**Introduction**

The goal of this project is to help the city of Austin access the data they have collected from bicycle accidents after they got complaints that they aren’t doing enough to prevent them from motor vehicles. We help them identify what they can do to prevent this and what factors are contributing to these accidents. We also suggest which model they should use to predict the severity of these accidents.

**Data Cleaning**

Data cleaning is one of the most important steps in the data analysis process and to get accurate results. For the purposes of this project, we have been given data of cyclists related incidents collected by the Austin city. We can observe that for the column Average daily traffic amount, 85.7% of the rows have the value no data. So, we remove this column. We do not have missing values in any of the columns. "$1000 Damage to Any One Person's Property", "Active School Zone Flag" and "Construction Zone Flag" have yes or no values which we change to 0 or 1. We also convert 'At Intersection Flag' from Boolean to integer. We have 10 object columns. We apply one hot encoding on these columns since the models we are using require the independent values to be numerical. We also remove the column ‘Crash Total Injury Count' since the number of injuries cannot help us come with any action points.

**Data Loading**

Initially we load the Austin city accident data using scikit learn which is a machine learning library. We then manually divide the data into training and testing sets. We use the training dataset to train the model and then we use the same model on our testing dataset so we can compare the results from training data and testing data to check the accuracy of the model.

**Logistic Regression model**

Chart, scatter chart

Description automatically generated It is a supervised machine learning algorithm. It is used to predict the probability of the dependant/ target variable. It works by using a Sigmoid function to map the output probabilities. We put 0 and 1 on x and y axis and the graph looks as follows.

In our example, we are considering crash severity as our dependant variable. We assign zero to 'Non-Incapacitating Injury','Not Injured' and 1 to 'Possible Injury','Incapacitating Injury','Killed'. We have 80 independent variables after we one hot encoded 10 columns. To chose appropriate features and those that are of significance, we use forward selection.

**Forward Selection**

In the forward selection method, we use a model/method that looks at all the predictor variables you selected and picks the one that predicts the most on the dependent measure. That variable is added to the model. This is repeated with the variable that then predicts the most on the dependent measure. This little procedure continues until adding predictors does not add anything to the prediction model anymore.

We use **SelectKBest** for the feature selection and **chi2** for the score we use to select these features.

1. SelectKBest: The SelectKBest class just scores the features using a function (in this case chi2 but could be others) and then "removes all but the k highest scoring features"
2. Chi2: The chi2 function from the sklearn feature selection package returns the chi-square statistic and the p-value.

We can give the k value as 15 to determine 15 most significant features.

**SMOTE or the Synthetic Minority Over-sampling Technique.**

We can observe that most of the data is classified as 0 in the crash severity column. To overcome this imbalance in the data, we use a method of over sampling i.e you add copies of instances from the under-represented class. We use the algorithm SMOTE. It works by creating synthetic samples from the minor class instead of creating copies. The algorithm selects two or more similar instances (using a distance measure) and perturbing an instance one attribute at a time by a random amount within the difference to the neighboring instances. [1]

|  |  |  |  |
| --- | --- | --- | --- |
| **Predicted** | | | |
| **Actual** |  | **Positive** | **Negative** |
| **Positive** | 561 | 753 |
| **Negative** | 357 | 957 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Predicted** | | | |
| **Actual** |  | **Positive** | **Negative** |
| **Positive** | 1306 | 8 |
| **Negative** | 639 | 17 |

**Results**

The first table depicts the results after applying SMOTE algorithm and the second table depicts the results when we apply logistic regression on the data directly. Refer apppendix A result 2 to see the summary that depicts the probability of each feature. We can observe the p and coeeficient values and see that accidents are prone to be not severe when intersection has driveway access or they don’t occur in the intersection at all and when the cyclist is wearing helmet. We can observe in appendix A result 1 that when we don’t apply SMOTE algorithm, we can also see that the number of false negatives are more i.e., the model is predicting as not severe but they were severe accidents and the results are not providing useful insights.

