Predicting the severity of car accidents

Vitor Domingues Presutti

September 11, 2020

1. Introduction

1.1 Background

The Seattle Department of Transportation's annual traffic report illustrates the constant challenge to the city posed by car accidents. In 2017, Seattle police reported 10,959 motor vehicle collisions on city streets. According to the report, in 2017 there were a total of 187 fatal and serious injury collisions on Seattle streets. According to The Seattle Department of Transportation's besides the number of deaths increased, the number of several injuries has decreased.

1.2 Problem

Car accidents are a big modern society problem and cost millions of dollars to government and affect people's lives. This project aims to predict the severity of an accident based on certain conditions.

1.3 Interest

Government agencies are the biggest interested in this because they can have a better understanding of which conditions are more likely to causa sever accidents and reduce the number of accidents or at least decrease the severity. The people are another interested because nobody wants damages to their properties or unwanted outcomes that affects their lives.

2. Data Source

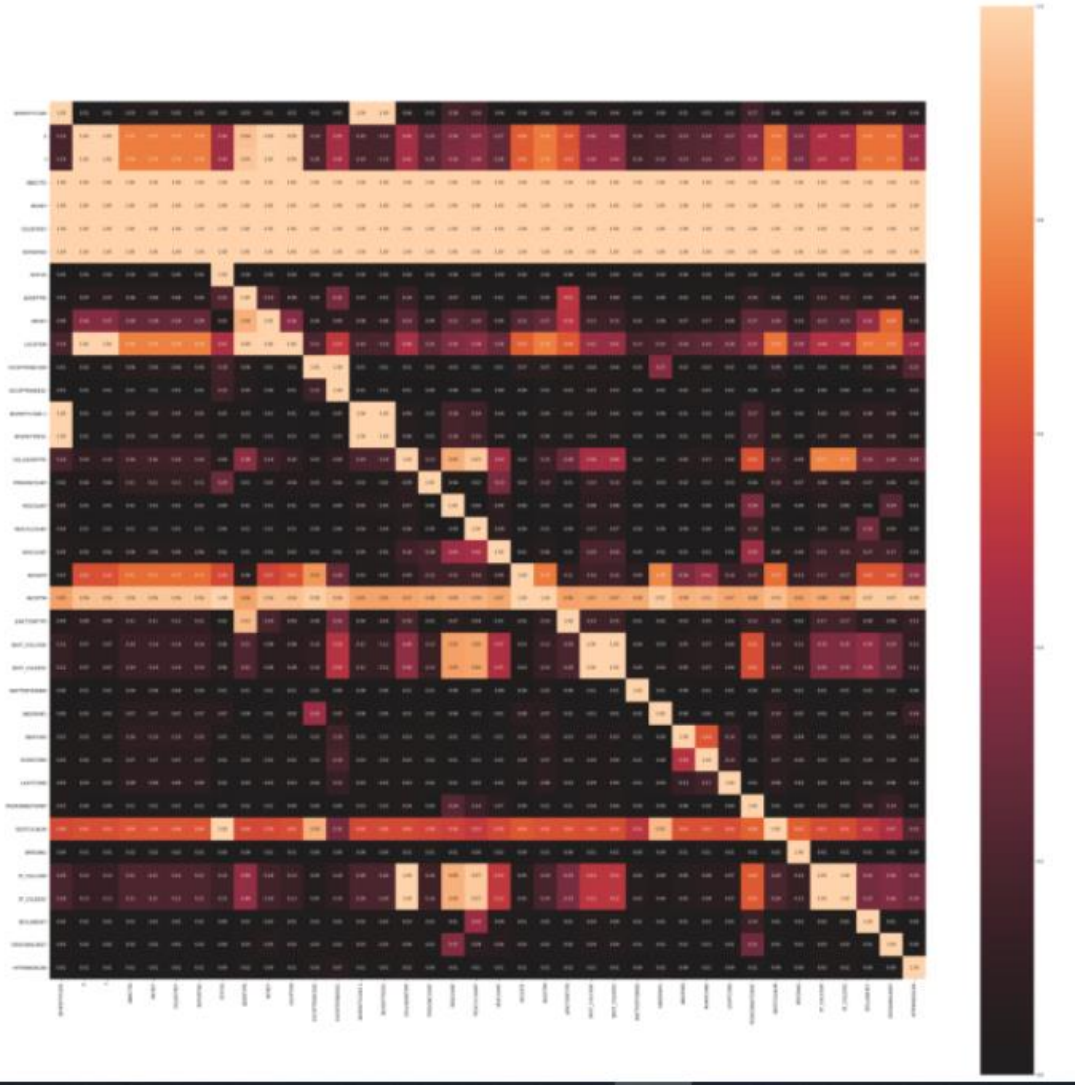
The dataset was provided by The Seattle Department of Transportation's here and the metadata was founded here. The dataset has 38 columns (37 possible features) and 194,673 rows.

