Predicting the Severity of an Accident

**Background**

In a country such as the United States (US), transportation in, out of and within a city is very important, especially for the working population. With exorbitant rental rates in the city, many workers may opt to stay in suburbs. Thus, many of these workers would travel at least an hour from their homes to their workplaces in the city. As such, even a minor accident could prolong their travelling time significantly, decreasing productivity. Hence, being able to predict an accident, including its severity, would empower drivers with the option to choose alternative travel routes or perhaps provide an option to drive an alternative time with less traffic.

**Problem**

The project aims to use traffic data to predict the occurrence and severity of an accident at a specific location and help travelers reduce their travel time before they start their journey.

**Application**

This ability to predict the severity of an accident would obviously benefit anyone who travels to-and-fro to the city. If workers are travelling to their workplaces, this would help ensure workers promptly arrive at meetings. In the case of students, it would ensure they arrive on time and not miss any lessons, otherwise there will be a need for make-up lessons or for personal catch up. Overall, it would help reduce wastage of time on the road for any traveler, especially during peak hours, when the traffic is most intense.

**Data Source**

It is obtained directly from the Coursera page, which is directly from the Seattle Department of Transportation (SDOT). The file originally contains 194,673 rows and 38 columns.

**Feature Identification**

At first glance, it is difficult to identify the relevant features for model prediction. However, as the objective is to *predict* the severity of an accident, any post-accident data is irrelevant.

This is despite the clear correlation between the severity of an accident and the post-accident data column, such as the number of injuries, serious injuries, fatalities and vehicle count.

Of course, any column with SDOT codes are also irrelevant. Thus, with much thought of the relevant data, the features used for the machine learning model will include data columns that fulfil two characteristics:

1. It must be information available before an incident happens, enabling prediction.
2. It has correlation with the severity of an incident.

By method of elimination, many columns do not meet the first criteria, such as post-accident data or report-related data. The only relevant categorical variable that could be relevant is the location column. However, since it is a categorical variable and contains numerous entries, it is difficult to encode them all to help predict the severity of an incident. Nonetheless, this data column may be used for further development in the application of this model.

As such, there are only a few features that fulfil the criteria above. Namely, they are the columns on weather, road condition, light condition and address type.

**Data Cleaning**

All features are categorical variables and contain many unique values, making it difficult to replace null values. Thus, I decided to remove rows that contain null values for all four columns. After this operation, the dataset is left with 187,525 rows.

Next, I noticed there are rows with “Unknown” values for three columns (weather, road condition, light condition). I decided to delete these rows due to these variables being categorical ones. Each column contains a variety of values, making it difficult to replace the unknown values. After removal of unknown values, 169,781 rows remain, which is still a considerable amount of data. Furthermore, I saw that the same three columns contained “Other” values, which is difficult to replace. Thus, I removed these rows from the dataset, leaving a total of 169,247 rows. Instead of replacing unknown values with possibly erroneous data, and since there is a considerable amount of data left, I think removal of these rows is justifiable.

**Feature Encoding**

Using the python function to one hot encode each column, the four features are further split into twenty-six columns. Three of the columns are from the address type column, nine columns from weather, seven columns from light condition and seven columns from road condition. Column names are not modified as there is no overlap.

**Modelling**

From the entire dataset, although the severity code column has five possible values, the current dataset only contains two values. It is either a one or two, meaning the incident is a prop injury or an injury.

For the splitting of data, I decided to split the data into three parts: training, testing and evaluation. Sixty percent is used to train the model, twenty percent to test the model and tune the parameters for the model. Lastly, the remaining twenty percent is used as an unbiased source for model evaluation. There was no need to normalize the data as all feature columns are binary values. Thus, I proceeded straight to modelling the data.

As covered, I tested the four models: K nearest neighbors, decision tree, support vector machines and logistic regression. The first model I tried is the K nearest neighbor model, which gave an optimum K of six, with an accuracy of sixty-six percent. This seems like a rather low value of accuracy.

Next, the decision tree model was adopted, giving an optimum depth of five, with an accuracy of sixty-seven percent. It is slightly better than the previous model, but still a sub-optimal rate of accuracy, prior even to model evaluation. However, upon trial with support vector machine models and logistic regression models, both types of models also returned approximately a sixty-seven percent accuracy score.

From initial thought, it seems like a clear case to select logistic regression as the appropriate model for this scenario, as there are only two possible outcomes and no in-betweens. However, all models have a very similar accuracy score and the optimal model will likely have a minimal edge over the others.

**Model Evaluation**

As mentioned earlier, twenty percent of the data has been reserved for this step. For all models, I used the optimal parameters and calculated the various f1 scores, jaccard scores and log loss scores (applicable for Logistic Regression).

There was not a sharp difference between the scores of all models, as clearly seen from the modelling stage. The poorest performer is still the K nearest neighbor model. It is followed by the Decision Tree model and Support Vector Machine model, with minimal advantage. As postulated, the Logistic Regression model, with a liblinear kernel, has the best scores. It has an f1 score of over fifty percent, jaccard score of sixty-seven percent and log loss score of sixty-two percent.

Overall, the choice for the logistic regression model is a sub-optimal choice l for predicting the severity of an incident, as the scores do not cross the seventy percent mark.

**Improvement**

The model has many potential areas of improvement. Firstly, I think predicting the severity of an incident is a step ahead of its relevance for travelers. The first step for this model should be to predict the probability of an incident at a specific location, given the same features (address, weather, road condition and light condition). In this way, travelers are better able to plan their routes better, given such information. This would be the first layer of prediction, followed by the prediction of an incident, given that it occurs. For this first layer to be modelled, the same data needs to be collected on all days, not just on days when there were incidents. In this way, the probability of an incident, given such conditions, can be calculated.