Predicting Severity of Collision

Goh Yok Khim

Oct 6, 2020

Introduction

This is a case study which is to predict the severity of a traffic collision in Seattle City. Many of us might have experienced terrible traffic jam due to the unpredictable road accidents. In order to improve travel experience of road users, Seattle Department of Transportation (SDOT) aims to create an application that can help to predict severity of a road accident based on historical datasets. With this, people can drive more carefully or even change their travel route if possible so.

Data Understanding

For this case study, I use historical dataset that contains collections of collision records provided by Seattle Police Department (SPD). This includes all types of collisions that happened from 2004 to present. The label for the data set is severity, which describes the fatality of an accident. The dataset consists of 194,674 records. Each record represents a road collision took place in Seattle City. The data collection has only two category of severity which is 1 - (Property Damage Only Collision) and 2 (Injury Collision). This means the model we are building will be trained and tested to classify the two class of accident severity. The dataset contains 38 features including a dependent variable “SEVERITYCODE”. Some of the records is categorized as exception due to not having sufficient information, such records only accounted 1% of the data and may not be useful for prediction. For the same reason, approximately 2% of records flagged as “Not Enough Information / Not Applicable” by SDOT should be disregarded in order to produce a good model. On the other hand, there are 8 categorical variables such as “JUNCTIONTYPE”, “INATTENDTIONIND”, “UNDERINFL”, “WEATHER”, “ROADCOND”, “LIGHTCOND” that some of them have missing data or data being not standardized and should be handled accordingly. Some of features may be important to the model training section.

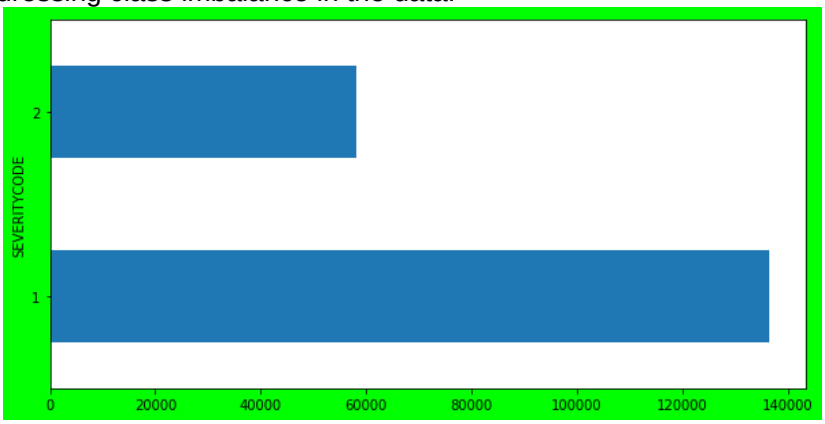
```
In [8]: df.columns[df.isna().any()].tolist()

Out[8]: ['X',
         'Y',
         'ADDRTYPE',
         'INTKEY',
         'LOCATION',
         'EXCEPTRSNCODE',
         'EXCEPTRSNDESC',
         'COLLISIONTYPE',
         'JUNCTIONTYPE',
         'INATTENTIONIND',
         'UNDERINFL',
         'WEATHER',
         'ROADCOND',
         'LIGHTCOND',
         'PEDROWNOTGRNT',
         'SDOTCOLNUM',
         'SPEEDING',
         'ST_COLCODE',
         'ST_COLDESC']
```

Picture 1: Illustrates columns with NaN values

Below table illustrates data definition of categorical variables that contain missing values:

Attribute	Description
ADDRTYPE	Collision address type: • Alley • Block • Intersection
COLLISIONTYPE	Collision type
SDOT_COLDESC	A description of the collision corresponding to the collision code.
INATTENTIONIND	Whether or not collision was due to inattention. (Y/N)
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.
JUNCTIONTYPE	Category of junction at which collision took place

As many other datasets do, this is a class imbalance problem where the number of data points belonging to the minority class, in this case “Severity 2 - Injury Collision” is only approximately 30% of entire data collection which is far smaller than the number of data points belonging to the majority class “Severity 1 - Property Damage Only Collision”. There is need to perform sampling-based methods for addressing class imbalance in the data.



