Car accident severity prediction for Coursera Capstone project

Tereza Stefkova

August 2020

1 Introduction

This project aims at better understanding of causes that lead to severe car accidents. Being able to predict under which conditions severe accidents are more likely to happen, we could adopt measures such as strict speed limits in certain areas to reduce number of severe accidents.

We use data from Seattle in which car accidents are divided into two classes 'injury collision' and 'property damage collision', the former one viewed as serious car accident, the latter considered not severe accident. Using these data, we try to build a model that could predict severity of a car accident based on accident's details such as date and time, number of people involved, location, weather etc..

In this project we will mainly use classification algorithms to build a model that could predict to which of the two above mentioned classes accident belongs. Such a model could potentially be useful for officers who could estimate severity of car accident based on information from emergency calls before actually examining the accident in person.

2 Data

In this project we use dataset provided by Coursera ,metadata describing this dataset are available at https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Metadata.pdf. This dataset consists of 194673 car accident cases each assigned 38 attributes. The entries were collected from 2004 till today (2020).

Data in this dataset are suitable for classification algorithms since each entry (car accident) is labeled by "accident severity". We will use the category of "severity" that corresponds to column "SEVERITYCODE" as a target value in the modelling. Accidents in this dataset are classified as "not severe" (SEVERITYCODE taking value 1) when the accident resulted in property damage only, and "severe" (SEVERITY CODE equal to 2) for accidents with injured persons. As we are primarily interested in severe cases identification, we label "severe" as 1 and "not severe" for the sake of f1 score, to correctly detect true positives. Severe incidents represent 30% of all entries in the dataset, we see a clear bias towards not severe accidents in our data.

2.1 Data cleaning

At the cleaning stage of data wrangling we solve two problems of the dataset: redundant data and missing values.

First, we focus on dropping redundant and not useful data

1. We drop data corresponding to accidents identifiers (OBJECTID, INCKEY, COLDETKEY, REPORTNO, STATUS, INTKEY, EXCEPTRSNCODE, EXCEPTRSNDESD, SEVERITYCODE.1, SEVERITYDESC, SEGLANEKEY, CROSSWALKKEY, SDOTCOLNUM, STCOLCODE, STCOLDESC, SDOTCOLCODE, SDOTCOLDESC)

2. By checking for missing values we find out that the following columns lack more than 80% of data: INATTENTIONIND, PEDROWNOTGRNT, SPEEDING.

3. The data in column LOCATION do not provide the exact address of the accident. We drop this column as well, as we decided to generate the formatted address in further steps.

Secondly, we deal with missing data. To do so, we take a closer look at individual columns to find the best data to replace with. Columns with missing values are ADDRTYPE, COLLISIONTYPE, JUNCTIONTYPE, UNDERINFL, WEATHER, ROADCOND and LIGHTCOND.

For attribute ADDRTYPE we see that there is a significantly prevalent value "Block" by which we replace the missing values in this column. In the case of COLLISIONTYPE, JUNCTIONTYPE, WEATHER, ROADCOND and LIGHTCOND we do not observe any value that would be significantly more frequent than other values. As a result, we categorize these missing values as Unknown.

In the case of the feature UNDERINFL we observe two different values for "Yes, was under influence" - labeled either by 1 or "Y" and "No, was not under influence" characterized by "N" or 0. We unify the categorical labels and at the same time, we replace missing values by 0, since the majority of people involved in the accident were not under influence.

2.2 Feature selection

After cleaning, we have 14 candidates to be potentially included in the feature dataset. In the following, we examine each of the features and decide if there is some dependence between this feature and target data, namely car accident severity.

