Crash Risk Prediction Model using Data Science

This manuscript (permalink) was automatically generated from uiceds/cee-492-term-project-fall-2022-swifties@402dcd6 on December 10, 2022.

Authors

- **Lara Diab**
  0000-0001-8489-2015 · diablara
  Illinois Center for Transportation; Department of Civil and Environmental Engineering, University of Illinois Urbana-Champaign · Funded by The Grainger College of Engineering

- **Renan Santos Maia**
  0000-0002-0877-4006 · renanssmaia · resim__
  Illinois Center for Transportation; Department of Civil and Environmental Engineering, University of Illinois Urbana-Champaign · Funded by The Grainger College of Engineering

- **Gonzalo Farid Saud Medina**
  0000-0001-0001-0001 · fsaudm
  Illinois Center for Transportation; Department of Civil and Environmental Engineering, University of Illinois Urbana-Champaign · Funded by The Grainger College of Engineering

- **Johann Jhanpiere Cardenas Huaman**
  0000-0002-4695-7639 · Johann-Cardenas · transporter_pe
  Illinois Center for Transportation; Department of Civil and Environmental Engineering, University of Illinois Urbana-Champaign · Funded by The Grainger College of Engineering

Introduction

Overview

Road accidents are responsible for a significant number of injuries reported every year. According to the World Health Organization (**WHO**), approximately 1.30 million people die each year as a result of road traffic crashes (as of June, 2022). In addition to this, the cost of road traffic crashes represent roughly 3% of a the gross domestic product (**GDP**) of a country (Safarpour et al, 2020; WHO, 2022). Consequently, a thourough understanding of what factors influence these accidents on roads is of utmost importance. However, it is not easy to decide which specific conditions lead to these accidents. Different road, climate, vehicle and driver conditions affect the likelihood of a road user to be involved in a fatal/serious car accident.

The ability of predicting accurately the potential occurence of a car crash is a valuable contribution for road safety. A common approach found in the literature, is the use of crash records data for the development of prediction models, so that agencies can allocate funds to priority areas within the roadway network. However, given that the available budget for infrastructure maintenance and rehabilitation is quite limited, the adoption of countermeasures for all the facilities where crashes are likely to occur is not feasible from an economic standpoint. Therefore, the ability of informing drivers about potential risks is an attractive way to proactively compensate the aforementioned limitations. Moreover, with the deployment of connected and autonomous vehicles (**CAV**), this information can be provided optimally helping vehicles with processes such as route selection, and can also deliver real-time alerts to instigate drivers to take the necessary safety precautions to operate their vehicles (Yu et al, 2021).

Plan Proposal

Project Objective

The objective of this project is to use the extensive crash database collected by the Illinois Department of Transportation (**IDOT**), to explore, detect, analize and extract trends from it to later develop a crash risk prediction model based on its most relevant categorical data. The prediction model will take categorical data as inputs and will yield the likelihood of a crash as an output. By using the results, we could categorize different sections of a given road based on the probability of crash accidents occurence. Considering that this database will keep being fed and the previous entries could also be updated (such as the weather/lighting/pavement conditions), the predictive model aims to be used by navigation systems to encourage drivers to adopt more cautious behavior as they enter high-crash risk sections.

Work Plan

The plan to be carried out will follow the sequence described below:

1. Download, and store IDOT's crash databases. The datasets are available as an open data source on IDOT's website in .CSV format.
2. Explore and clean the data by deleting irrelevant columns, handling missing data, and removing unimportant observations. Notice that rows and columns will be labeled as unimportant or irrelevant based on the objective of this project, so any categorical data not related to our target prediction will be filtered out.

3. Tidy the data by organizing the variables into columns and the observations into rows.
4. Analyze and visualize the data by finding correlations both analytically and graphically.
5. Test different machine learning models to build a crash prediction model, according to the categorical variables.
6. Assess the results, discuss the findings and determine which algorithm performs better.

