Team What Is This Behavior?

AT THE INTERSECTION OF DATA: PREDICTING VEHICLE CRASH SEVERITY USING MACHINE LEARNING

Ankit Agrawal
aa6eu

Supriya Savaram
ss6nnd

Nikita Sivakumar
ns4dg

April 26, 2020

ABSTRACT

This project developed an algorithm that predicts crash severity in Northern Virginia based on a variety of crash features in the "Crash Data" dataset from the Virginia Department of Roads. The following study identified the most prominent features impacting crash severity, including the number of people fatally injured and the number of pedestrians killed. We successfully developed a predictive classification algorithm that utilizes logistic regression to identify crash severity. Our model achieved a validation accuracy of approximately 99.997% and 100% accuracy against our test data. Based on this model, first responders could have a preliminary prediction of crash severity and allocate the appropriate emergency services. Moreover, the algorithm will help reduce traffic congestion by allowing for more efficient emergency responses and a clearer understanding of traffic hotspots in the region.

***K*eywords** Machine Learning · Virginia · Crash Data · Logistic Regression

1 Motivation

Virginia boasts "the third largest state highway system in the United States" [1]. Despite its impressive road system, Virginia has the seventh highest average commute time, which was reported to be 28.2 minutes in 2015. Much of this is due to the state's significance as a part of the Washington Metropolitan Area. In fact, the Washington Metropolitan Area, which includes Northern Virginia, has become notorious for having the highest rate of traffic congestion in the nation [2].

With commute already one of the most dreaded parts of a Northern Virginian's day, it becomes a nightmare when an accident unfortunately occurs. In order to minimize the additional traffic congestion in Northern Virginia, our machine learning project aims to identify the severity of a crash from given features about the accident. Our supervised classification model could be directly applied by first responders and the Virginia Department of Transportation to determine the necessary emergency services to send and an efficient way to clear traffic congestion in a timely manner.

2 Dataset

The dataset we chose to analyze is titled "Crash Data," courtesy of the Virginia Department of Roads' "Virginia Roads" website [3].

3 Related Work

Prior research has been done in the field of crash data. For example, in 2017, students under UVA's Department of Systems and Information Engineering and Department of Computer Science developed a machine learning model utilizing mixed effect logistic regression that could predict crash fatalities. Their research analyzed demographic and

police crash data, with an emphasis on interpreting the relationship between the probability of fatalities and the location of the crash [4].

However, work has also been done to analyze other factors that impact crash data. Some of these factors include social media (e.g., tweets) and kinematics data such as lateral acceleration and pedal position [5, 6]. These studies also test a variety of models, including support vector machines, random forests, and adaptive boost models. However, these models either had a success rate of 70-88%, or were not specifically tied to crashes in Virginia.

Our dataset appears to contain additional information not utilized by these models. More specifically, the "Crash Data" dataset includes features like light condition, roadway surface type and defect, intersection type, and accident-specific information (e.g., if the crash was a hit-and-run, whether a motorcycle was involved or not, if animals were involved, etc.) that could give us more insight into the severity and resources necessary to assist in a crash setting, and subsequently produce a stronger model.

4 Methods

4.1 Data Visualization

Prior to cleaning and experimenting with our data, we first wanted to visualize it. Our main goal with doing so was to get a better sense of our data, and determine if there were any particular characteristics that we could draw from the visualizations that would inform our steps further in the process.

To start, we produced a bar graph to represent the counts of the different crash severity types in Northern Virginia. Our dataset categorized crash severity into 5 types: (1) Property-Damage Only, (2) Visible Injury, (3) Nonvisible Injury, (4) Severe Injury, and (5) Fatal Injury.

Figure 1 confirmed our beliefs that there would be fewer severe crashes than non-severe ones. In particular, we noticed that a majority of the data appeared to be labelled as property-damage-only crashes. This is a key insight for our data, as it highlighted the need to utilize stratified sampling when we split our data into training and test data–otherwise, our training data could be an improper representation of our original dataset.

After this, we plotted our data points based on the level of their crash severity for all of Virginia and only Northern Virginia, as seen below.