Picture 2: Illustrates imbalance data distribution on Severity

Methodology

Exploratory Data Analysis

To get started with exploratory data analysis for this case study, first is to perform sampling-based methods for addressing class imbalance in the data. In this case, the dataset is resampled by oversampling minority class which means randomly replicates the examples from minority class “2 Injury Collision”. Prior to that, the categorical variables need to be converted to numeric data for correlational analysis and model training/testing. Some categorical features are having blank category and being categorized as “Unknown” as the affected data is very small. When both classes are having balanced dataset, I can now perform detailed exploratory data analysis for some features against the dependent variable.

```
Out[6]: SDOT_COLCODE          int64
        EXCEPTRSNCODE        object
        INCKEY                int64
        INATTENTIONIND         int8
        UNDERINFL             int64
        ADDRTYPE           category
        EXCEPTRSNDESC        object
        SDOT_COLDESC         object
        SEVERITYCODE          int64
        WEATHER            category
        SPEEDING           category
        ROADCOND           category
        LIGHTCOND          category
        JUNCTIONTYPE       category
        PERSONCOUNT           int64
        VEHCOUNT              int64
        ADDRTYPE_CAT           int8
        WEATHER_CAT            int8
        SPEEDING_CAT           int8
        ROADCOND_CAT           int8
        LIGHTCOND_CAT          int8
        JUNCTIONTYPE_CAT       int8
        dtype: object
```

Picture 3: Illustrates categorical variable converted from text to category type

WEATHER_CAT	0	1	2	3	4	5	6	7	8	9	10	11
WEATHER												
BLANK	4167	0	0	0	0	0	0	0	0	0	0	0
Blowing Sand/Dirt	0	46	0	0	0	0	0	0	0	0	0	0
Clear	0	0	108071	0	0	0	0	0	0	0	0	0
Fog/Smog/Smoke	0	0	0	549	0	0	0	0	0	0	0	0
Other	0	0	0	0	698	0	0	0	0	0	0	0
Overcast	0	0	0	0	0	27007	0	0	0	0	0	0
Partly Cloudy	0	0	0	0	0	0	5	0	0	0	0	0
Raining	0	0	0	0	0	0	0	32479	0	0	0	0
Severe Crosswind	0	0	0	0	0	0	0	0	25	0	0	0
Sleet/Hail/Freezing Rain	0	0	0	0	0	0	0	0	0	110	0	0
Snowing	0	0	0	0	0	0	0	0	0	0	867	0
Unknown	0	0	0	0	0	0	0	0	0	0	0	9781

JUNCTIONTYPE_CAT	0	1	2	3	4	5	6	7
JUNCTIONTYPE								
At Intersection (but not related to intersection)	2032	0	0	0	0	0	0	0
At Intersection (intersection related)	0	61944	0	0	0	0	0	0
BLANK	0	0	172	0	0	0	0	0
Driveway Junction	0	0	0	10613	0	0	0	0
Mid-Block (but intersection related)	0	0	0	0	22713	0	0	0
Mid-Block (not related to intersection)	0	0	0	0	0	86169	0	0
Ramp Junction	0	0	0	0	0	0	159	0
Unknown	0	0	0	0	0	0	0	3

Table 2: Illustrates categorical features with NaN values

To study the correlations of each feature versus target variable, I used Cross Tab function and plot horizontal bars to visualize. From the plots, here are some of observations made:

A. WEATHER: “Unknown” and “Other” weather usually caused more SEV 1 than SEV 2 collisions but the information may not be useful for the study at all due to the weather description is ambiguous. However, looking at the data distribution as overall weather somehow has impact to classify severity of a collision.