Description of the Dataset

IDOT has generated annual datasets with statewide crash locations produced by the Crash Information Section of the Illinois Department of Transportation. The crash data has been collected throughout the years using Application Programming Interfaces (**APIs**) that provided streaming traffic incident data. There are about 300,000 accident records per year in these datasets, and each record contains attributes that include conditions like the ones described below (among others that are not listed or described here because these are not relevant for this study):

1. Time and date (day, month, year)
2. Coordinates (x,y)
3. Type of collision
4. A quantitative description of fatalities and injuries
5. Crash severity classification based on their impact on traffic
6. The road surface condition (“Dry”, “Wet”, “Snow or slush”, “Ice”, or “Sand/Dirt/Mud”)
7. Road defects (“Debris on roadway”, “Rut/Holes”, “Unkown”, or “No defects”)
8. Lightning conditions (rated in a scale from 1 to 9)
9. Geometric characteristics of the road section
10. Work Zone (“construction”, “maintenance”, “utility”, “unknown”, or “N/A”)
11. Possible causes of the accident.

The datasets for different years are available for download as .CSV files at the IDOT’s website:

https://gis-idot.opendata.arcgis.com/search?groupIds=6d2862031a6d47c7a8c211e38e423e05

Exploratory Data Analysis

Open-source crash data is published by the Illinois Department of Transportation (**IDOT**) yearly. Each crash report was found to have extensive entries with up to 85 attributes, which include several independent variables to describe each crash occurence. Each dataset is filled out with observations according to the **IDOT** Traffic Crash Report SR 1050 Instruction Manual (2019), in which each entry represents an observation. The dataset for each year is available online in .CSV format on the **IDOT** website, and they contain observations arranged in rows and attributes organized in columns. The datasets from 2017, 2018, and 2019 were included in the Exploratory Data Analysis **(EDA)**. The datasets from 2020 and 2021 were discarded in this analysis due to the COVID-19 pandemic outbreak, which altered drastically the dynamics of traffic worldwide, and thus crash-related data (Yasin, Grivna & Abu-Zidan, 2021). Regarding the dataset size, each of them had originally over 300,000 entries (944,328 in total, once combined). The **EDA** will be carried out following the steps described in the following sections.

Reading Data

The selected datasets were imported to the widely-used Integrated Development Environment (**IDE**) and code editor **Visual Studio Code** using its CSV library. It was found that the latest report (2019) contained the following 5 additional attributes that could not be used since it was not included in any of the previous reports.

1. Access Control
2. Flow Condition
3. Did Involve Secondary Crash?
4. Urban Rural
5. Toll

It was later verified that any other attribute was arranged in the same order for each of the datasets, thus it was decided to discard the aforementioned additional information for the sake of compatibility between the datasets. After deleting these attributes, all 03 datasets were merged into a unified file, which contains 80 variables (columns) and 944,328 observations (rows). This final database served as the initial point to start with the following step: the cleaning process.

Cleaning Process

It was found that several independent variables would not provide fruitful information due to missing, unknown or incomplete data. First, this observation was obtained by visual inspection, and later by analyzing the number of different and unique values present in each of the attributes. Thus, the datasets were processed to filter out irrelevant or incomplete variables. For instance, information pertaining the location (latitude & longitude, or X & Y coordinates) has not been taken into account. A map plot was initially produced just to check that the entries corresponded to several locations within the state of Illinois, but no further use was needed. Columns containing codes describing the city, county or ID of the location where the crash took place have also been excluded. Columns involving duplicate information (e.g. two columns describing the same independent variable with a label and a number), and traffic structures were also removed. For few other independent variables, information that could potentially be useful was found to be significantly incomplete. For instance, this was the case of attributes such as the number of lanes and the type of intersection. As a consequence, these variables were not included in the clean dataset. Finally, additional cleaning was carried out for independent variables with a high number of description labels. For example, the “Railroad Crossing

Number" variable contained around 100 different values which would have not been handy information for the end-user. After filtering out all the attributes that won't be utilized for this analysis, the number of independent variables went down from 80 to 24.

