highest contributing factors towards fatalities in crashes, while they focused a lot more on which was the best model.

1 Introduction

The cause of automobile fatalities can have a wide range of factors that are too complicated for humans to determine which the most important ones are. We can use deep learning to predict a fatality in an automobile crash and then extract the important features from the model. This paper will start by showing our dataset and algorithms for our model. From there we will go into our experimental results. Lastly, we will reflect on this paper by sharing our resources and what we learned during the process.

2 Problem Formulation

A dataset from https://www.kaggle.com/tbsteal/canadian-car-accidents-19942014/ was used. The dataset contained automobile accident data in Canada from 1999-2014. This dataset contains various information regarding traffic accidents (See figure below)



Data element	Columns	Column size	Definition
Collision level data elements			
C_YEAR	1 – 4	4	Year
C_MNTH	5 – 6	2	Month
C_WDAY	7	1	Day of week
C_HOUR	8 – 9	2	Collision hour
C_SEV	10	1	Collision severity
C_VEHS	11 – 12	2	Number of vehicles involved in collision
C_CONF	13 – 14	2	Collision configuration
C_RCFG	15 – 16	2	Roadway configuration
C_WTHR	17	1	Weather condition
C_RSUR	18	1	Road surface
C_RALN	19	1	Road alignment
C_TRAF	20 – 21	2	Traffic control
Vehicle level data elements			
V_ID	22 – 23	2	Vehicle sequence number
V_TYPE	24 – 25	2	Vehicle type
V_YEAR	26 – 29	4	Vehicle model year
Person level data elements			
P_ID	30 – 31	2	Person sequence number
P_SEX	32	1	Person sex
P_AGE	33 – 34	2	Person age
P_PSN	35 – 36	2	Person position
P_ISEV	37	1	Medical treatment required
P_SAFE	38 – 39	2	Safety device used
P_USER	40	1	Road user class

All the features except for C_SEV and P_ISEV are used as inputs. C_SEV is the classification if there is a fatality that occurred during the accident or not and is used for our output.

3 System/Algorithm Design

3.1 System Architecture

There was quite a bit of preprocessing that had to be done with the data such as configuring many NaN datas within the set, then one hot encoding all the categorical data in the set, and normalizing it all afterwards. The set had a serious problem with it being completely skewed towards non-fatal crashes withholding 97% of the data to only 3% of data being fatal crashes. Therefore, to fight that we had to balance out the data to become a little bit more even in those terms which led to cutting the dataset quite the amount. After all, of this preprocessing we then pushed the data through two models, which both provided similar results, showing a strong evaluation of our model. Then with a Lasso regularization, we were able to pinpoint the highest contributing features that lead to fatalities

3.2 Module 1

3.2.1 Data Pre-processing

In the dataset, there are duplicated data and they were all dropped. All of the unknown values are labeled using the letter “U”, “X”, “N”, or “Q”. Because the dataset is extremely unbalanced, so all nonfatal rows contains any unknowns were dropped. Next is to balance the dataset. We decided to down sample the nonfatal because it will be easier for the model to train on less records. The fatal records also have many unknown values. Instead of dropping data with unknowns, we filled in the missing value with the mean if the data column was a range of numbers. And we are treating the unknown values in the categorical data as a new feature and assigned them all with a new category. Next, we combined and shuffled the fatal and nonfatal dataframes and we one hot encoded all of the categorical features before normalizing all of the non-categorical ones.

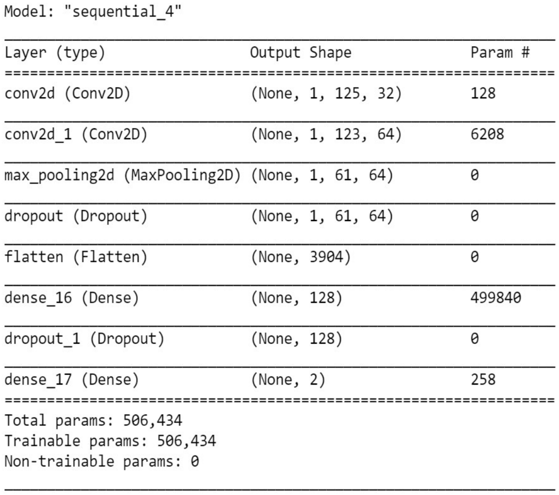
3.3 Module 2

3.3.1 Data Training

We split the processed dataset to 25/75 for testing/training.

a) FCNN is our first model. We used 3 dense layers and one output layer with activation relu and optimizer adam. We looped the training 5 times and pick the best one for the model.

b) CNN is our second model. We set the batch size to 128 and we did one round of conv2D, Maxpooling, Dropout, and Flatten with a dense of 128. The output has the activation of softmax.



```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 1, 125, 32)        128
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 1, 123, 64)        6208
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 1, 61, 64)         0
_________________________________________________________________
dropout (Dropout)            (None, 1, 61, 64)         0
_________________________________________________________________
flatten (Flatten)            (None, 3904)              0
_________________________________________________________________
dense_16 (Dense)             (None, 128)               499840
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_17 (Dense)             (None, 2)                 258
=================================================================
Total params: 506,434
Trainable params: 506,434
Non-trainable params: 0
_________________________________________________________________
```

3.4 Module 3

3.4.1 Regularization

We used the L1 regularization for our classification problem. Lasso regularization can tell us what features are most important for our machine learning model, so we can drop unnecessary features or learn what features are important.