3. Methodology

3.1 Data Cleaning and Feature Selection

During the exploratory analysis, some issues were raised. The dataset has lots of missing values throughout the whole dataset. According to the metadata some features are related to coordinates, address, and other unique keys thar government agencies uses to identify each accident. Though other features like road condition, light condition and weather look promising.

First all the missing values from road condition, light condition and weather were dropped. Them all features were converted do strings to analyze the correlation between the features and the accident severity and plotted in the figure below.



All the strongest features were related to keys used to identify the accident by government agencies. The features road condition, light condition and weather show the following results

Feature	Correlation
Road Condition (ROADCOND)	0.027938
Light Condition (LIGHTCOND)	0.028834
Weather (WEATHER)	0.027361

Besides the results indicate a weak correlation all other features had poor information to be useful to this project.

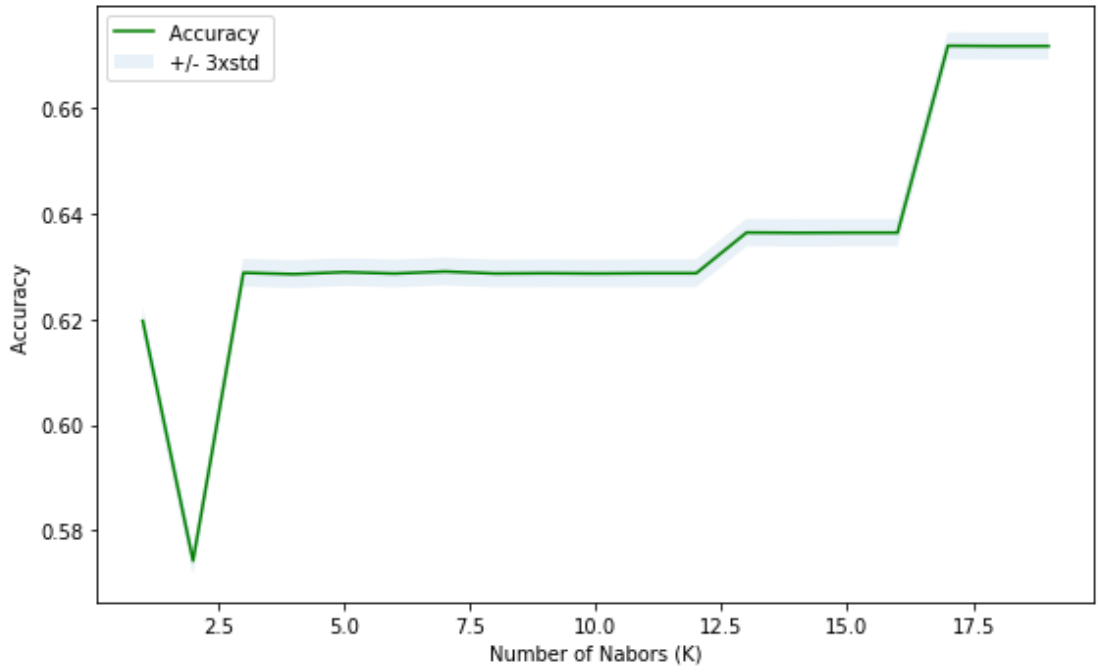
Digging into these three features some labels issues were raised. First some values like “Unknown” and “Other” were found. Second values with the same information but described different like “Dark - No Street Lights”, “Dark - Street Lights Off” and “Dark - Unknown Lighting”. The solution to these issues were define a unique label to the second issue and drop “Unknown” and “Other” values. Using the information provided by the metadata the severity code was all change to strings. During this process was identified that only property damage (value 1) and injury (value 2) were found in the dataset, Although the metadata shown 5 different values:

