Predicting Severity of Car accidents in Seattle

Author: Hugo Cosme
Contact: hualcosa@gmail.com

The problem with car accidents

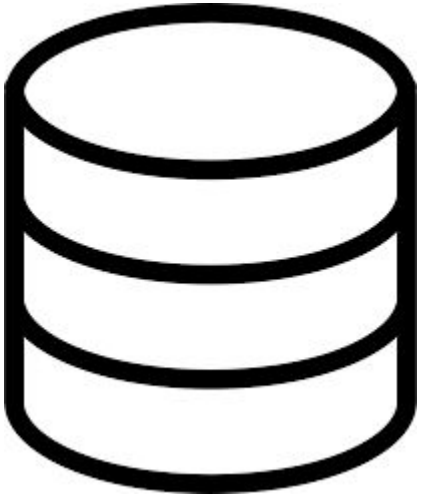
According to a report, in 2017 there were a total of 187 fatal and serious injury collisions on Seattle streets. Data available from the Washington State Department of Transportation (WSDOT) reflect an even worse tally in 2018, with 212 crashes that resulted in serious injury or wrongful death.

When a severe accident happens, crash victims have a better chance of recovery, or avoiding death, if they receive quick medical treatment at the scene of an injury (often referred to as 'the golden hour'). For this reason, improving first aid skills for the general public is also a good way to improve survivability after a crash has occurred.

Would it be possible to improve emergency response?

The answer is yes! Using records from the Seattle Department of Transportation, a model could be built using statistical and machine learning techniques. That's what I did and I will be presenting the details to you right now.

Data source



As mentioned before, the dataset that was used to build the model comes from the Traffic Management Division of the Seattle Police Department and contains accident records that range from 2004 up to the present. The Dataset contains 194673 rows, 37 feature columns and one target column, SEVERITYCODE.

SEVERITYCODE is a numeric column that contains a code which corresponds to the severity of the collision. The possible values are:

- 1 means 'property damage'.
- 2 means 'injury'.

Feature	Description
X	x coordinate of the accident in the map
Y	y coordinate of the accident in the map
ADDRTYPE	Collision address type
SDOT_COLCODE	
COLLISIONTYPE	type of collision
PERSONCOUNT	The total number of people involved in the collision
PEDCOUNT	The total number of pedestrians involved in the collision
PEDCYLCOUNT	The total number of bicycles involved in the collision
VEHCOUNT	The total number of vehicles involved in the collision
INATTENTIONIND	Whether or not the collision was due to inattention
UNDERINFL	Whether or not a driver involved was under the influence of drugs or alcohol
WEATHER	A description of the weather conditions during the time of the collision
ROADCOND	The condition of the road during the collision
LIGHTCOND	The light conditions during the collision
SPEEDING	Whether or Not speeding was a factor in the collision

Feature selection

From the 37 feature columns, I initially selected the ones on the left based on their meaning.

Data preprocessing

The following processing steps were made:

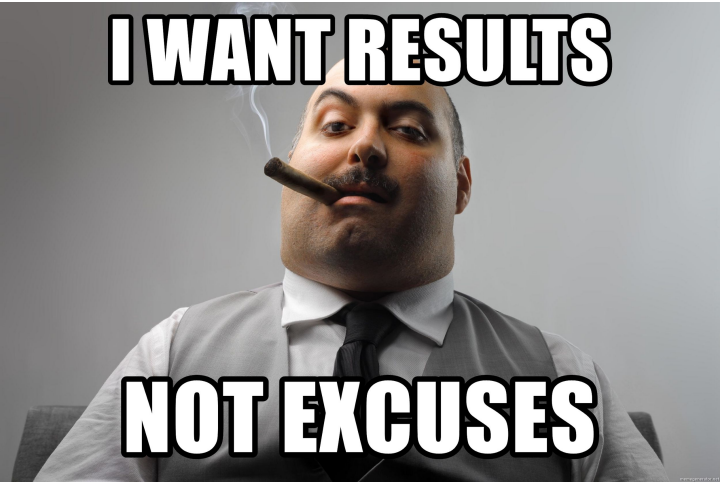
1. Remove the columns that were mainly comprised of missing values.
2. The remaining rows with missing values were dropped.
3. The values in the “UNDERINFL” were uniformed, because there were different labels for the same category(1 means the same as Y and 0 means the same as No).
4. The categorical features were encoded using One Hot Encoding.
5. Finally, columns “X” and “Y” were standardized to allow faster training of the model.

Predictive modeling

I decided to use a XGBClassifier, which is an ensemble machine learning model. This means that it consists of several models(yes! models within the model) that are called base learners, which have an accuracy slightly better than chance. The output of each of these models count as a vote. These votes are counted and the majority is the final output.
Besides that, the 'stratify' parameter of the train_test_split function was used to separate the data into equal proportions of values (1s and 2s labels in our case) in both the training and test set. This diminishes the chances of our model becoming biased towards one or another label.

The dataset was split into training and test data, using 20% of total rows as the test set. I did some hyperparameter tuning and built a XGBClassifier with the best parameters found. Finally the model was fitted using the training set.

Results



The model got a f1-score of approximately 84% in the test set, even though it has never seen the data before. This is a good result for first a prototyping solution!!

the model is really accurate in predicting type 1 accidents(predicts 24961 correctly and only 669 wrongly) but it is not that good in predicting type 2(8323 incorrect predictions against 2877 correct predictions).

One of the reasons that might explain this biased performance is the lesser amount of training examples for label 2 in relation to label 1. This can be noticed in the modeling section of the jupyter notebook.

Future improvements

In order to improve the model's performance I would try to get additional data of accidents of type 2. Hence, it would be possible to produce a less biased model without sacrificing the predictive capacity of accidents of type 1.

Another strategy is to use natural language techniques from the original data set in order to extract meaningful features for the prediction.

This presentation was made as an assignment of the Capstone Project course in IBM Data Science Professional Certification.

A Jupyter notebook with all the manipulations that were made to the data and a technical report can be found in my github page:

https://github.com/hualcosa/Coursera_Capstone