Accident Prediction Models for Illinois Highways: Progress Report 2

Jacob Mathew (jmathew7)

Trisha Das (trishad2)

Jinsong Cui (jinsong4)

## Update since Progress Report 1

In the first progress report, we presented our methodology for data cleaning to obtain a meaningful dataset, results of our exploratory data analysis, and preliminary results using three different models. We also listed out some of the challenges that we faced and this report shows our methodology that we adopted to overcome these.

Feature Selection

Among the features available in the data set, we selected the following features to build our model. These features were also modified slightly as shown in Table 1.

Table 1: Features used in analysis

| **Feature Name** | **Raw Data** | **Modified Data** | **Comments** |
| --- | --- | --- | --- |
| no\_lanes | A continuous positive integer representing number of lanes at a location | None |  |
| lane\_wid | A continuous positive integer representing the width of lanes in feet | None |  |
| curve\_rad | Indicates the radius of curvature of the road | Converted into a binary variable (was there a curve or not) | Curve radius of 0 indicates straight segment |
| seg\_len | A continuous positive integer representing the length of the segment under consideration | None |  |
| access | Three categories representing access control at the location | None | 1. Uncontrolled 2. Partial control 3. Full control |
| oneway | Binary variable indicating one way or not | None | 1. One way 2. Two way |
| med\_type | Categorical variable with 8 categories indicating type of median | Converted into two categories | 1. Median present 2. Median absent |
| spd\_limt | Continuous variable indicating posted speed limit at location | Missing values in this variable was imputed (details in the next section) |  |
| surf\_typ | Categorical variable with multiple categories indicating type of surface of road | Converted into three categories | 1. Flexible 2. Rigid 3. Others |
| rururb | Binary variable indicating whether location is a rural or urban area | None |  |
| rodwycls | Categorical variable indicating the class of roadway | None |  |

## Estimating Missing Values of Speed Limit ( feature ‘spd\_limt’):

Speed limit is an important feature for accident prediction. However, there were many missing values in this column in the data. We estimated the missing speed limit values by using a multiclass classification. We used a random forest model to train and validate only on the rows which have known speed limits. We separated the rows with unknown speed limits and predicted the missing values using our model. Our model for predicting speed limits has an accuracy of 0.87 on the validation data. After filling out the missing values, we used this new dataset for estimating likelihood of accidents.

## Accident Count Categorization

Accident count at a location is either 0 or a positive integer. We divided this into four categories for the purpose of building the model. The categories are

1. No accident (Accident count = 0)
2. Low number of accidents (1 <= Accident Count <= 30)
3. Medium number of accidents (31 <= Accident Count <= 50)
4. High number of accidents (Accident Count > 50)

## Current Model for Estimating Likelihood of Accident

The following models were developed on the filtered and merged data. A total of 1000 records were held out for testing.

### XGBoost

We used the XGBoost model to estimate no, low, medium and high accidents. We standardized the numerical features before building the model. We used the XGBClassifier from xgboost which is capable of doing multiclass classification. We used 1000 estimators with max depth 5 to build the model. We are still tuning the hyperparameters. Also, we will need to upsample or downsample the imbalance dataset.

### Random Forest

We use the random forest model to estimate the likelihood of accidents. There are two important features in our dataset. First is the imbalanced character in accident likelihood. Under most conditions, no accident will be observed. In order to get rid of the biased data and biased model, we need to upsampling or downsampling the data to make the model more practical.

### Logistic regression

A multi class logistic regression was performed to classify the roadway records as likely to have no, low, medium and high accidents. Before performing the logistic regression, the aadt feature was scaled using the log scale. Scaling the aadt gives a more spread distribution of this feature. Logistic regression was performed using the sklearn library. Since this is a multiclass problem, the One Vs Rest algorithm was implemented.

## Evaluation of the Current Model

### XGBoost

With current selection of features we are getting 68.80% accuracy on test data.

### Random Forest

Currently, with the selection of features and up/down sampling of data, the accuracy of training can reach nearly 100% compared to the 35% accident rate in our gathered dataset.

### Logistic regression

The logistic regression model performed fairly well on the test dataset with a classification accuracy of 75.5%. However, it failed to predict locations with accidents in the category “medium” and “high”. This needs to be further analyzed. A potential solution to this might be upsampling of data before training the model.

Table 2: Results of model evaluation

| **Model** | **Accuracy Score** |
| --- | --- |
| XGBoost | 68.80% |
| Random Forest | 100% (on upsampled data) |
| Logistic Regression | 75.5% |

## Current Challenges

The accuracy rates of XGBoost and Logistic Regression models could be improved

The random forest model has a higher than normal accuracy rate which could result from the manually upsampled data and many features in the dataset.

Training upsampled data with random forest model and CPU can take a long time. For each iteration, the training time can go up to 2hrs with AMD 5600X and 64g of RAM. It will be feasible to build a model with GPU.

## Plans for Upcoming Weeks

1. Hyperparameter tuning to improve performance of each of the models.
2. Explore the use of deep learning engines to solve this problem.