When it comes to crash reports, several inconsistencies are unfortunately frequent. In the literature, for instance, it is mentioned that "investigation of traffic safety by means of crash records is a reactive approach, where researchers need to deal with imprecise, incomplete, inconsistent, and, sometimes, inexistent records", and that is why the acquisition of historical series to provide minimal consistency to the analysis of crashes to reduce misinterpretations and misleading conclusions is crucial (Hauer & Hakkert, 1989; Chin & Quek, 1997; Farmer, 2003). This justifies the need of dedicating a considerable amount of time, after filtering the columns (variables) of interest, to the cleaning process of the rows (observations). For each column, any observation labeled as "blank", "unknown", or "other" needs to be handled, by either replacing the original value for a more meaningful lable or by deleting it. For all variables, the rows having "blank" observations were immediately removed from the dataset.

Analysis and Visualization

In this section, a set of plots, charts, and visuals were developed using the following tools:

1. Julia Programming Language.
2. Python Programming Language.
3. Microsoft Power BI.

The plots are presented here to display visually the findings and trends, as a product of the **EDA**. The visualization include distributions of vehicle crashes over time, and for the identified variables discussed in this section of the report.

Crash Data Distribution

A visual representation of the number of crashes throughout time can be seen in **Figure** 1. This served as a foundational step for the later-developed visuals. Here, the data is subdivided by year, and then by month to get a quick glance of the number of crashes throughout time (**Figures** 2, 3, 4).

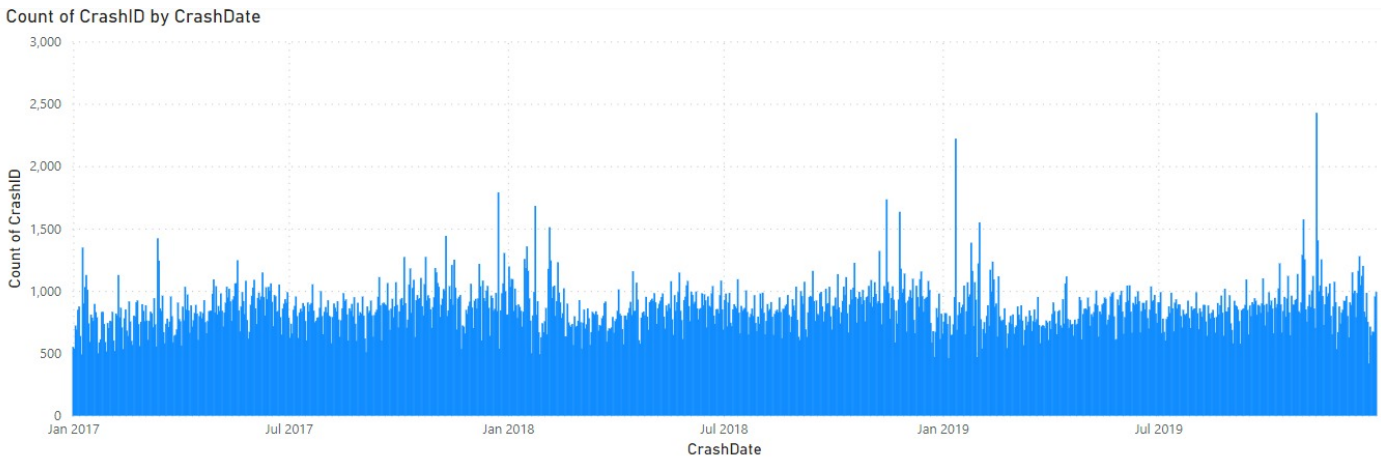

Figure 1: Distribution of crashes over time. IDOT's Crash Data from 2017, 2018 and 2019.