Figures 2 and 3 offered unique insights for two main reasons. Firstly, these plots highlight the greater frequency of crashes along major highways/state-roads, which could therefore be an important factor in our model. Secondly, the plots (especially that of Northern Virginia, our area of interest) appear to indicate some form of clustering for non-visible injury crashes, which are labelled as yellow points on the plots. This clustering signifies that a potential clustering algorithm could be valuable as our model–based on the plots, crashes of similar severity occur near each other, and could therefore potentially increase the accuracy of our model.

Lastly, given the possibility of interstates/road-types to impact crash severity, we produced another set of bar graphs that plotted crash severity counts based on the road system. There were four possible road system categorizations (Interstate, Primary, Secondary, and Other), and to help with our analysis, we also pulled the exact frequency counts of crash severities for each type of system.

Interstate systems had 32000 property-damage-only crashes, 8484 visible injury crashes, 1705 nonvisible injury crashes, 1206 severe injury crashes, and 77 fatal injury crashes. Primary systems had 34680 property-damage-only crashes, 10145 visible injury crashes, 6656 nonvisible injury crashes, 1757 severe injury crashes, and 168 fatal injury crashes. Secondary systems had 43034 property-damage-only crashes, 12535 visible injury crashes, 8275 nonvisible injury crashes, 2642 severe injury crashes, and 215 fatal injury crashes. All other systems had a combined total of 24429 property-damage-only crashes, 6692 visible injury crashes, 5850 nonvisible injury crashes, 1439 severe injury crashes, and 71 fatal injury crashes.

From this analysis (Figures 4 to 7), we were able to find particular differences between road systems (e.g., interstates had the fewest nonvisible injury crashes, secondary systems had the most of each type of crash, primary roads had more visible injury crashes than interstate systems). These differences therefore seem particularly valuable, making this feature appear like a particularly useful one for our model building process. Once our final model is built, we can refer to the feature importance of this feature (among others) to determine for certain if these differences are significant for our model's classification of crash severity.

4.2 Data Cleaning

The dataset contains vehicle crash information for the entire state of Virginia. Since we aim to investigate crash history and existing relationships within Northern Virginia, we filtered the dataset to only this region. After doing so, the dataset still contained over 200,000 data points that would be separated into training and testing sets.

In order to further clean the dataset, we removed many features which we believed would not have any meaningful correlation with crash severity or provided repeated information. Such features included government identification of accident reports and location information that only had one value for Northern Virginia. Additionally, we decided to remove features that listed extremely specific information with very little frequency among values, such as street and route names, that could potentially skew our results. To compensate for these location features, we rely on latitude and longitude coordinates. Finally, although our dataset contained very few missing data, we also removed features that listed a majority of its values as "Unknown" or "Not provided."

The final set of features we use to find any relationships with crash severity in Northern Virginia are featured in our Appendix under Figure 8. Prominent features include location, weather type, and injury and road descriptions.

4.3 Data Preparation

The data contains both numerical and categorical features, which led us to use a pipeline to scale the numerical data using StandardScaler and represent the categorical data using OneHotEncoder. The data was then split into training and testing sets, such that a stratified sample composing 80% of the data was selected for training. We utilized a stratified sample in order to accurately represent all types of crashes because property-damage-only crashes composed approximately 66.4% of the data.

5 Experiments

In order to find the best machine learning model that predicts crash severity based on the features in the crash dataset, we trained a variety of classification algorithms, detailed below. For each classification algorithm, five one-versus-all models were trained to map the features in one of the five crash severity indexes.

Because clusters were observed in the preliminary data visualization, we theorized that a K-nearest-neighbors algorithm would perform well in classifying the data. As a preliminary experiment, we ran a K-nearest-neighbors algorithm with k = 5 neighbors and three-fold cross-validation. The accuracies yielded by this algorithm were [0.98952934, 0.99029955, 0.99002893].

In addition, we trained a logistic regression model with 500 maximum iterations and used three-fold cross-validation [7]. We had to utilize a high number of iterations, as a smaller number of iterations would prevent our model from converging. The accuracies yielded by this algorithm were [1. , 0.99995837, 0.99995837].

