Analysis of Seattle Collision Data

Miranda Childs

August 24th, 2020

1. Introduction and Business Understanding

1.1 Background

In Seattle, WA, from 2004 to present there have been 194,673 collisions reported by the Seattle Police Department (SPD) to the Seattle Department of Transportation (SDOT). 58,188 of those collisions involved an injury.

1.2 Problem

Through analysis of the collision data, we will discover which factors contribute most heavily to severe collisions. By determining and then focusing on those features, we will be able to create meaningful strategies to reduce the number of accidents, especially those with injuries, in order to increase the wellbeing and longevity of our community.

1.3 Interest

We will prepare a presentation for SDOT and the Vision Zero Network, "a collaborative campaign helping communities reach their goals of Vision Zero -- eliminating all traffic fatalities and severe injuries -- while increasing safe, healthy, equitable mobility for all." (Vision Zero Network). Through our thorough analysis we will make recommendations for the next campaigns and strategies that Vision Zero can execute in collaboration with SDOT.

2. Data Understanding and Preparation

2.1 Acquisition of Data

We will be using the shared data set on collisions from 2004 to present, provided by the Traffic Records Group in conjunction with the Seattle Police Department and Seattle Department of Transportation. (Here are links to the data set and corresponding metadata.)

2.2 Data Preparation

We will be preparing the data using the following methods:

Balancing the labeled data: as we can see the labeled data is imbalanced, with 194,673 type 1

```
In [31]: df['SEVERITYCODE'].value_counts()

Out[31]: 1    136485
         2     58188
         Name: SEVERITYCODE, dtype: int64
```

entries ("property damage only"), and 58,188 type 2 entries ("injury"). We must balance the data so that we can use machine learning algorithms most effectively. As we have a fairly large dataset, we will achieve this by undersampling, i.e. removing type 1 entries.

Removing and replacing missing data: exploring the data, we have many null values which must be remedied.

```
In [219]: #And let's also take a look at the null values in each column
          df.isnull().sum()

Out[219]: SEVERITYCODE           0
          X                   5334
          Y                   5334
          OBJECTID               0
          INCKEY                 0
          COLDETKEY              0
          REPORTNO               0
          STATUS                 0
          ADDRTYPE            1926
          INTKEY            129603
          LOCATION            2677
          EXCEPTRSNCODE     109862
          EXCEPTRSNDESC     189035
          COLLISIONTYPE       4904
          PERSONCOUNT            0
          PEDCOUNT               0
          PEDCYLCOUNT            0
          VEHCOUNT               0
          INCDATE                0
          INCDTTM                0
          JUNCTIONTYPE        6329
          SDOT_COLCODE           0
          SDOT_COLDESC           0
          INATTENTIONIND    164868
          UNDERINFL           4884
          WEATHER             5081
          ROADCOND            5012
          LIGHTCOND           5170
          PEDROWNOTGRNT     190006
          SDOTCOLNUM         79737
          SPEEDING          185340
          ST_COLCODE            18
          ST_COLDESC          4904
          SEGLANEKEY             0
          CROSSWALKKEY           0
          HITPARKEDCAR           0
          dtype: int64
```

```
In [32]: df['WEATHER'].unique()

Out[32]: array(['Overcast', 'Raining', 'Clear', nan, 'Unknown', 'Other', 'Snowing',
                'Fog/Smog/Smoke', 'Sleet/Hail/Freezing Rain', 'Blowing Sand/Dirt',
                'Severe Crosswind', 'Partly Cloudy'], dtype=object)
```

Furthermore, attributes such as weather have both NaN values as well as the values 'Unknown' and 'Other'.

Transformation: for example, the attribute UNDERINFL is type object but can easily be transformed to integers, converting N to 0 and Y to 1.

```
In [38]: print(df['UNDERINFL'].dtype)
         print(df['UNDERINFL'].unique())

         object
         ['N' '0' nan '1' 'Y']
```

Cleaning the dataset: we will drop unnecessary columns, especially those with mostly null values, and rename others for ease of reference.

2.3 Feature Selection

We will examine attributes in our dataset such as driver inattention (INATTENTIONIND), whether a driver was under the influence (UNDERINFL), whether a driver granted a pedestrian right-of-way (PEDROWNOTGRNT), and whether they were speeding (SPEEDING), as well as the weather, road, and light conditions (WEATHER, ROADCOND, LIGHTCOND). Analysis of this data will inform us of the most relevant topics to focus on for the next Vision Zero campaign.

3. Methodology

section which represents the main component of the report where you discuss and describe any exploratory data analysis that you did, any inferential statistical testing that you performed, if any, and what machine learnings were used and why.

(Discussing our modeling) In this phase, various algorithms and methods can be selected and applied to build the model including supervised machine learning techniques. You can select SVM, XGBoost, decision tree, or any other techniques. You can select a single or multiple machine learning models for the same data mining problem. At this phase, stepping back to the data preparation phase is often required.

4. Results

discuss the results. “Certain metrics can be used for the model evaluation such as accuracy, recall, F1-score, precision, and others” — or should this be in discussion section???

5. Discussion

- section where you discuss any observations you noted and any recommendations you can make based on the results.

“Before proceeding to the deployment stage, the model needs to be evaluated thoroughly to ensure that the business or the applications' objectives are achieved. Certain metrics can be used for the model evaluation such as accuracy, recall, F1-score, precision, and others.”

6. Conclusion

The deployment phase requirements vary from project to project. It can be as simple as creating a report, developing interactive visualization, or making the machine learning model available in the production environment. In this environment, the customers or end-users can utilize the model in different ways such as API, website, or so on.