**Seattle Car** **Accident Severity**

1. **Introduction**

Car accidents or on road collisions and something we witness daily on the news. The vehicle count on road today is much larger than it used to be 10 years ago.

The predictive analysis performed here aims towards analyzing the “Severity” of the accident/collision based on road conditions, lighting conditions, area of collision, number of people involved and many more factors as such. Knowing the severity of any such collision beforehand will lead to prevention and prompt action.

1. **Data**

All the collision data used in this analysis is taken from ArcGIS, which was provided by Seattle Police Department and recorded by traffic records. The data provided is that of collisions which took place in the city of Seattle, from year 2004 till present.

Mentioned below is list of features that was available in the raw data:

|  |  |
| --- | --- |
| **SEVERITYCODE** | Target Column (1: Property Damage Only Collision, 2: Injury Collision) |
| **SEVERITYCODE.1** | Copy of Target Column |
| **SEVERITYDESC** | Description of Target Column |
| SDOTCOLNUM | A number given to the collision by SDOT |
| JUNCTIONTYPE | Category of junction at which collision took place |
| X, Y | Coordinate of accident |
| LIGHTCOND | The light conditions during the collision |
| WEATHER | A description of the weather conditions during the time of the collision |
| ROADCOND | The condition of the road during the collision |
| ST\_COLCODE | A code provided by the state that describes the collision |
| ST\_COLDESC | A description that corresponds to the state's coding designation |
| COLLISIONTYPE | Collision type |
| UNDERINFL | Whether or not a driver involved was under the influence of drugs or alcohol |
| LOCATION | Description of the general location of the collision/address |
| ADDRTYPE | Collision address type (Alley, Block, Intersection) |
| SDOT\_COLCODE | A code given to the collision by SDOT |
| SDOT\_COLDESC | A description of the collision corresponding to the collision code |
| SEGLANEKEY | A key for the lane segment in which the collision occurred |
| CROSSWALKKEY | A key for the crosswalk at which the collision occurred |
| VEHCOUNT | The number of vehicles involved in the collision |
| INCDTTM | The date and time of the incident |
| INCDATE | The date of the incident |
| PEDCYLCOUNT | The number of bicycles involved in the collision |
| PEDCOUNT | The number of pedestrians involved in the collision |
| PERSONCOUNT | The total number of people involved in the collision |
| STATUS | (Matched, Unmatched) \*\*\* |
| REPORTNO | Report identifier |
| COLDETKEY | Secondary Key for incident |
| INCKEY | Unique key for incident |
| OBJECTID | ESRI unique identifier |
| HITPARKEDCAR | Whether or not the collision involved hitting a parked car |
| EXCEPTRSNCODE | \*\*\* |
| EXCEPTRSNDESC | \*\*\* |
| PEDROWNOTGRNT | Whether or not the pedestrian right of way was not granted |
| SPEEDING | Whether or not speeding was a factor in the collision |
| INATTENTIONIND | Whether or not collision was due to inattention |
| INTKEY | Key that corresponds to the intersection associated with a collision |

\*\*\* Column details not available.

There are in total 38 data columns in the dataset including the 3 target related columns. We will keep various aspects in mind while deciding the importance of a particular column or the transformation it may need before we feed it to the model.

Some of the given data columns are features related to or identifying a single particular accident, thus may not be very much useful for our predictive analysis. These features include:

*SDOTCOLNUM, Coordinates, LOCATION, INCDTTM, INCDATE, REPORTNO, COLDETKEY, INCKEY, OBJECTID.*

There are some description columns for a given code. Columns *ST\_COLDESC, SDOT\_COLDESC and EXCEPTRSNDESC* are description columns for code which is already specified in the given dataset.

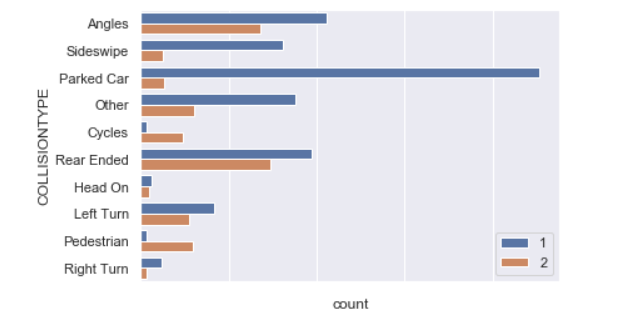
There are also data columns which has missing data in abundance. *Column EXCEPTRSNCODE,* EXCEPTRSNDESC, *PEDROWNOTGRNT, SPEEDING, INATTENTIONIND* and *INTKEY* have more than 50% of data missing. Although few of these columns can be very crucial indicator of collision severity, it would be misguiding to use it with so many missing rows and very difficult to fill in these categorical values.