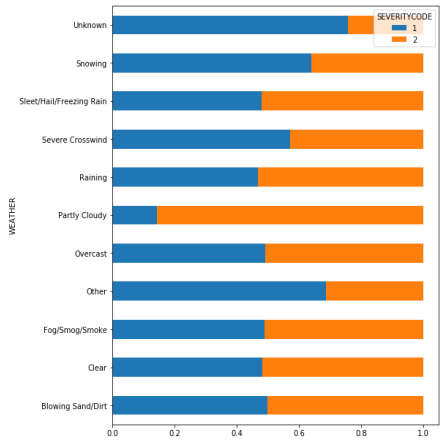


Figure 1: WEATHER VS SEVERITYCODE DISTRIBUTION

B. ADDRTYPE: Most of SEV 2 accidents happened in intersection while most of SEV 1 accidents happened in alley. This feature will play a vital role in classification predicting.

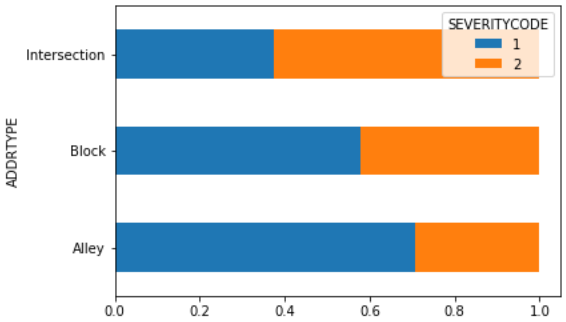


Figure 2: ADDRTYPE VS SEVERITYCODE DISTRIBUTION

C. LIGHTCOND: Likewise, the "Unknown" and "Other" category distribution is high on SEV 1. However, the data distribution of other groups is slightly significant hence this variable may somehow useful for model predicting.

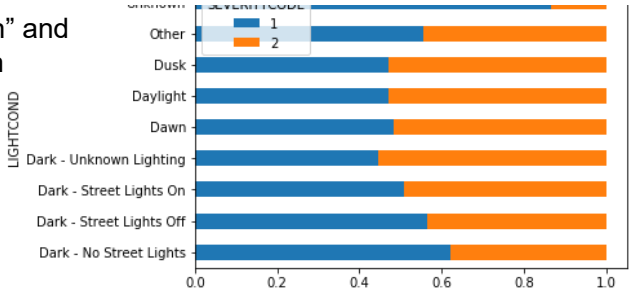


Figure 3:LIGHTCOND VS SEVERITYCODE DISTRIBUTION

D. ROADCOND: Again, the "Unknown" and "Other" road condition distribution is high on SEV 1. However, the data distribution of some groups is slightly significant hence this variable may somehow useful for model predicting.

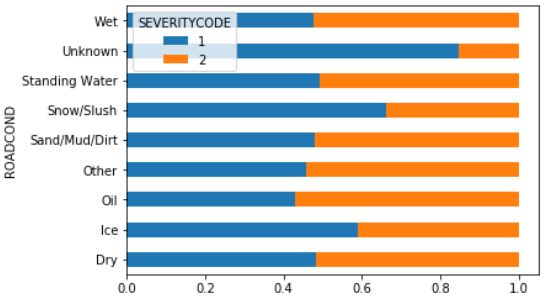


Figure 4: ROADCOND VS SEVERITYCODE DISTRIBUTION

E. INNATTENTIONIND (Inattention Indicator): There were more SEV 2 collision caused by inattentive drivers compared to SEV 1. This shows that the feature will have some impacts to predicting.

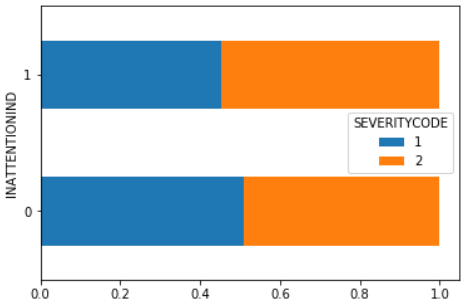


Figure 5: INATTENTIONIND VS SEVERITYCODE DISTRIBUTION

F. UNDERINFL: There were more collision caused by drivers whom consumed drug/ alcohol are SEV 2 compared to SEV 1. This shows that the feature will have some impacts to predicting.