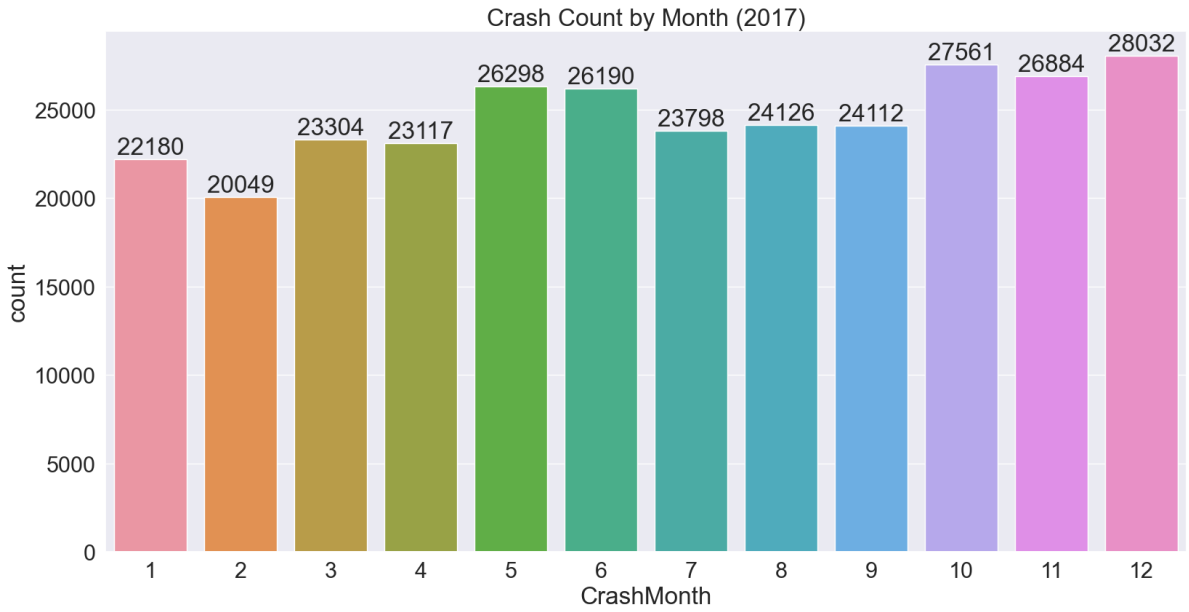

Figure 2: Bar Plot Historical Crashes. Crash reports from 2017 in the state of Illinois, USA.

For 2017, the month with most incidents is December ("12"), with **28,032** crashes (out of **295,651** for that year), and the month with the least crashes is February ("2"), with **20,049** incidents that month. These findings reflect the effect of weather conditions as well, given that during the very first and the last months of the year, the number of accidents increases. These months correspond to the winter season and englobes certain holidays where people might be very active and more accidents are likely to occur. The "*February effect*", as we would call it, where a decrease in the number of accidents can be observed, is not necessarily unexpected, because although it's still part of the winter, this is the month with the least number of days.

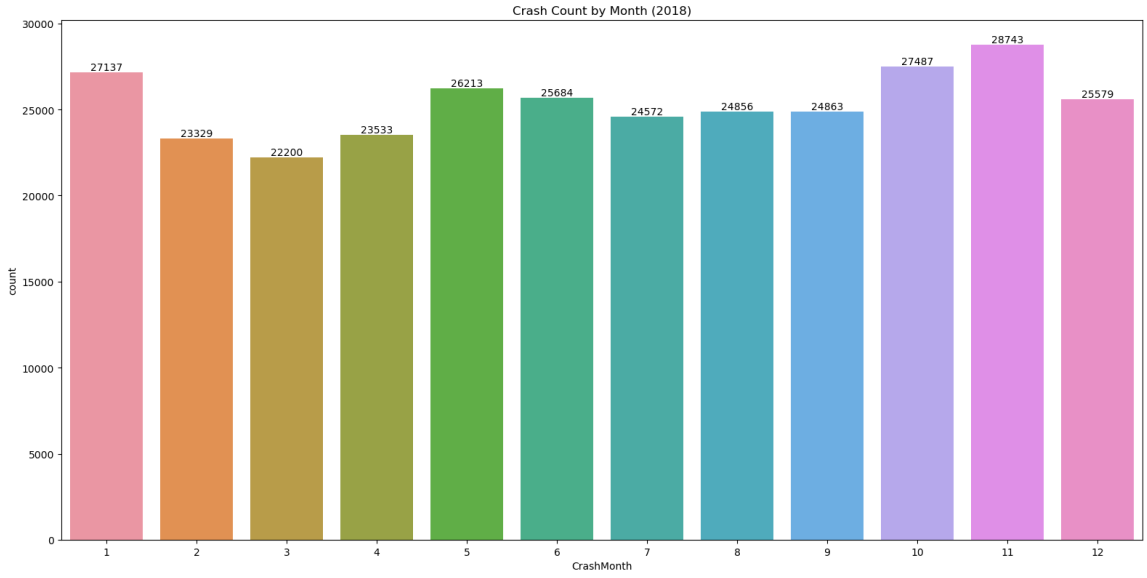

Figure 3: Bar Plot Historical Crashes. Crash reports from 2018 in the state of Illinois, USA.