After that, we attempted to fit an SVM model to our data. However, the model would not converge even after hours of training. We believe that this was due to the high dimensionality of our data–with over 30+ features in our dataset, it may have been challenging for an SVM model to find a clean division in the data, regardless of the kernel option.

Then, we trained two decision tree models using maximum depths of 2 and 3. After fitting the model, testing against the validation data yielded accuracies of [0.96402923, 0.96407086, 0.96413331] for a depth of 2. However, for a depth of 3, we achieved accuracies of [0.99891755, 0.99893836, 0.99900081]. We were interested to see what the decision trees deemed as useful variables, so we created a diagram of our models, as shown in Figures 9 and 10. The primary split occurred along the K-PEOPLE variable, followed by splits based on PERSON-INJURED and PEDESTRIANS-INJURED. In the model with a depth of 3, there was an additional split based on PEDESTRIANS-KILLED. This diagram highlights important variables that could be playing a role in some of our other models. It's also important to note that the decision tree with a depth of 2 was not able to classify all five crash severity types as there was no leaf for property-damage-only crashes. In contrast, the decision tree with a depth of 3 was able to classify all of the crash severity categories.

Based on these results, the logistic regression model appeared to perform the best out of all the different models we trained. Therefore, we decided to move forward with this model for our hypertuning and final testing procedures.

The primary hyperparameter for which the logistic regression model was tuned was 'C,' which signifies the degree of regularization strength in the model. Based on a grid search that examined values of C = 0.001, 0.1, 1, and 10 over 1000 iterations, the optimal hyperparameter was C = 10. This hypertuned logistic regression model yielded a cross-validation accuracy of 99.997%.

6 Results

Our final hypertuned logistic regression model achieved a validation accuracy score of 0.9999791835800079. The model was then applied to our test data and yielded 100% accuracy, with all 5 levels of our categorical response variable being properly predicted for each data point. Figure 11 shows the confusion matrix summarizing the results.

Upon obtaining these results, we were interested in seeing what features were deemed most important by the hypertuned model for each level of our response variable. We found that PEDESTRIANS-KILLED was most important for severe-injury and fatal-injury crashes, PEDESTRIANS-INJURED was most important for visible-injury and non-visible injury crashes, and K-PEOPLE was most important for property-damage-only crashes.

After looking more closely at the feature importance for each crash severity level, we found that the following features were deemed most important for each crash severity level (with the most important feature and its corresponding weight as well):

Crash Type	Important Features	Most Important	Weight of Most Important Feature
Property-Damage Only	K-People Pedestrians-Killed Pedestrians-Injured Veh-Count	K-People	7.73752864227459
Nonvisible Injury	K-People Pedestrians-Killed Pedestrians-Injured Veh-Count	Pedestrians-Injured	4.463147998484327
Visible Injury	K-People Pedestrians-Killed Pedestrians-Injured	Pedestrians-Injured	4.855048292587223
Severe Injury	K-People Pedestrians-Killed Pedestrians-Injured Persons-Injured	Pedestrians-Killed	3.8502853792733185
Fatal Injury	K-People Pedestrians-Killed Pedestrians-Injured Persons-Injured	Pedestrians-Killed	2.9991195845109524

Our full Collab document, which includes all of our procedures and code, can be viewed at this link: https://colab.research.google.com/drive/1EVs5nsd85WIgITsBXP_6yEnXHCuuwCKx.

7 Conclusion

In this project, we trained a logistic regression model based on the crash dataset that predicts crash severity in our test data with 100% accuracy. The number of pedestrians killed, pedestrians injured, overall people injured, and number of vehicles in the accident were identified as the most important features in the developed classification algorithm. All other features in the dataset had weights less than 0.01. This result is interesting because the features identified as most important to the model are also the easiest to determine on-site of an accident, whereas the other features in the dataset may not be readily available after the crash.

Thus, the developed model can be used to accurately predict crash severity based on initially-reported features from an accident scene regarding the number of injuries. Based on the model predictions, the appropriate emergency response resources may be allocated according to projected crash severity.

While the model only has 3-4 significantly important features, it was trained based on all 50 features in the dataset. However, this may not be as practical in real life because not all of these features may be readily known immediately after a crash. A future step would be to train the classification algorithm only based on the top 5-10 most important features and see if the same degree of predication accuracy can be achieved.