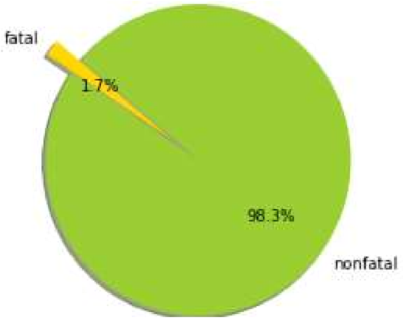
4 Experimental Evaluation

4.1 Methodology

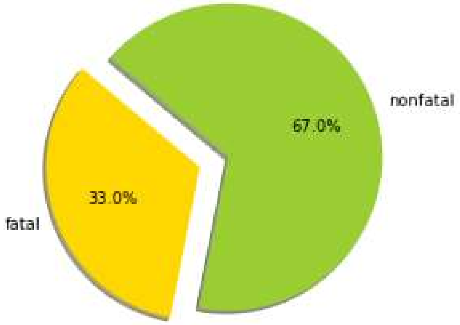
All of the data was used, except for P_ISEV. P_ISEV is similar to C_SEV as it classifies if the accident had a fatality or not, thus, the data is a duplication. The rest of the data is split into 75% training data and 25% testing data. We down-sampled the non-fatalities to balance the dataset. Different optimizer and activation parameters will be tested, along with different kernel sizes for the CNN model. Since this problem is a classification problem, average F1-Score, average precision, and average recall will be used for our metrics when comparing models. The baseline metrics have been taken from the paper cited in related work (F1-Score of 0.87). We will then compare a fully connected neural network and a convolutional neural network (CNN). Once we find the optimal parameters for our models, we will extract the key features from our model using regularization. This should leave us with the highest contributing factors to automobile accidents.

4.2 Results

Our data before down sampling:

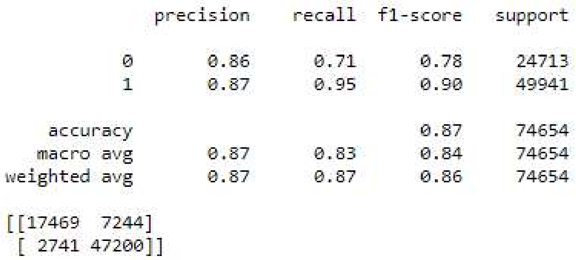


Our data after down sampling:

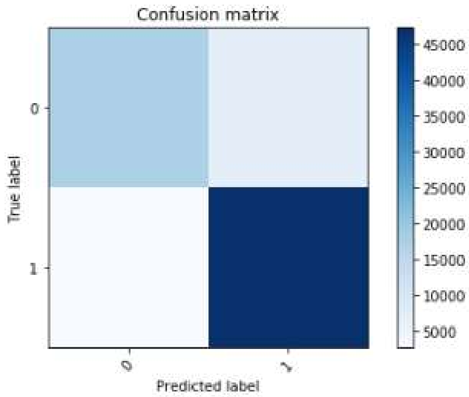


We found it better for us to down sample our data since the dataset had over five million records. Having sufficient amount of fatal records to train our model, the non-fatal records were not needed.

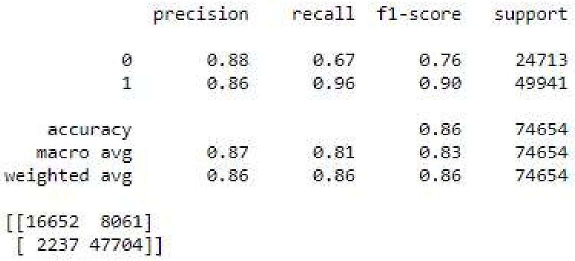
Fully connected neural network:



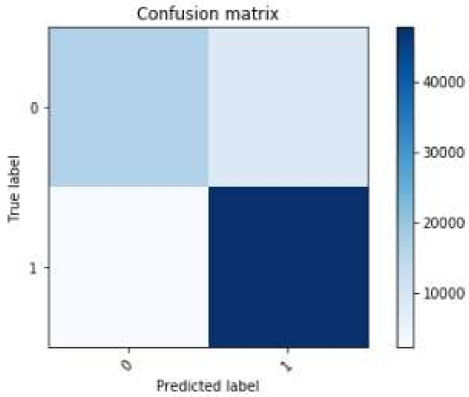
```
              precision    recall  f1-score   support

           0       0.86      0.71      0.78     24713
           1       0.87      0.95      0.90     49941

    accuracy                           0.87     74654
   macro avg       0.87      0.83      0.84     74654
weighted avg       0.87      0.87      0.86     74654

[[17469  7244]
 [ 2741 47200]]
```