Now, for 2018, different from the previous year, the month with the largest count of crashes is November ("11"), with **28,743** recorded crashes (out of **304,196**). The month with the least number of accidents is March ("3"), with **22,200** crashes. For this year, just like for 2017, the months where peaks happen can be associated with the worst seasons for a driver (winter).

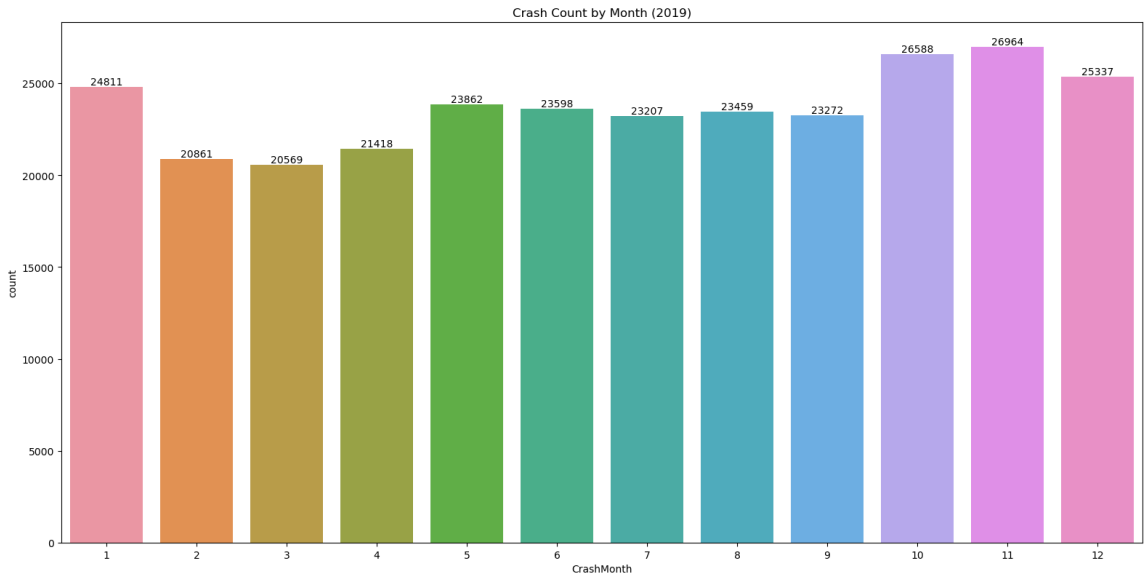

Figure 4: Bar Plot Historical Crashes. Crash reports from 2019 in the state of Illinois, USA.

For the year 2019, the month with the most number of crash records is November ("11"), with **26,964** recorded crashes (out of **283,946**), and the month with the least number of accidents is March ("3"), with **20,569** incidents. in a similar way to the years 2017 and 2018, the months with the highest records belong to the same season.

Crash Data Location

The location data contained in the dataset was discarded for the modeling purposes. However, this can provide an idea about the distribution of the crash data over the state of Illinois, giving insights in terms of the nature of the data and potential contributing factors. Using the "X" and "Y" information present in the dataset, a visual distribution of the recorded crashes is illustrated in Figure 5.

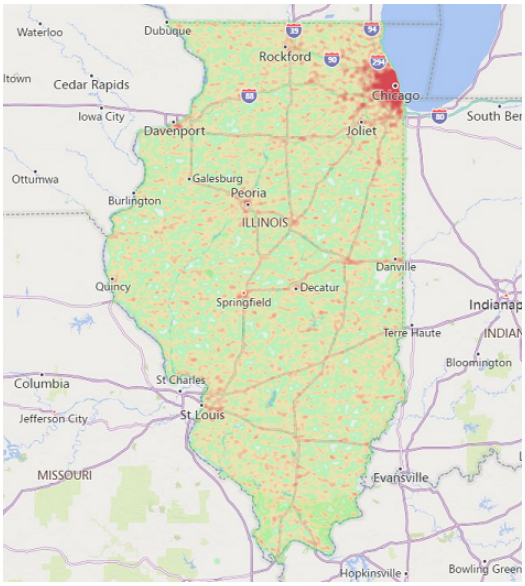

Figure 5: Distribution of crash occurrences. Crash reports from 2017, 2018, and 2019 in the state of Illinois, USA.