Additionally, the data was extensively cleaned and filtered by the original source and our team. For example, the data was filtered to only reflect crashes in Northern Virginia. We may expand this model in the future by including data from more regions.

In addition to being immensely useful for emergency services in Northern Virginia, this model can help further understand traffic hotspots based on accident severity and be used to inform critical measures that alleviate traffic congestion.

8 Team Contributions

As a team, we collaborated on cleaning the data, debugging the training process of each classification algorithm, and constructing the proposal, checkpoint, and final report documents. Specifically, Nikita was responsible for writing code to automatically import the Crash Dataset from GitHub, pipelining the data, training the K-nearest Neighbors algorithm, and hypertuning the final model Supriya and Ankit were responsible for developing the data visualizations and formatting the report to Overleaf. Ankit also helped with running the logistic regression model, attempting the SVM model, and developing the summary feature table. Supriya conducted and performed the analysis of the decision tree model. All members collaborated on applying the hypertuned model to the test data and identifying the most significant model features.

9 Acknowledgements

We would like to wholeheartedly thank Professor Nguyen and the TAs, especially our grader Virginia Layne Berry, for their support and aid throughout this course and project. We would also like to thank the University of Virginia's Department of Computer Science for providing the resources necessary to accomplish this project. Amidst the global pandemic, we are grateful for the community that formed around ML4VA and hope that everyone continues to stay safe during these difficult times.

