Module 5 Assignment: Crash Detection

Dhairyav Jatin Shah

College of Professional Studies, Northeastern University

Professor: Justin Grosz

February 12, 2023

**Introduction**

The data in this dataset pertains to recent complaints from a large number of bikers in the City of Austin. The city must install safety steps to protect cyclists from motor vehicles, according to the concerns. The city then gathered a dataset connected to the most recent accidents that have occurred in order to comprehend and take appropriate action for safety. "Crash Severity" is the variable we are aiming for. The city has requested the creation of a model to determine the severity of the collisions and, depending on those aspects, what policy and regulatory modifications should be suggested for implementation by the city.

**Data cleaning**

In order to begin data cleansing and investigation, we first begin by looking at the data set's columns. Over 80% of the records in the columns **"Average Daily Traffic Amount"** and **"Highway System"** contain null values. Therefore, we would exclude these columns from subsequent analyses. A categorical variable will be created from the Boolean value of the column **"At Intersection Flag".** In the columns labeled **"Surface Condition"** and **"Weather Condition",** we swap out **"Other (Explain in Narrative)**" for **"Unknown"** to make categorization easier. In the column **“Speed Limit”** we observe that there are **343 records** where the speed limit was set at -1 which is not possible as speed is a quantity which cannot contain negative measures. Hence, we drop these 343 rows as well.

**EDA**

Once we have completed the data cleaning, we first plot a histogram for all numerical variables to understand the distribution. From fig no 1(appendix 1) we can see that column “Crash Time” is skewed towards the right side where we can understand that majority of the accidents have happened in the evenings between 16:00 and 20:00. The columns “Crash Total Injury Count” shows the number to people injured in the incident. Where we can see that over 1750 incidents only 1 person was injured in the incident which would probably be the cyclist. Whereas there have been incidents where more than 2 people have been injured in the incident. Over 350 incidents were reported in the year of 2011 followed by over 300 incidents in the year 2012. And we have seen a decline in the number of incidents reported after 2013. Over 600 incidents have been reported when the speed limit on the road was 35 km. Such roads are often the roads connecting cities with State Highways or Interstate Highways. To understand and simplify categorical variables using a for loop we create new columns where we give labels to every category of the categorical variables. We can see the formed dictionaries for reference in fig no 2 (appendix 1). For our target variable, we divide it into 2 categories i.e., **Possibly Injury, Incapacitating Injury, Killed (Given 1) vs. Non-Incapacitating Injury, Not Injured (Given 0).** Before we start modelling the data set we plot a heatmap to understand the correlation between the dependent and independent variables. From fig no 3 (appendix 1), the 6 selected variables for modelling are :

1. **'Crash Total Injury Count'(0.14)**
2. **"$1000 Damage to Any One Person's Property\_new"(0.07)**
3. **'Surface Condition\_new'(0.07)**
4. **'Crash Year'(0.05)**
5. **'Intersection Related\_new'(-0.05)**
6. **'Light Condition\_new'(-0.04)**

These variables are selected based on the correlation value they have with the dependent variable.

**Analysis**

**Modeling**

We start first by building a Logistic Regression Model using the 6 variables selected earlier we fit the logistic regression model. From table no 1.1 (from appendix 2) we can see that 4 features are not significant to the model hence we would be dropping these features and refit the model with remaining variables. The accuracy of the first model is 63% and looking at the confusion matrix in table no 1.5 (appendix 2) the model was able to predict 214 true positives and 52 true negatives, 64 false negatives and 94 false positives. After refitting the model, the 2 features used in the model are “Crash Total Injury Count” and “Crash Year”. The accuracy of the model increased to 67% and confusion matrix in table no 1.5 (appendix 2) the model was able to predict 276 true positives and 9 true negatives, 2 false negatives and 137 false positives. After fitting logistic regression model we try to fit random forest classification model. Upon prediction we can see the accuracy of the model is only 63% and looking at confusion matrix we can see that the model was able to predict 232 true positives and 36 true negatives, 46 false negatives and 110 false positives. From fig no 4 (appendix 1), we can see the feature importance and for Random Forest classification the model has Crash Year as the most significant variable followed by light condition ,intersection, crash total injury count and surface condition. For the third model we fir a neural network classification model. To prepare this model we use the MLP classifier, with 10 neurons in the hidden layer, the activation function od the hidden layer is the **rectified linear unit function** which returns **f(x) = max(0,x)**, the solver for weigh optimization is set to **stochastic gradient descent,** The **learning\_rate\_init is set to 0.01**. It controls the step-size in updating the weights and the last parameter is the max\_iter which is the maximum iterations. Since we have used the solver as SGD, max\_inter determines the number of epochs (number of times each data point would be used to predict. 1st we take max\_iter = 15000, for which we obtain an accuracy of 67% and the following confusion matrix 277 true positives and 9 true negatives, 1 false negatives and 137 false positives. 2nd we take max\_iter = 20000, for which we obtain an accuracy of 67% and the following confusion matrix 273 true positives and 10 true negatives, 5 false negatives and 136 false positives. Comparing the 2 models we can see that this model was not able to predict as many true positives as the earlier model, but it predicts more false negatives then the earlier model. Hence, we would increase the max\_iter to 25000, for which we obtain an accuracy of 68% and the following confusion matrix 274 true positives and 15 true negatives, 4 false negatives and 131 false positives. In the 3rd model we can see that the neural network was able to predict better true negatives and less false positives and negatives compared to the earlier models. Even though this neural network is not that accurate, but it gives the most accurate predictions out of all the earlier trained models including Logistic Regression and Random Forest Classification.