In this figure, as in a heat map, the red regions represent clusters of points, where every point is defined by an "X" and "Y" coordinate. "X" and "Y" describe the longitude and latitude of the accident location. The majority of crashes appear to have happened in cities and towns (urban areas). The best

example would be Chicago, where all around the area, the number of occurrences, or red dots, is significantly higher, which may be presumably because Chicago and the surrounding areas host a big fraction of the state's population. On a smaller scale, this is also visible in other cities and towns, such as Springfield and Davenport. In other locations, the recorded crashes are much more spaced out, with certain "hotspots" in some highways and roads.

Categorical Data Distribution

Figure 6 displays the different percentages of the different lighting conditions presented in the dataset. The condition with the most crashes associated with it is "Daylight", representing 66% of the crashes in the dataset. This might be an effect of the traffic being more concentrated from early morning until evening. During the night time, "Darkness/Lighted Road" accounts for **18%** of the crashes, and "Darkness", for **11%**. Eaxh of the conditions "Dawn" and "Dusk" account for **2%** of the recorded crashes.

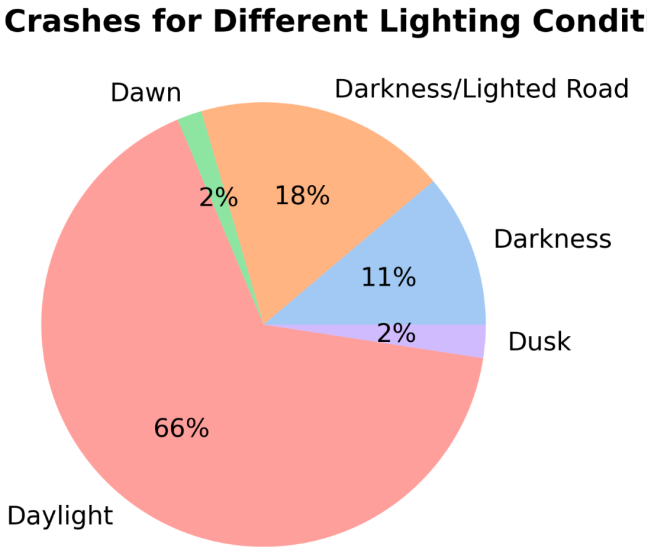

Figure 6: Distribution of accidents by Lighting Condition. From 2017 to 2019.

The chart in **Figure** 7 summarizes the analysis performed on the data on the influence of road surface conditions on the number of crashes. It was found that **76%** of the recorded crashes correspond to a "dry" road surface, which can be thought of as the least dangerous condition. For the not-too-favorable road surface conditions, **16%** of the crashes analyzed correspond to a "wet" road surface, **5%** to "snow", **2%** to "ice" and other "unknown" road surface conditions. Again, this can be interpreted as a reflex of the fact that rainy/snowy days are less frequent than "dry" days.

