Predicting Severity of Car Accidents in Seattle

Hugo Albuquerque Cosme da Silva

September 30, 2020

1. Introduction

When a Traffic accident happens the difference between saving or losing lives lies in the capability of delivering the proper response in the shortest period of time possible. Thus the idea of this project is to build a model that is capable of predicting the degree of severity of the accident allowing stakeholders(paramedics, police, firefighters, hospitals, etc) to set up the infrastructure as agile as possible. This way, the goal is to minimize the loss of lives and treat the injured as effectively as possible.

2. Data Acquisition and Preprocessing

2.1 - Data Source

The dataset that will be 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. The meaning of each of the columns and further metadata can be found in the following link: https://github.com/hualcosa/Coursera_Capstone/blob/master/Metadata.pdf .

3. Methodology

3.1 - Data processing

3.1.1 - Feature selection

After reading carefully the meaning of each of the columns in the metadata file, i preselected the following features:

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

The target variable we want to predict is:

SEVERITYCODE	A code that corresponds to the severity of the collision: 1 = 'property damage'; 2 = 'injury'

3.1.2 - Data preprocessing

The next thing made was to remove some of the columns that were comprised mainly of missing values. Then the remaining rows with missing values were dropped. The values in the "UNDERINFL" were standardized, because there were different labels for the same category(1 means the same as Y and 0 means the same as No). After that, the categorical features were encoded using One Hot Encoding. Finally, columns "X" and "Y" were standardized to allow faster training of the model.

3.1 - 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 hehe) 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. For more information about XGBClassifier click the following link: xgboost documentation.

The dataset was split into training and test data, using 20% of total rows as the test set. 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.

I performed some hyperparameter tuning and built the model with the best parameters I found. I left the code responsible for hyperparameter tuning commented because this may take days(depending on your computing power) if you want to test all values in the grid.

4. Discussion

4.1- 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 a prototyping solution. I used the fitted model to predict the labels in the test set and then computed the F1-score, because it is a solid metric that is capable of evaluating the model's performance in predicting both the positive and the negative labels. Besides that, I built a confusion matrix.

We can see that 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.

5. Conclusion

4.2 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.