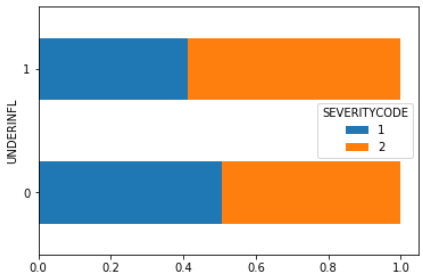


Figure 6: UNDERINFL VS SEVERITYCODE DISTRIBUTION

G. SPEEDING: More collisions caused by speeding are SEV2. This is an important feature for model predicting.

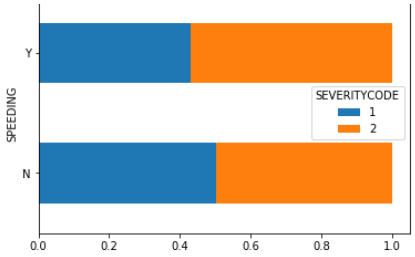


Figure 7: SPEEDING VS SEVERITYCODE DISTRIBUTION

H. JUNCTIONTYPE/PERSONCOUNT/VEHCOUNT: The data distribution point of the features is quite signification among the two classes. This shows that the feature will have impacts to predicting.

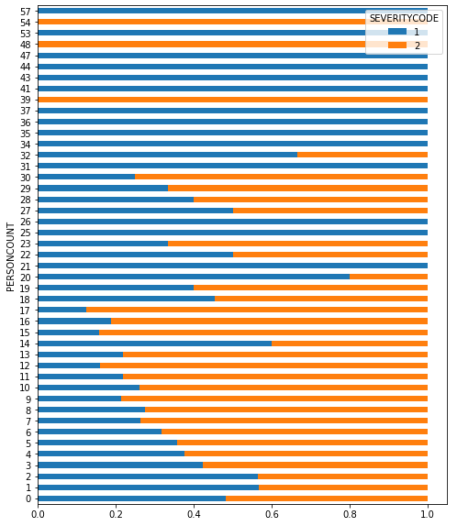


Figure 8: PERSONCOUNT VS SEVERITYCODE DISTRIBUTION

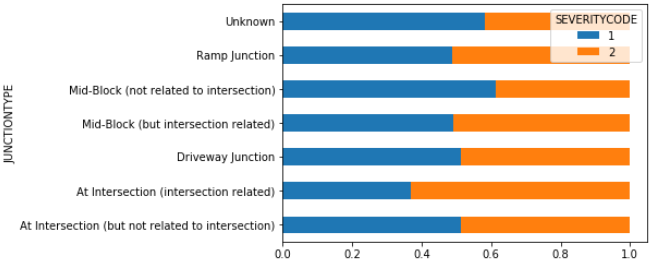


Figure 9: JUNCTIONTYPE VS SEVERITYCODE DISTRIBUTION

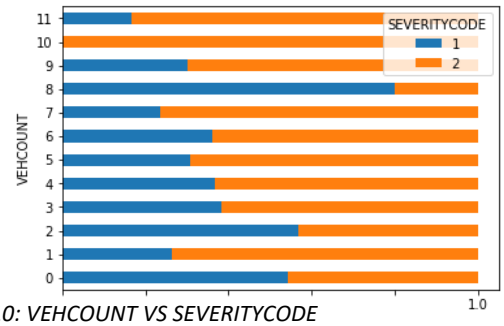


Figure 10: VEHCOUNT VS SEVERITYCODE DISTRIBUTION

Predictive Modelling

From above exploratory analysis, I concluded to use below features and drop other irrelevant variables to conduct model predicting.

- INATTENTIONIN
- UNDERINFL
- ADDRTYPE
- WEATHER
- SPEEDING
- ROADCOND
- LIGHTCOND
- JUNCTIONTYPE
- PERSONCOUNT
- VEHCOUNT

There are several types of algorithm to train a classification model. In this case study, I experimented using Logistic Regression (LR), Decision Tree Classifier (DT) and Random Forest (RF) Classifier at the same time manipulating the imbalanced dataset using different re-sampling models such as Scikit Learn oversampling, undersampling and Imblearn oversampling and undersampling. As a result, DT with Imblearn sampling technique produces the best number in term of testing accuracy, precision, F1-score and recall.