Severity Code	Description
0	Unknown
1	Prop Damage
2	Injury
2b	Serious Injury
3	Fatality

With the features settled and all similar labels converted we start to setup the features and labels to preprocessing the data and find the best algorithm to our project. One hot encoding was used to covert all string features into integer features. The next step was preprocessing the data and split the dataset into two, one for train the model and other do predict. Considering the goal of this project, the best model to achieve the goal is classification model. That said, we will work with KNN, Logistic Regression, SVM and Decision Tree to find which is best to this problem.

3.2 Model Selection

The first tested algorithm was KNN. A loop to test different K values up to 20 were generated to find the best K and the results are the following

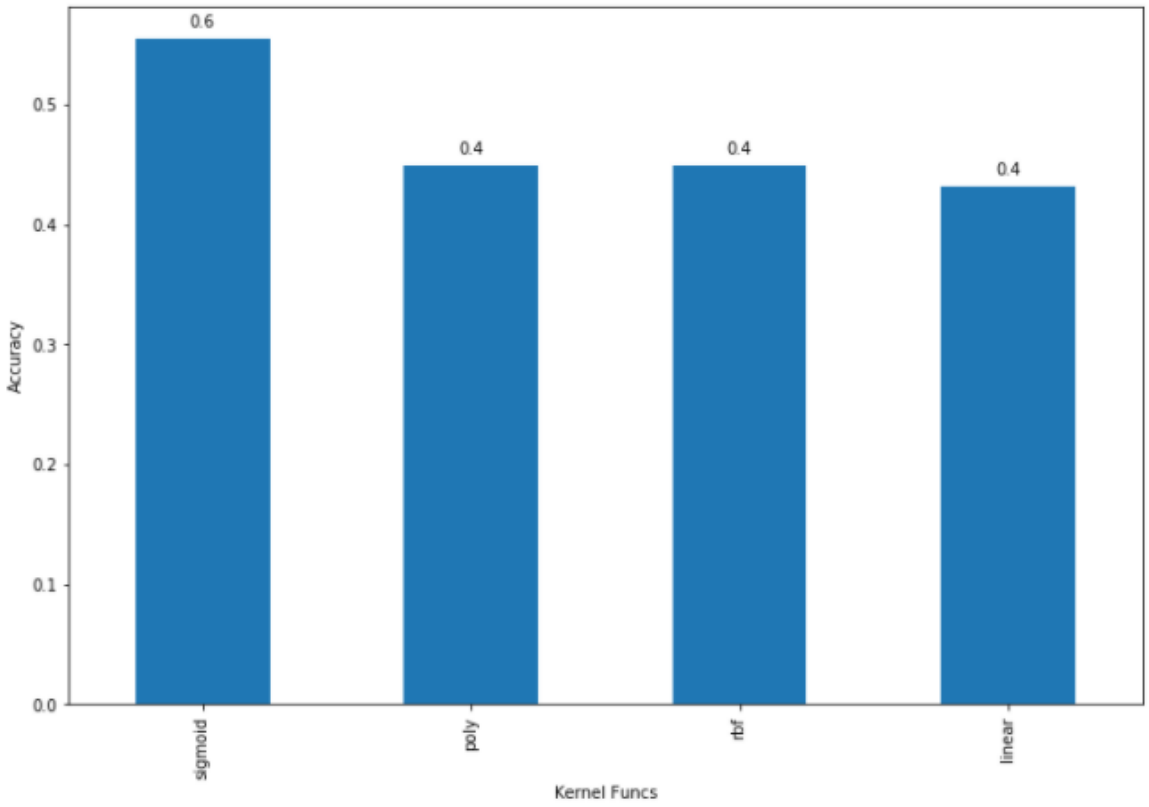


As shown above the accuracy peak at 17 and kept stabilized up to 20. Thus, the K value was settled to 17 to train the model. After train and use the test set to predict we found F1-Score of 54% and Jaccard index of 67%. We will keep this information to later when compare to other models performance.