To this end, we group data corresponding to the given feature by "severity" and observe where there exists difference between severe/not severe ratio for

ADDRTYPE	severity	percentage of cases
Alley	1	89%
Alley	2	11%
Block	1	76%
Block	2	24%
Intersection	1	57%
Intersection	2	43%

Table 1: Feature ADDRTYPE grouped by severity. This table also shows the ratio of severe/not severe accidents for each type of ADDRTYPE.

	severity	percentage of cases
Weekday	1	69%
Weekday	2	31%
Weekend	1	71%
Weekend	2	29%
Summer	1	69%
Summer	2	31%
Winter	1	71%
Winter	2	29%

Table 2: Weekday/weekend, summer/winter grouped by severity.

different categorical values. Let us consider feature ADDRTYPE (Table.1). In this case we see that the percentage of severe/not severe cases differs for each ADDRTYPE, which indicates that this feature might be a good candidate for the feature set. We follow this procedure for columns COLLISIONTYPE, JUNCTIONTYPE, WEATHER, ROADCOND and LIGHTCOND as well.

Contrary to previous features, columns PERSONCOUNT, PEDCOUNT, PEDCYLCOUNT, VEHCOUNT contain numerical data. We would like to transform these data to categorical data to leverage classification process. In order to do so, we cut the values into bins with similar behaviour with respect to car accident severity. For example, in the case of PERSONCOUNT we see that the percentage of severe cases changes rapidly between values 1 and 2 (in other words, if two or more people were involved in the accident, it is more likely to have severe results). This observation enable us to create a new characteristics of the data, namely PERSONCOUNT-BINNED which only takes two categorical values.

Finally, we examine date information about the accident. One of possible questions with regard to the date of the accident is whether there tend to be more severe accidents on weekends/in winter due to different road conditions. We extracted these information from INCDATE column (see Table. 2). However, the data show no significant difference and we discard the information about date altogether.

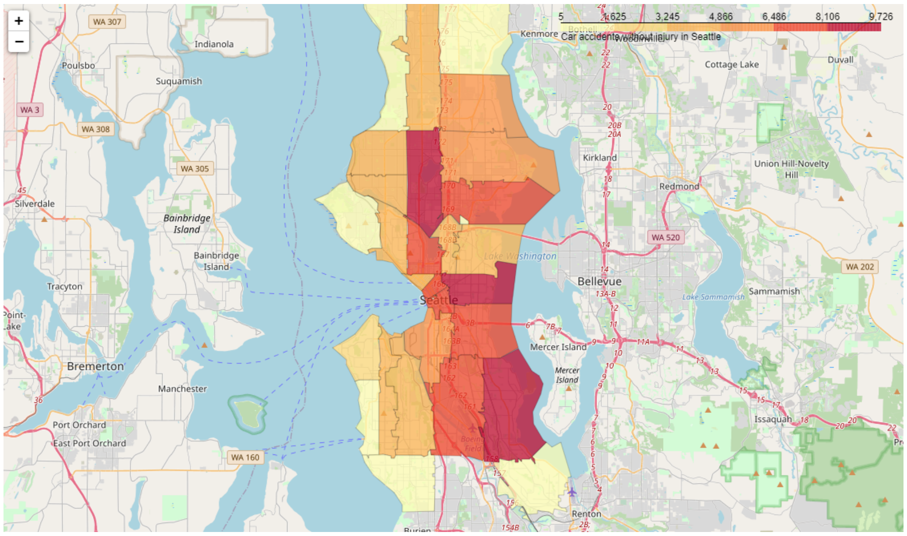

Figure 1: Accidents without injuries in Seattle

2.3 Feature set and data visualization

In the previous section we selected 12 features for our feature set. Since classification algorithms expect numerical inputs only, we create dummy variables for every feature. The resulting feature sets then consists of columns with entries 0 and 1 only. The number of columns arised to 61.

Before we proceed to data modelling stage, let us visualize the data. Our aim is to create choropleth maps that would show number of accidents with respect to area defined by zip codes. First, we have to extract zip code from the data. Since column LOCATION did not include this information, we will use Google GeoData API. This API returns results including formatted address based on the coordinate input which is part of our dataset. We split the number of accident in each area into two groups defined by severity of the accident. The maps are shown in Fig 1 (for incidents without injuies) and Fig 2 (for severe accidents). We see that both distributions of cases are very similar in both cases, however, for some areas we can observe differences. As a result, it might be useful to consider zip code area as a potential feature for our analysis.

To add the information about zip code to the feature set we define three groups of zip codes according to their severity score (group one corresponds to severity=1 percentage between 68 and 72%, the other two groups show percentage less than 68% and more than 72%). Then we create dummy variable for every group and add this information to the feature set. The final feature set has 64 columns.