10 Appendix

Figure 1: Crash Severity Counts in NOVA

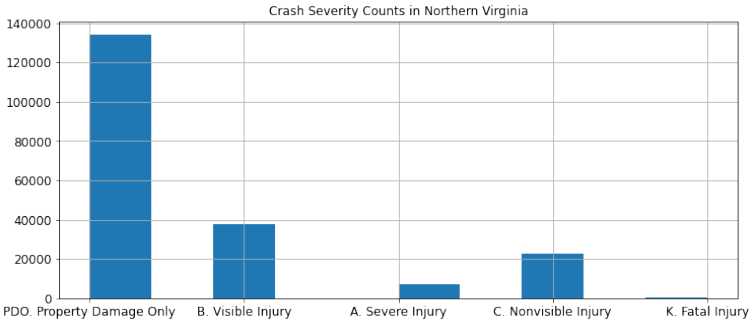

Figure 2: Plot of Crashes in VA

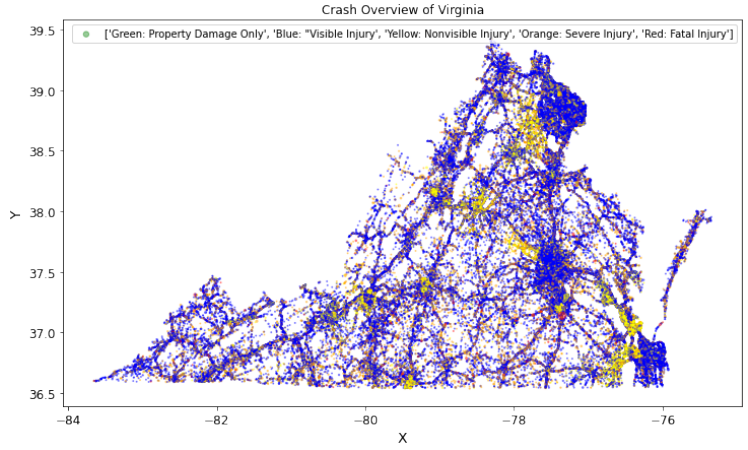

Figure 3: Plot of Crashes in NOVA only

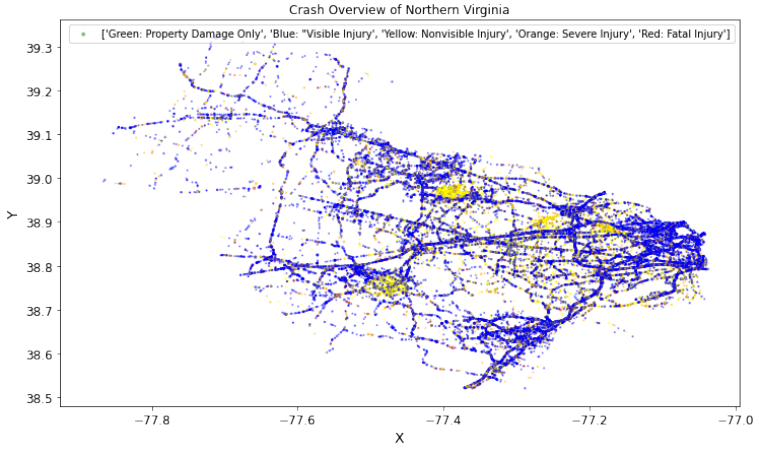

Figure 4: Interstate Crash Severity Distribution

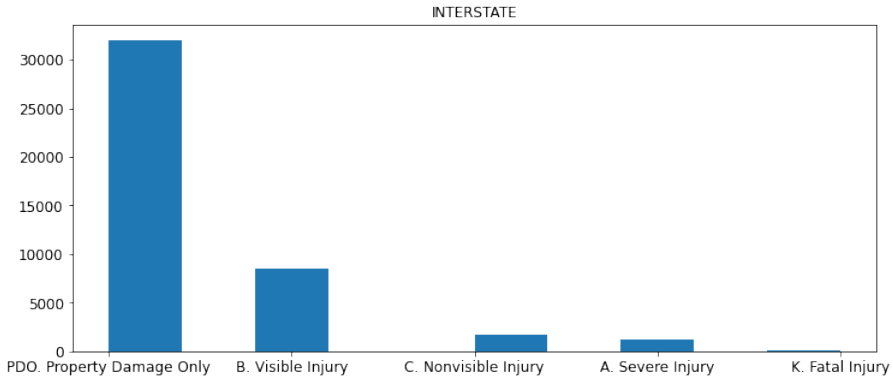

Figure 5: Primary System Crash Severity Distribution

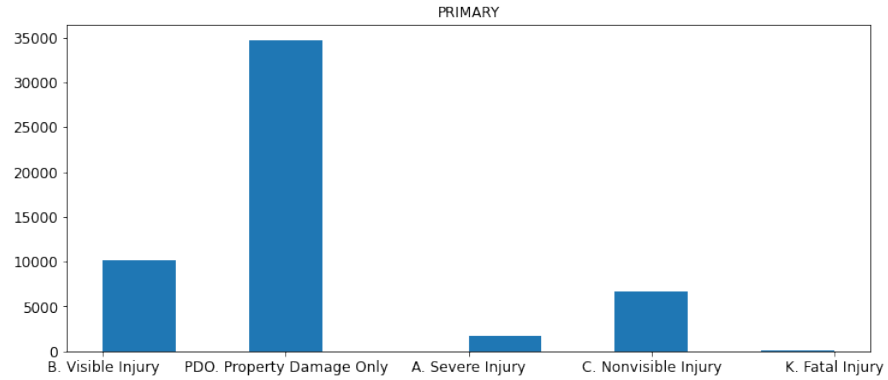

Figure 6: Secondary System Crash Severity Distribution

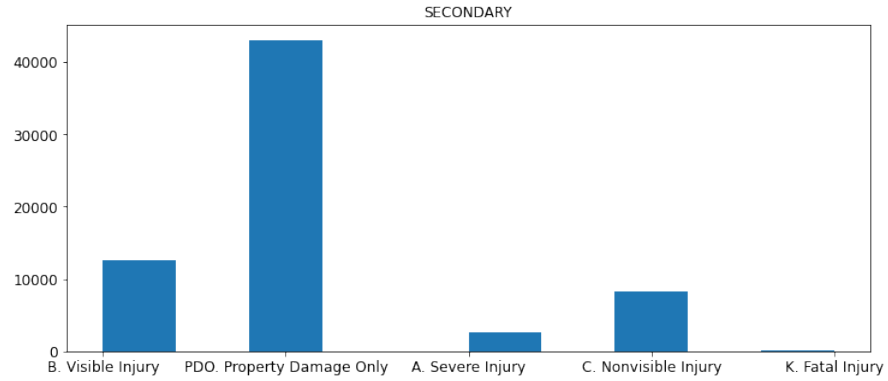

Figure 7: Crash Severity Distribution for Non-Categorized Systems

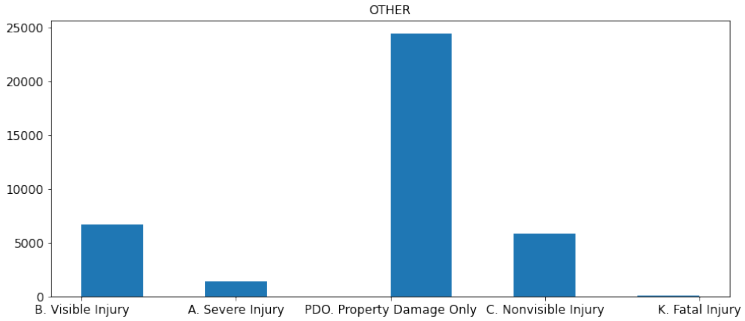

Figure 8: Features Used in Model Training

```
Index(['X', 'Y', 'CRASH_MILITARY_TM', 'K_PEOPLE', 'A_PEOPLE', 'B_PEOPLE',
       'C_PEOPLE', 'PERSONS_INJURED', 'PEDESTRIANS_KILLED',