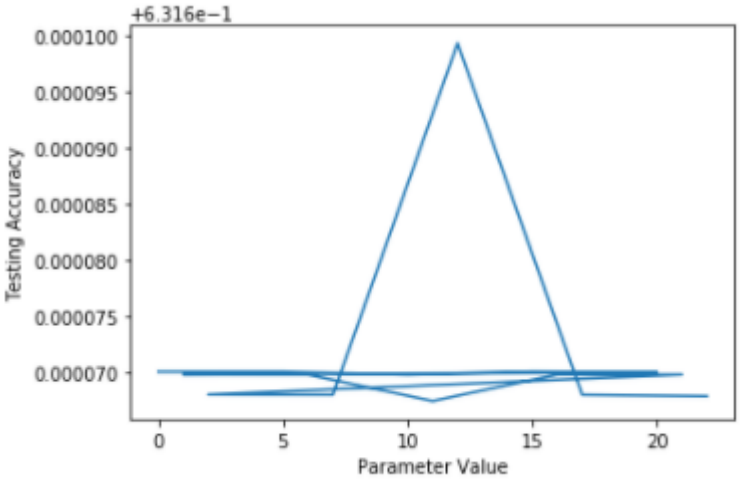
Then we start to work with Decision Tree and create a loop to test up to 10 depth using gini criteria. The best accuracy found during train was with 3 as max depth.

Evaluation Metrices	d = 1	d = 2	d = 3	d = 4	d = 5	d = 6	d = 7	d = 8	d = 9
Jaccard	0.672169	0.672169	0.672169	0.671906	0.671906	0.671848	0.671848	0.671848	0.671848
F1	0.540389	0.540389	0.540389	0.540263	0.540263	0.540235	0.540235	0.540235	0.540235

SVM was the third model select. During the set up we tried different kernel functions and a balanced class weight was used because the dataset has balance issues. The comparison between then shown that the best kernel is sigmoid.



The final model is Logistic Regression. To find the best set up different C parameters and solvers were used. The results shown that the best C parameter is 0.001 and liblinear solver.




```
Test 1: Accuracy at C = 0.1 when Solver = lbfgs is : 0.6316701029161814
Test 2: Accuracy at C = 0.1 when Solver = saga is : 0.6316701056365223
Test 3: Accuracy at C = 0.1 when Solver = liblinear is : 0.6316698193864787
Test 4: Accuracy at C = 0.1 when Solver = newton-cg is : 0.6316701085223504
Test 5: Accuracy at C = 0.1 when Solver = sag is : 0.6316700476869669


Test 6: Accuracy at C = 0.01 when Solver = lbfgs is : 0.6316698353438042
Test 7: Accuracy at C = 0.01 when Solver = saga is : 0.6316698324075275
Test 8: Accuracy at C = 0.01 when Solver = liblinear is : 0.631667487325514
Test 9: Accuracy at C = 0.01 when Solver = newton-cg is : 0.6316698406415793
Test 10: Accuracy at C = 0.01 when Solver = sag is : 0.6316698378985031


Test 11: Accuracy at C = 0.001 when Solver = lbfgs is : 0.6316680645081749
Test 12: Accuracy at C = 0.001 when Solver = saga is : 0.6316680429372415
Test 13: Accuracy at C = 0.001 when Solver = liblinear is : 0.6316993648102899
Test 14: Accuracy at C = 0.001 when Solver = newton-cg is : 0.6316680360800628
Test 15: Accuracy at C = 0.001 when Solver = sag is : 0.6316679156259604
```

Now we gather all the accuracy metrics from all different models to analyze which model perform better. The table below show the metrics to each of them