Columns mentioned in all the three categories above will not be used in the model that we are going to build. Most of the columns that remains are categorical and will require one-hot and label encoding before we can use them as a feature for our model.

1. **Methodology**
   1. **Exploratory Data Analysis**

First part of the process will be to explore the data and understand that how a particular data column is distributed.

Most of our data columns are categorical and we need to know that how affect the severity of the accident. Figure below showing frequency of Property Damage Only Collision and Injury Collision with respect to collision type feature.



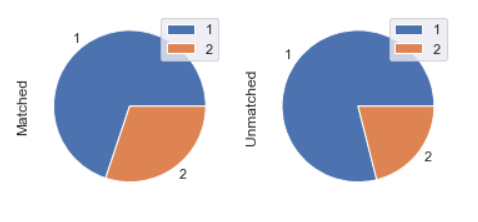


Figure above showing class distribution of ‘Matched’ and ‘Unmatched’ categories of status variable.

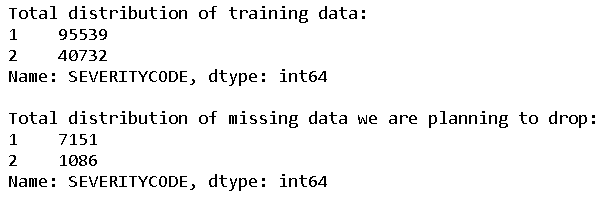
There are low cardinality categorical variables with 6-7 categories, moderate cardinality categorical variables with 40-70 categories and very high cardinality categorical variable with 1500+ categories.

* 1. **Feature Engineering**

Mostly all the variables (except the features which defines the number of people, vehicles etc.) are nominal features; i.e., features where the categories are only labelled without any order of precedence. Preferred encoding for these categories is One-Hot encoding. However, One-Hot encoding will generate around 1500 data columns for just one high cardinality categorical variable, which will be very expensive to work with.

We can get over this hurdle by using feature hashing. Feature hashing is an encoding technique which is used to encode high cardinality feature by hashing them. By this we can pull down the number of encoded data columns to 32-64 even for variables with >1500 categories.

Distribution of all missing data in the training set was found to be:



As the class proportion is not getting much affected by dropping these data rows, we will proceed to do so.

After the process of feature hashing and one-hot encoding, we obtain in total 208 feature columns. We are using Random Forest to get the feature importance, eliminating 40 least important features and correlation matrix to detect >90% correlations.

After removing the least important and highly correlated features we are left with 160 features to train the model with.

* 1. **Modelling**

As it was clear from above analysis that we have had a skewed dataset. This resulted in a low recall on class 2 and as a result low F1 score.

To solve this problem, we used smote to oversample the rare class and generated the cross-validation score again. While doing oversampling we have to keep in mind that oversampling should be done on each iteration of cross-validation and not on the whole training set.

As a result, we observed that although the recall on class 2 and F1 increased a little bit, it decreased the accuracy too. Considering the increase in computational expense due to increased data, oversampling didn’t prove to be worth the effort in this case.

We used the XG Boost Classifier to start with and plotted the learning curve to see if the model is overfitting the training data. We observed that converged training and validation error were close to each other, which means that we can use high variance algorithms like Random Forest, XG Boost and Support Vector Machine, and we can also use the high number of features that we are using.

Below are the cross-validation results for both the algorithms.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm** | **Accuracy** | **Weighted Precession** | **Weighted Recall** | **Weighted F1** |
| Random Forest | 0.74 | 0.73 | 0.74 | 0.72 |
| XG Boost | 0.75 | 0.75 | 0.75 | 0.73 |

As expected, we got the best performance from XG Boost Classifier. We will further try hyperparameter tuning to improve the performance.

1. **Results**

For final prediction we have to preprocess the whole test data-set. While encoding the feature columns we made sure that the one-hot encodings are same as the train set and feature hasher transformer used should be fitted on train data.

Following are the Final result on the test data:

|  |  |
| --- | --- |
| Accuracy | 0.76 |
| Weighted Precession | 0.77 |
| Weighted Recall | 0.76 |
| Weighted F1 | 0.71 |
| Jaccard Score | 0.74 |

1. **Discussion**

Many more analysis and methodologies can be added to this project as a future work. We haven’t used the coordinates. Those coordinates could result in some unforeseen clusters which could exponentially improve the study.

Further other encoding techniques can be used in place of feature hashing or feature hashing with different feature count can be used. The performance of these changes can be evaluated using cross-validation.

1. **Conclusion**

The results are satisfactory but expectations were much higher. A lot of improvement can be done on class 2 predictions. Overall a lot of improvement can be observed from the basic model.