Diagram

Description automatically generated**Random Forest Algorithm**

It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. We can give the maximum depth of the tree and how many trees the algorithm needs to create. We have checked for 5000, and for 10000 the confusion matrix for them are below. We can see that they have the same values and they also have the same accuracies. We have also checked the most important features as seen in Appendix A graph 1. We can see that most of the samples are being classified as incapacitated even when the accident is not very severe.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Predicted** | | | | | | |
| **Actual** |  | **Non-Incapacitating Injury** | **Possible**  **Injury** | **Incapacitating Injury** | **Not Injured** | **Killed** |
| **Non-Incapacitating Injury** | 3 | 0 | 54 | 0 | 0 |
| **Possible Injury** | 0 | 0 | 2 | 0 | 0 |
| **Incapacitating Injury** | 0 | 0 | 284 | 0 | 0 |
| **Not Injured** | 0 | 0 | 38 | 0 | 0 |
| **Killed** | 0 | 0 | 112 | 0 | 0 |

**Neural Network**

A [neural network](https://www.sciencedirect.com/topics/computer-science/neural-networks) model is represented by its architecture that shows how to transform two or more inputs into an output. The transformation is given in the form of a learning algorithm. In this work, the feed-forward architecture used is a [multilayer perceptron](https://www.sciencedirect.com/topics/computer-science/multilayer-perceptron) (MLP) that utilizes [back propagation](https://www.sciencedirect.com/topics/computer-science/backpropagation) as the learning technique. The input layer in our example are the independent variables and the output layer is the severity of the accident. The model modifies the weight and bias in every layer until it can predict the category. We can use multiple types of activation functions to deal with whether the network continues to the next node. The most Diagram

Description automatically generatedcommon and the default one is Relu. When we have negative values, we use tanh as this model can classify between -1 to +1 whereas relu and sigmoid can only classify in the positives. Sigmoid tries to fit the results between 0 to 1 while relu can fit it between any positive number we decide to assign as the range. In our problem, we have used neural networks model with various activation methods and we see they yield the same results and they have a 61% accuracy and all the values are being predicted as non-incapacitating injury which would be dangerous when the accident is severe.

In neural networks, it is also not possible to understand which feature has had the most impact on the results which is not suitable for our problem as we are tasked with providing recommendations to the Austin city.

**Comparison between the modules**

We now pick various benchmarks that are important to us to create a table for comparison, we are comparing the false negatives i.e. where we are saying it is not a severe accident but is actually severe, accuracy of the mode, significant/important features we got, precision i.e, ability of a classifier not to label positive to the negatives and speed in seconds since the epoch.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | False negatives | Accuracy  (%) | Significant/Important  Features | Precision for false |
| Logistic Regression  (Over sampling) | 357 | 50.7 | Intersection Related\_Driveway Access and not intersection, helmet-damaged and not damaged | 0.61 |
| Logistic Regression | 639 | 66.9 | Roadway-service,frontage road, no traffic control, Intersection Related\_Driveway Access | 0.67 |
| Random forest n=5000 & n=10000 | Kindly refer the confusion matrix as the classification is not binary | 58.2 | Crash time,speed limit, intersection related-non intersection,helmet worn-not damaged, | NA |
| Neural Networks | NA | 61.6 | NA | The precision for NonIncapacitating Injury  Is 0.61 and all other values are 0 |

**Conclusion**

From the analysis in the code attached and the above results we can conclude the following:

1. Neural networks model is not the best model as we can’t give recommendations for the city and provide insights on the features.
2. Random forest model has less accuracy and when seen in the above random forest section, we see most of the accidents are being predicted as incapacitating and based on this classification, recommendations cannot be provided.
3. We can recommend the city use logistic model as we can draw more insights from this model and when the data was balanced using over sampling, we get the least number of false negatives.
4. We can see those cyclists wearing helmets are in less severe accidents so the city can make strict rules about always wearing helmets.
5. We can also observe that when the accidents are not at intersections, they are less severe. The city can invest in signals and stop signs at the intersections to avoid accidents in these areas. The city can also make sure there is quick and easy access from intersections to the hospitals as a precautionary measure.

**References**

1. **8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset** by Jason Brownlee dated August 19th 2015 on machinelearningmastery.com

**Appendix A**

***Result 1:***

***Table

Description automatically generated***

***Result 2:***

***Table

Description automatically generated***

***Graph 1:***

***Chart

Description automatically generated***