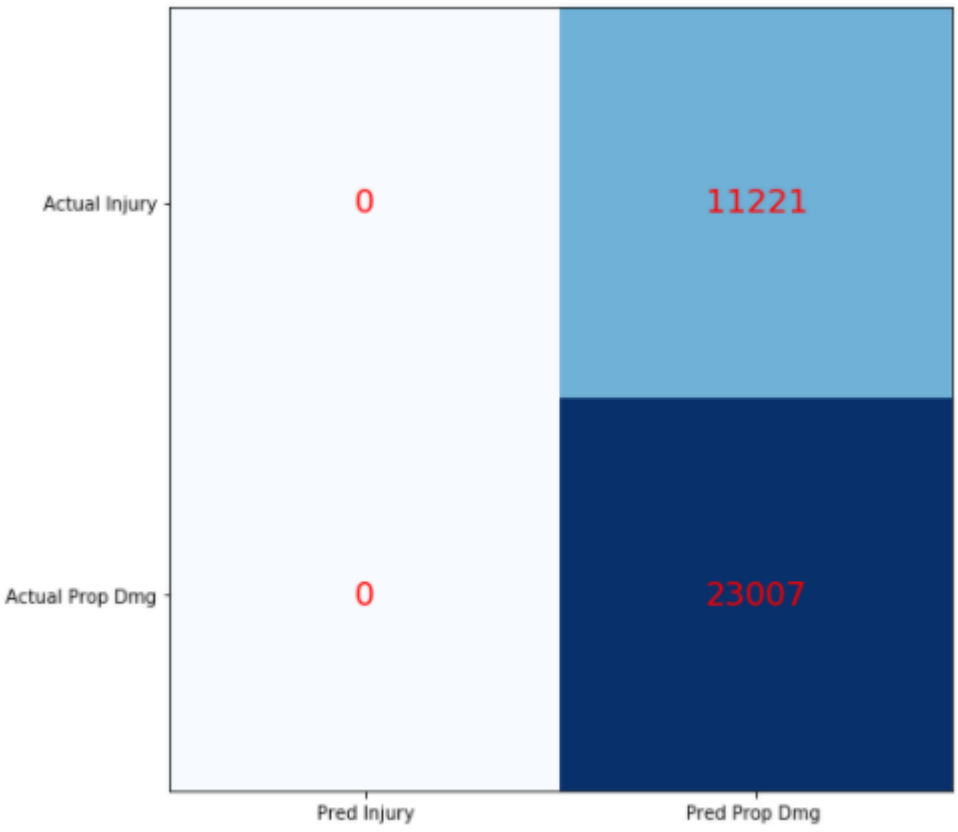
Algorithm	Jaccard	F1-Score	Logloss
KNN	0.67	0.54	NA
Decision Tree	0.67	0.54	NA
SVM	0.55	0.55	NA
Logistic Regression	0.67	0.54	0.63

Besides the technical tie between KNN, LR and Decision Tree the best model to this case is LR because KNN is a model that is not human readable and our goal is to have

something that humans can read and understand what they should do to avoid sever accidents. In the Decision Tree model, the dataset does not have enough information to create a tree that actually bring value to the project.

4. Results

The results were plotted in a confusion matrix to check the precision of the model. The image bellow show the following results



The image bellow shows the final report of the predicted results. Looking to the plot and the report we figured out that 100% of Injury were classified as Prop Damage besides the train and test data had Injury cases.

	precision	recall	f1-score	support
injury	0.00	0.00	0.00	11221
prop damage	0.67	1.00	0.80	23007
micro avg	0.67	0.67	0.67	34228
macro avg	0.34	0.50	0.40	34228
weighted avg	0.45	0.67	0.54	34228

5. Discussion

After checking all the training and test data no issues regard to issue of 0% precision related to Injury results. Both datasets have labels to injury and prop damage. To increase the precision the model needs more data. First because the current dataset only has 2 out of 5 labels. Second because other features like whether the person behind the wheels was on drugs or alcohol and whether the person was on the phone (texting or making calls). Some of these new features are on the dataset but the amount of missing values is big enough to interfere on the results.

6. Conclusion

Therefore, we conclude that the finished model is not optimal enough to delivery a good result that can provide information to the people and government agencies. To improve the model and implement an optimal solution we need more data with all labels instead 2 out of 5. We also need more features or at least more precise data to the ones that we already have.