**Conclusion**

Comparing all 3 models Logistic Regression, Random Forest Classification and Neural Network Classification, we can see that Neural Network Classification and Logistic Regression are performing very similarly and when we compare the confusion matrix we can see that the logistic regression was able to predict the true positives with a little better accuracy but was not able to predict true negatives when compared to the neural network classification model. The accuracy of the models is also very close. The target variable that we have is also very biased towards the non- sever cases hence it is able to predict the non- sever cases better than the sever cases. One major challenge we are facing in the neural network classification is that there is a possibility of overfitting. My recommendation would be the city should use the neural network classification. As it is classifying better true positives and negatives compared to the other models.

**Reference**

1. *sklearn.neural\_network.MLPClassifier* <https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html>
2. Raj.A (Oct 2021) *An Exhaustive Guide to Decision Tree Classification in Python 3.x* <https://towardsdatascience.com/an-exhaustive-guide-to-classification-using-decision-trees-8d472e77223f>
3. N.d. (May 2018) *Understanding Random Forests Classifiers in Python Tutorial* <https://www.datacamp.com/tutorial/random-forests-classifier-python>
4. Chouinard.J (May 2022) *How to use Confusion Matrix in Scikit-Learn (with Example)* <https://www.jcchouinard.com/confusion-matrix-in-scikit-learn/>
5. Chouinard.J (May 2022) *How to use Classification Report in Scikit-learn (Python)* <https://datatofish.com/statsmodels-linear-regression/>

**Appendix 1**

**Chart, histogram

Description automatically generatedFig no 1: Distribution Plot**

**Text, letter

Description automatically generatedFig no 2: Data Dictionary**

**Fig no 3: Heat Map**

**A picture containing chart

Description automatically generated**

**Fig No 4: Feature Importance of Random Forest Classification**

**Chart, bar chart

Description automatically generated**

**Appendix 2**

**Table 1: Logistic Regression Model**

**Table

Description automatically generatedTable 1.1 : Logistic Regression Model 1 Results**

**Table 1.2 : Logistic Regression Model 2 Results Table

Description automatically generated**

**Table 1.3 : Classification Report of Logistic Regression Model 1**

Table

Description automatically generated

**Table 1.4 : Classification Report of Logistic Regression Model 2**

Table

Description automatically generated

**Table 1.5 : Confusion Matrix of Logistic Regression Models**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model 1** | | | **Model 2** | | |
|  | **Yes** | **No** |  | **Yes** | **No** |
| **Yes** | 214 | 64 | **Yes** | **276** | **2** |
| **No** | 94 | 52 | **No** | **137** | **9** |

**Table 2 : Random Forest**

**Table 2.1 : Confusion Matrix for Random Forest.**

|  |  |  |
| --- | --- | --- |
|  | **Yes** | **No** |
| **Yes** | 232 | 46 |
| **No** | 110 | 36 |

**Table 2.2 : Classification report of Random Forest.**

Table

Description automatically generated

**Table 3 : Neural Network Classification**

**Table 3.1 : Confusion Matrix for Neural Network Classification.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Iter 1** | | | **Iter 2** | | | **Iter 3** | | |
|  | **Yes** | **No** |  | **Yes** | **No** |  | **Yes** | **No** |
| **Yes** | 277 | 1 | **Yes** | 273 | 5 | **Yes** | **274** | **4** |
| **No** | 137 | 9 | **No** | 136 | 10 | **No** | **131** | **15** |

**Table 3.2 : Classification report of Neural Network Classification.**

**Iteration 1:**

**Table

Description automatically generated**

**Iteration 2:**

Table

Description automatically generated

**Iteration 3:**

**Table

Description automatically generated**