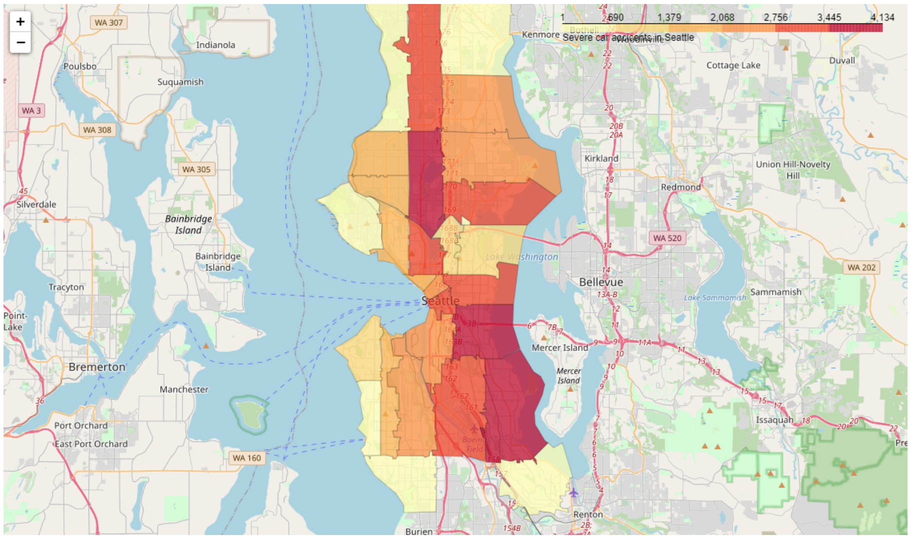

Figure 2: Accident leading to injuries in Seattle

3 Modelling

The structure of the data as well as the definition of our task lead to classification problem. To tackle this problem, we employ two classification algorithms, namely decision tree and logistic regression. These two are robust enough to perform in reasonable time on a relatively big dataset.

We used GridSearchCV from Sci-kit learn library that has a built-in train-test split as well as cross-validation. Moreover, this algorithm finds the best parameters from the pre-specified list of parameters we want consider together with the mean scores for these parameters. In the analysis, we consider four scoring methods:

- acurracy score
- f1 score
- recall score
- precision score

The refit function is set to recall, since we want to optimize the number of true positive severe cases.

4 Results

4.1 Decision Tree

The best parameters found by grid search for the decision tree are as follows

accuracy	f1 score	recall	precision
0.76	0.41	0.28	0.75

Table 3: Evaluation of the decision tree classification model

accuracy	f1 score	recall	precision
0.76	0.44	0.32	0.71

Table 4: Evaluation of the logistic regression classification model

- criterion: 'entropy',
- max-depth: 8,
- splitter: 'best'.

Information about model evaluation are summarized in Table 3

4.2 Logistic Regression

The best parameters found by grid search for the logistic regression are as follows

- C: 1,
- solver: 'newton-cg'. .

Evaluation scores are displayed in Table 4.

5 Discussion

Decision tree model did not perform very well in the task of severe cases identification. The most important characteristics is the recall score, since it determines how accurately the model detects severe cases. In the case of the decision tree model, only 28% of severe accidents were identified correctly. The accuracy score is relatively high, however, this can be attributed to the bias toward not severe cases in our target data. However, this model performs better than simple random classification.

The results for logistic regression show better performance in comparison with the decision tree model, since recall score is slightly higher than in the previous case. However, this model was able to accurately identify severe incidents in only 32%.

6 Conclusions

In this project we selected features for analysis in available data and trained two classification models to tackle the problem of car accident severity prediction. Both models did not perform very well with respect to severe accidents predictability, which was our main goal. This deficiency could be solved using better training data. The data available did not contain features that could set a clear boundary between severe and not severe accidents.

In the future, we would like to improve our model using data with stronger correlation with accident severity. It might also be good to use more powerful computational units such as servers since we work with big data which and were restrained by personal computer limits.