       'PEDESTRIANS_INJURED', 'VEH_COUNT', 'COLLISION_TYPE',
       'WEATHER_CONDITION', 'LIGHT_CONDITION', 'ROADWAY_SURFACE_COND',
       'RELATION_TO_ROADWAY', 'ROADWAY_ALIGNMENT', 'ROADWAY_SURFACE_TYPE',
       'ROADWAY_DEFECT', 'ROADWAY_DESCRIPTION', 'INTERSECTION_TYPE',
       'TRAFFIC_CONTROL_TYPE', 'TRFC_CTRL_STATUS_TYPE', 'WORK_ZONE_RELATED',
       'FIRST_HARMFUL_EVENT', 'FIRST_HARMFUL_EVENT_LOC', 'ALCOHOL_NOTALCOHOL',
       'BELTED_UNBELTED', 'BIKE_NONBIKE', 'DISTRACTED_NOTDISTRACTED',
       'DEER_NODEER', 'DROWSY_NOTDROWSY', 'DRUG_NODRUG', 'GR_NOGR',
       'HITRUN_NOT_HITRUN', 'LGTRUCK_NONLGTRUCK', 'MOTOR_NONMOTOR',
       'PED_NONPED', 'RR', 'SPEED_NOTSPEED', 'SCHOOL_ZONE', 'SENIOR_NOTSENIOR',
       'YOUNG_NOTYOUNG', 'MAINLINE_YN', 'NIGHT', 'PHYSICAL_JURIS', 'FUN',
       'FAC', 'AREA_TYPE', 'SYSTEM'],
      dtype='object')
```