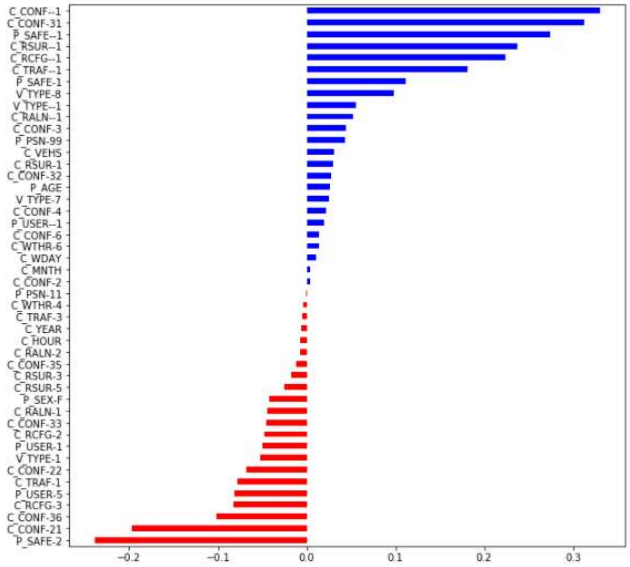
CNN:



```
              precision    recall  f1-score   support

           0       0.88      0.67      0.76     24713
           1       0.86      0.96      0.90     49941

    accuracy                           0.86     74654
   macro avg       0.87      0.81      0.83     74654
weighted avg       0.86      0.86      0.86     74654

[[16652  8061]
 [ 2237 47704]]
```



Both the fully connected network and CNN used an activation of relu and optimizer of adam. The two models performed similar to eachother (F1-Score: 0.86-0.87) and almost exactly the same as the baseline (F1-Score: 0.87). After the two models were tuned, the features were then extracted from the CNN model using Lasso regularization.



The highest contributors to fatalities came out to be what our intuition tells us. C_CONF-1 is a single vehicle hiding a person or animal, C_CONF-31 is a head on collision, P_SAFE-1 is no safety device used (Such as a seat belt or car seat). The other top features are normal driving conditions, such as, dry roads and working traffic lights. These results are most likely due to driving habits and drivers being less careful in normal conditions. The full feature list can be found at https://www.kaggle.com/tbsteal/canadian-car-accidents-19942014/#drivingLegend.pdf

5 Related Work

A paper entitled "Traffic Accident's Severity Prediction: A Deep-Learning Approach-Based CNN Network" speaks about the past approaches to this same problem. The paper mentions that statistical methods and machine learning methods are most common. The previous machine learning methods have used GA and MLP structural modeling; the best-fit model had an R-value of 0.87. Out of all the models that have been created to predict fatalities, an RNN model achieved the best accuracy of 71.77%.

The paper states that the problem is that the features of the dataset are seen individually and the previous models do not maintain a relationship between the features. The solution to this problem is to use a CNN network, the same solution that we have decided to use.

While our method is similar, we are using a dataset from a different area of the world. The paper gives us a baseline to compare to when we create our model since their highest F1-Score is 0.87. The purpose of their model is not to come up with the best metrics, but rather compare the CNN model to other models [2].

6 Conclusion

In conclusion, our CNN model performed the same as the research paper that was used for a baseline. However, we learned that when dealing with the issue of automobile fatalities, the factors change from area to area. Our dataset also did not include features such as drunk driving or distracted driving. The results were not surprising and line up with our intuition on what might cause a fatal accident.

7 Work Division

All three members researched and decided on the subject material and the initial plan for the project. We found a dataset that works for us and then decided on which models we will be using for comparison. Minquan Li did the data pre-processing and the fully connected network. James Tath coded the CNN model and the metrics for both models. Ian Colvin produced the report. All three members worked on the presentation slides together

8 Learning experience

We have gained invaluable experience with this project. Our learning process is a reflection of what we learned in our CSC 180 class at CSUS. During our research of the problem, we found that artificial intelligence could be used to solve a wide range of problems. It is a matter of choose the right model then tuning the parameters.

Automobile crashes is a major problem and has been studied since automobiles have been around. Our instinct for this problem was to use a CNN deep learning network and reading other papers that say the same thing lets us know that we are on the right track.

REFERENCES

[1] "Road Safety Facts." *Association for Safe International Road Travel*, www.asirt.org/safe-travel/road-safety-facts/. 2019

[2] [2] Zheng Ming, "TASP: Deep-Learning Approach-Based CNN Network", IEEE Access, Vol. 7, (2019), pp. 39897-39910.