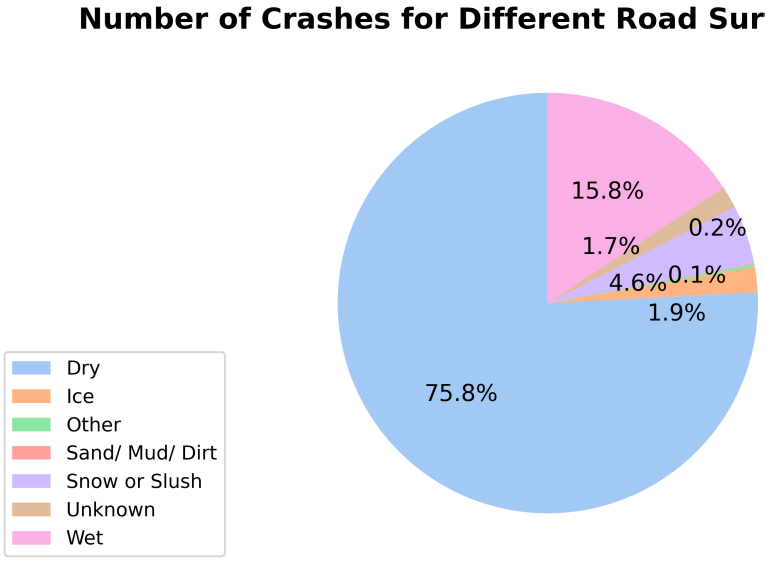

Figure 7: Distribution of accidents by Road Surface Condition. From 2017 to 2019.

When it comes to the distribution of crashes by road defects (**Figure** 8), the vast majority of the occurences (**808,835** out of **883,793** observations, which accounts for **91.52%**) happened were "no defects" were present in the location. Given that a great number of crashes happened without any road defects, it might be interesting to account for the likelihood of road defects being associated with a car crash. By observing the different percentages of the road defects accounted for in the dataset, "unknown" represents **78%** of the data, followed by 6% for "worn surfaces", "debris on the roadway" and "other" road defects, and **2%** for "ruts and holes" and "shoulder" defects.

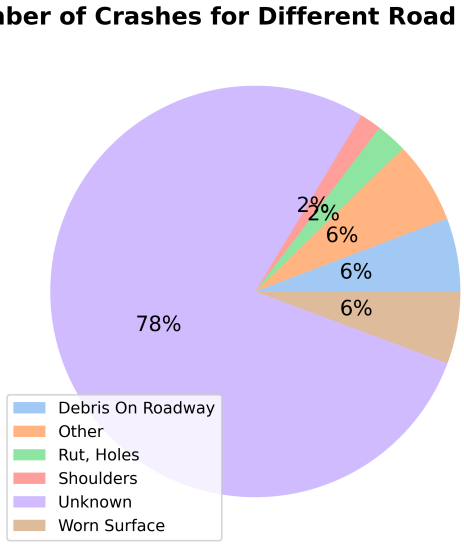

Figure 8: Distribution of accidents by Road Defects, From 2017 to 2019, excluding "No Defects" condition.

Correlation between Variables

As mentioned before, one of the objectives of analyzing this data is understanding how different road and environment conditions would affect crashes and their severity. This can be obtained by finding the associations between the different variables (columns) in the dataset, meaning how one variable is affected another variable. However, most of the variables are of categorical type, i.e., variables that are identified based on names or labels given to them and not based on numbers. This makes the built-in correlation functions in Python or Julia not helpful. One very commonly used method to measure the correlation between two categorical variables is Cramer's V statistic. Cramer's V is based on a nominal variation of Pearson's Chi-Square Test. Like correlation, the output takes values between

0 and 1 (inclusive), with 0 corresponding to no association between the variables and 1 corresponding to one variable being completely determined by the other. On the other hand, and unlike the usual correlation, there are no negative values. For this project, a function was created in Python that calculates the association between any 2 categorical columns using confusion matrix which can be obtained via built-in pandas method for categorical columns.

For this data that has 24 columns, running this function for every pair of variables would take a long time and may not give many insights. Therefore, the function was used to find how the column "CrashSeverity" is correlated with every other column. This column was chosen because finding how different conditions affect the severity of the crash is one of the most important outcomes of studying this dataset, and this would give an idea about the variables that have a significant impact on the crashes. The output is described in Table 1:

Parameter	**Association with CrashSeverity**
CrashYr	0.008
CrashMonth	0.027
CrashDay	0.005
NumberOfVehicles	0.101
DayOfWeekCode	0.011
CrashHour	0.025
CollisionTypeCode	0.259
TotalFatals	0.707
TotalInjured	0.706
NoInjuries	0.356
CrashSeverity	1.000
IntersectionRelated	0.141
RoadwayFunctionalClassCode	0.071
WorkZoneRelated	0.005
TypeOfFirstCrash	0.259
CityName	0.074
ClassOfTrafficWay	0.061
Cause1	0.166
TrafficControlDevice	0.098
TrafficControlDeviceCond	0.052
RoadSurfaceCond	0.033
RoadDefects	0.042
LightingCond	0.028
WeatherCond	0.031

Table 1: Correlation table (association with "CrashSeverity").

