**Seattle Car Accident Severity**

1. **Introduction**
   1. Background

According to World Health Organization, road traffic injuries are a major but neglected public health challenge that requires concerted efforts for effective and sustainable prevention. Of all the systems with which people have to deal every day, road traffic systems are the most complex and the most dangerous. Worldwide, an estimated 1.2 million people are killed in road crashes each year and as many as 50 million are injured. Projections indicate that these figures will increase by about 65% over the next 20 years unless there is new commitment to prevention.

* 1. Problem

Data that describes each car accident can also be used to determine the severity of each accident. This project aims to predict how severe a car accident could be based on address type, collision type, weather, road condition and time. Knowing the severity of any such accident beforehand will lead to prevention and prompt action.

* 1. Interest

Insurance companies would like to know what features influence the severity of car accidents and adjust their insurance premium. Government would plan new road layouts to decrease car accidents during city construction.

1. **Data**
   1. Data Source

The dataset comes from [Seattle GeoData](http://data-seattlecitygis.opendata.arcgis.com/datasets/5b5c745e0f1f48e7a53acec63a0022ab_0/data). But the target variable only contains two classes.

* 1. Data Pre-Processing

The dataset consists of 194673 car accidents. There are 37 input features and 1 target variable. The first column is the target variable, namely SEVERITYCODE. Each accident is labeled as 1(prop damage) or 2(injury). Some features were dropped because they are unrelated(such as the key for each accident) or ex-post(such as the number of people involved in the collision) . After this step, there remains 5 input features. There were a lot of missing values in each feature. I decided to drop entire rows with missing values. After this step, there are still 187706 accidents in the dataset.

|  |  |  |
| --- | --- | --- |
| **Input Feature** | **Data Type** | **Description** |
| ADDRTYPE | Text | Collision address type:   * Alley * Block * Intersection |
| COLLISIONTYPE | Text | Collision type |
| INCDTTM | Text | The data and time of the incident. |
| WEATHER | Text | A description of the weather conditions during the time of the collision. |
| ROADCOND | Text | The condition of the road during the collision. |

Next, we converted categorical features to dummy values. Since each feature has many classes, we had to drop those columns with sparse data to make sure the conciseness of our models. After this step, there are 17 input features in the dataset. Then we normalized data.

|  |  |
| --- | --- |
| **Input Feature** | **New Input Feature** |
| ADDRTYPE | Block |
| Intersection |
| WEATHER | Clean |
| Raining |
| Overcast |
| ROADCOND | Dry |
| Wet |
| COLLISIONTYPE | Parked Car |
| Angles |
| Rear Ended |
| Other |
| Sideswipe |
| Left Turn |
| Pedestrian |
| Cycles |
| Right Turn |
| Head On |

1. **Exploratory Data Analysis**

This part will be to explore the data and understand that how a particular data column is distributed. Most of our input features are categorical and we need to know how they affect the severity of accidents. Figure below showing frequency of collisions at different time with respect to collision type feature. We see that accidents that happened at midnight are less severe, so let’s use Feature binarization to set a threshold values less then hour 3. The new feature is called “MIDNIGHT”.

A picture containing drawing, room

Description automatically generated

1. **Models**

Tree-based ensemble algorithms are frequently used for supervised machine learning problems. Here, three tree-based ensemble algorithms are used to train and test models. They are AdaBoost, random forests and XGBoost.

* 1. AdaBoost

It is sequentially growing decision trees as weak learners and punishing incorrectly predicted samples by assigning a larger weight to them after each round of prediction. This way, the algorithm is learning from previous mistakes. The final prediction is the weighted majority vote (or weighted median in case of regression problems).

* 1. Random Forest

This algorithm is bootstrapping the data by randomly choosing subsamples for each iteration of growing trees. The growing happens in parallel which is a key difference between AdaBoost and random forests. Random forests achieve a reduction in overfitting by combining many weak learners that underfit because they only utilize a subset of all training samples. Another difference between AdaBoost and random forests is that the latter chooses only a random subset of features to be included in each tree, while the former includes all features for all trees.

* 1. XGBoost

XGBoost was developed to increase speed and performance, while introducing regularization parameters to reduce overfitting. Gradient boosted trees use regression trees (or CART) in a sequential learning process as weak learners. These regression trees are similar to decision trees, however, they use a continuous score assigned to each leaf (i.e. the last node once the tree has finished growing) which is summed up and provides the final prediction. For each iteration i which grows a tree t, scores w are calculated which predict a certain outcome y. The learning process aims to minimize the overall score which is composed of the loss function at i-1 and the new tree structure of t. This allows the algorithm to sequentially grow the trees and learn from previous iterations. Gradient descent is then used to compute the optimal values for each leaf and the overall score of tree t. The score is also called the impurity of the predictions of a tree.

1. **Results**

I divided the dataset into train set(80%) and test set(20%). AdaBoost, random forest and XGBoost were built. Following are the final results on the test set:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Weighted avg  precision | Weighted avg  recall | Weighted avg  F1-score |
| AdaBoost | 74.51% | 78% | 75% | 68% |
| Random Forest | 74.48% | 77% | 74% | 68% |
| XGBoost | 74.51% | 78% | 75% | 68% |

The differences between models are very small. Among the models, AdaBoost and XGBoost performed the best and they have similar results.

1. **Discussion**

Many more analysis and methodologies can be added to this project as a future work. We dropped many input features, but those features could result in some unforeseen results which could exponentially improve the study.

Further there are some hyperparameters could be set in the models. We could establish for-loops to find the optimum hyperparameters. Overall a lot of improvement can be observed from the models.

1. **Conclusion**

In the study, I analyzed the relationship between car accident severity and description data. I built AdaBoost, random forest and XGBoost to predict how severe a car accident can be. These models can be very useful in helping insurance companies know what features influence the severity of car accidents and adjust their insurance premium and helping government would plan new road layouts to decrease car accidents during city construction.