Result

In this section, the result of model training with different types of algorithms is discussed. When training model using imbalanced dataset, the average accuracy and precision of LR, DT and RC are closed to 0.7, which is relatively higher than using balanced dataset generated using Scikit learn sampling technique. However, the variance of measured metric is noticeable when DT and RC model is used to train using dataset that is under-sampling with the same technique. Overall, training DT model with Imblearn Random Under/Over Sampling technique gives best result of all.

Algorithm	Accuracy	Precision	F1 Score	Recall
Logistic Regression (Imbalanced Dataset)	0.703	0.673	0.644	0.703
Random Forest Classifier (Imbalanced Dataset)	0.732	0.72	0.692	0.732
Decision Tree Classifier (Imbalanced Dataset)	0.740	0.739	0.694	0.740
Logistic Regression (Up-sampling)	0.627	0.627	0.627	0.627
Random Forest Classifier (Up-sampling)	0.673	0.674	0.673	0.673
Decision Tree Classifier (Up-sampling)	0.660	0.662	0.659	0.660
Logistic Regression (Down-sampling)	0.615	0.622	0.608	0.615
Random Forest Classifier (Down -sampling)	0.485	0.474	0.422	0.485
Decision Tree Classifier (Down -sampling)	0.476	0.460	0.420	0.476
Logistic Regression (Resampling using Imblearn Random Under Sampling)	0.704	0.674	0.651	0.704
Random Forest Classifier (Resampling using Imblearn Random Under Sampling)	0.647	0.700	0.660	0.647
Decision Tree Classifier (Resampling using Imblearn Random Under Sampling)	0.740	0.729	0.704	0.740
Decision Tree Classifier (Resampling using Imblearn Random Over Sampling)	0.740	0.732	0.702	0.740

Table 3: Performance Metric of Model Training using different sampling techniques

Discussion

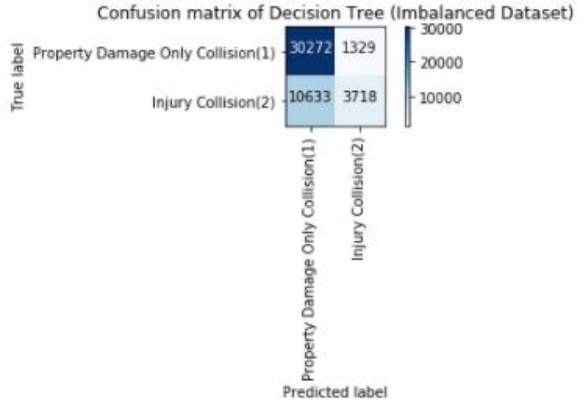
Class Imbalance is a common problem in machine learning, especially in classification problems. Imbalance data can affect our model accuracy in a long run. Although the performance metrics of imbalanced dataset is relatively good but it may result poor predictive performance when working on real world data.

A widely adopted technique for dealing with highly unbalanced datasets is resampling. It consists of removing samples from the majority class (under-sampling) and/or adding more examples from the minority class (over-sampling). Despite the advantage of balancing classes, these techniques also have their weaknesses. Undersampling can be defined as removing some observations of the majority class. This is exactly why the performance metrics of all algorithm are comparatively low. Undersampling can be a good choice when we have a ton of data -think millions of rows. But a drawback to undersampling is that we are removing information that may be valuable. Oversampling can be defined as adding more copies to the minority class. Oversampling can be a good choice when you don't have a ton of data to work with. A con to consider when undersampling is that it can cause overfitting and poor generalization to your test set. This explains the reason of getting lower performance metric for model training with resampled dataset.

Random under-sampling with Imblearn is a fast and easy way to balance the data by randomly selecting a subset of data for the targeted classes. Under-sample the majority class(es) by randomly picking samples with or without replacement. One way to fight imbalance data is to generate new samples in the minority classes. The naivest strategy is to generate new samples by randomly sampling with replacement of the currently available samples. The Imblearn Random Over Sampler offers such a scheme. This technique is proven to produce better performance metric with DT model.