Figure 9: Decision Tree with Depth = 2

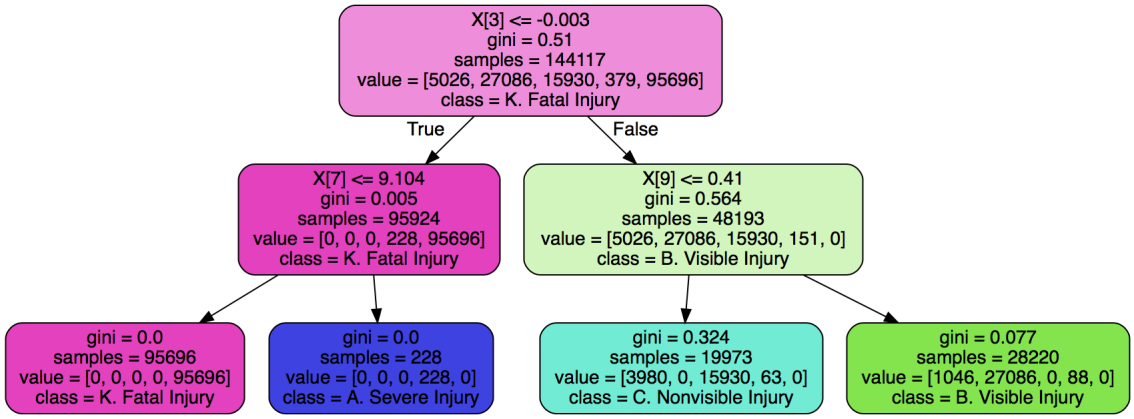

Figure 10: Decision Tree with Depth = 3

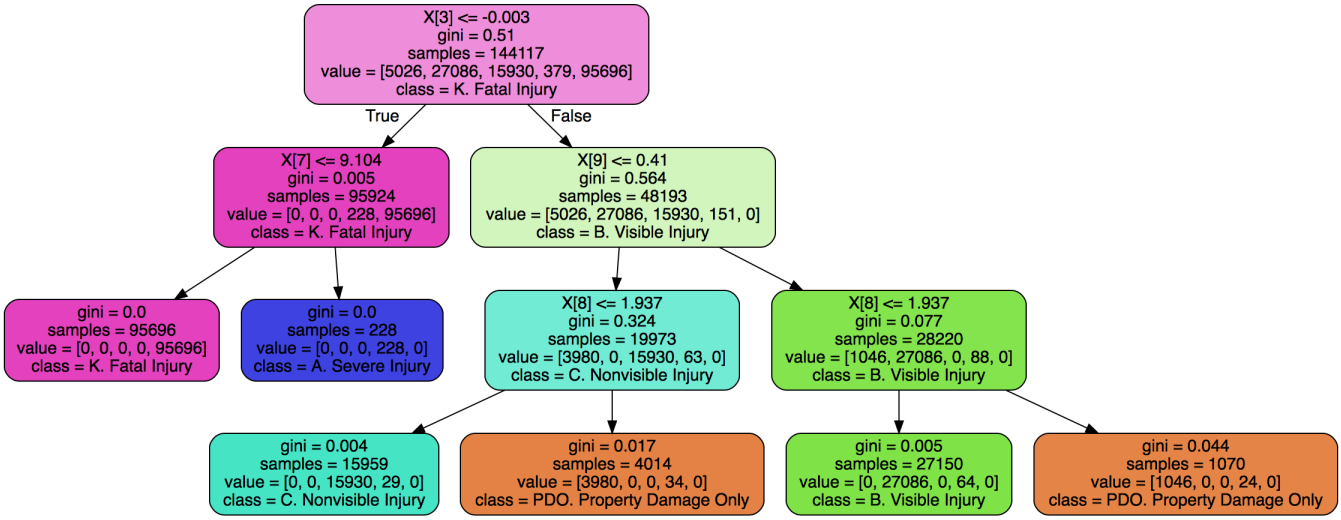

Figure 11: Final Logistic Regression Model Confusion Matrix Results

```
array([[ 1257,     0,     0,     0,     0],
       [    0,  6771,     0,     0,     0],
       [    0,     0,  3982,     0,     0],
       [    0,     0,     0,    95,     0],
       [    0,     0,     0,     0, 23924]])
```

References

[1] Virginia Department of Transportation. Virginia's highway system. http://www.virginiadot.org/about/vdot_hgwy_sys.asp. Accessed 02-16-2020.

[2] Virginia Performs. Traffic congestion. https://vaperforms.virginia.gov/transportation_trafficCongestion.cfm. Accessed 02-16-2020.

[3] Virginia Roads. Crashdata basic. https://www.virginiaroads.org/datasets/crashdata-basic. Accessed 02-16-2020.

[4] UVA Data Science. Can you predict motor vehicle accidents? https://datascience.virginia.edu/projects/can-you-predict-motor-vehicle-accidents. Accessed 02-16-2020.

[5] Zhenhua Zhang, Qing He, Jing Gao, and Ming Ni. A deep learning approach for detecting traffic accidents from social media data. *arXiv*, 2017.

[6] Osama A. Osman, Mustafa Hajij, Peter R. Bakhit, and Sherif Ishak. Prediction of near-crashes from observed vehicle kinematics using machine learning. *SAGE Journals*, 2019.

[7] SciKit Learn. Logistic regression. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html. Accessed 04-02-2020.