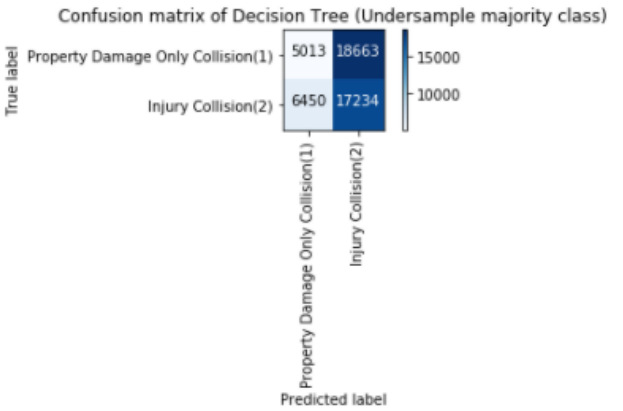
	precision	recall	f1-score	support
1	0.74	0.96	0.84	31601
2	0.74	0.26	0.38	14351
accuracy			0.74	45952
macro avg	0.74	0.61	0.61	45952
weighted avg	0.74	0.74	0.69	45952

```
Confusion matrix, without normalization
[[30272  1329]
 [10633  3718]]
```



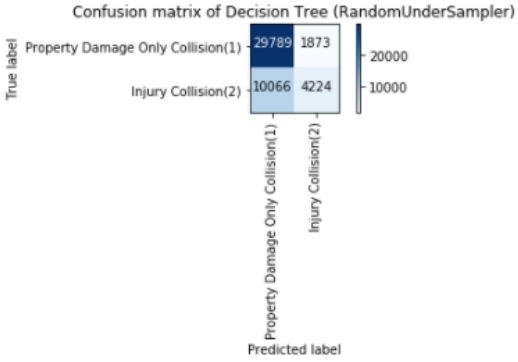
	precision	recall	f1-score	support
1	0.44	0.21	0.29	23676
2	0.48	0.73	0.58	23684
accuracy			0.47	47360
macro avg	0.46	0.47	0.43	47360
weighted avg	0.46	0.47	0.43	47360

```
Confusion matrix, without normalization
[[ 5013 18663]
 [ 6450 17234]]
```



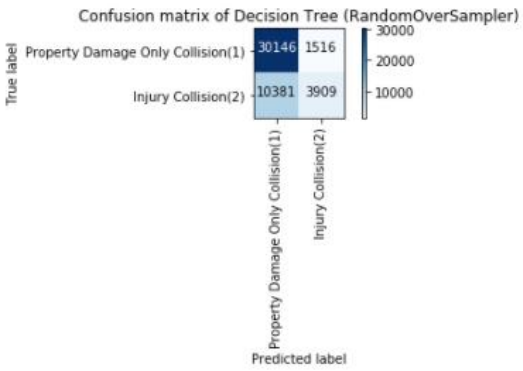
	precision	recall	f1-score	support
1	0.75	0.94	0.83	31662
2	0.69	0.30	0.41	14290
accuracy			0.74	45952
macro avg	0.72	0.62	0.62	45952
weighted avg	0.73	0.74	0.70	45952

```
Confusion matrix, without normalization
[[29789  1873]
 [10066  4224]]
```



	precision	recall	f1-score	support
1	0.74	0.95	0.84	31662
2	0.72	0.27	0.40	14290
accuracy			0.74	45952
macro avg	0.73	0.61	0.62	45952
weighted avg	0.74	0.74	0.70	45952

```
Confusion matrix, without normalization
[[30146  1516]
 [10381  3909]]
```



Conclusion

In this study, I learned that classification of collision severity is closely dependent on multiple key features like weather condition, road condition, driver behaviour, speed and etc. There are some efforts needed to refine data collection of collisions. For example, some of categorical variables that have high data distribution as “Unknown” and “Other” should be refined to collate more meaningful information. Undoubtedly, the performance metric can be further fine-tuned along the way with some extends of data refinement and model training error correction.

Reference:

https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/