From **Table 1**, it can be observed that the best-correlated factor in relation to crash severity is the "TotalFatals" column, which indicates the number of fatalities for each crash, with a value of 0.707.

Similar observations can be made for the number of injuries. This makes sense because it is expected that the higher the severity of the crash, the higher the number of fatalities and injuries is expected to be. However, this is not very helpful for understanding how different conditions affect the severity of the crash. For this purpose, the columns that can be compared are: “IntersectionRelated”, “RoadwayFunctionClassCode”, “WorkZoneRelated”, “ClassOfTrafficWay”, “TrafficControlDevice”, “TrafficControlDeviceCond”, “RoadSurfaceCond”, “RoadDefects”, “LightingCond” and “WeatherCond”. Comapring these, it can be seen that presence of intersections has the highest correlation with the severity of the crash followed by the traffic control device. In addition, the characteristics of the workzone seem to have the least correlation with the severity of the crash. This observation can be useful to understand the dataset and get an idea about which variables are important for making predictions. This was done to get a general idea about the data, but for performing predictions additional details should be considered.

Trends

Given that not all the variables have to be present for a crash to occur, it can be seen that most accidents happen in the absence of adverse conditions. However, it should be taken into account that this reflects the fact that adverse conditions are exceptions, and accidents happen on a daily basis with other factors as underlying reasons such as the human behavior itself. However, adverse conditions do increase the likelihood of accidents and it can be observed an increase in the overall number of occurrences in specific hours (evening), days (weekends), and months (winter). The road type is also found to be directly correlated with the maximum speed limit, and as a consequence it is potentially tied to the number of accidents per day.

Potential Issues for Modeling

As long as the number of entries containing a value for an independent variable overcomes by large any other value for the same independent variable, it can be expected that the analysis will potentially experience problems related to “imbalanced data”, due to the uneven distribution of observations. Similarly, it can be seen that most of the independent variables are “classifications”, and therefore their entries do not provide meaningul numerical values to be analyzed or correlated. For some of them, it is possible to replace the text values by boolean variables, but for some others a rating system may be needed if a numerical interpretation is required. It can be noticed that most of the attributes are subjective observations trying to describe the potential causes of a crash, and may be dependent on the observer itself. However, the casualties are a meaningful numerical observation that will be thoroughly used througout this report.

Predictive Model

Given the nature of the database and the primary established goal of predicting the severity of crashes according to different combinations of scenarios, a classification problem is faced, for which the following models will be tested:

- a. Decision Tree
- b. Random Forest
- c. Convolutional Neural Network (CNN)

The dataset in this study is large enough (with hundreds of thousands of entries), so an advantage could be taken from this by dividing it into training and validation datasets. Another option derives from the fact that the IDOT's crash databases for different years are also freely available online in .csv format. Therefore, additional hundreds of thousands of entries could be used for validation (e.g. the data from the years immediately before the ones selected for building the dataset of this report, such as the 2016 dataset). This way, the full dataset could be used to train and build the model, and the validation datasets will afterwards be useful to decide on the appropriate model parameters needed to achieve the optimal model accuracy.

Since machine learning algorithms are generally unable to work with categorical data when fed directly into the model, there is a need to convert our independent variables (inputs):RoadSurfaceCond,:RoadDefects,:LightingCond and :WeatherCond into numbers, and the same will be required for our output variable since it will also be categorical (:CrashSeverity). The task of assigning numerical values to make use of them has to be handled with the aim of avoiding undesired biases in the assignment process. If we assigned a float or a integer value, our machine learning model may wrongly allocate a higher weight to variables with higher values, affecting the accuracy of the prediction model. To avoid this issue, we will encode our categorical features as one-hot numeric arrays, a technique that is presented hereafter.

One Hot Encoding

The one-hot encoding scheme, also known as 'one-of-K' or 'dummy' creates a binary colum for each category, and returns a sparse matrix or dense array (depending on the sparse parameters).Our inputs to this transform will be strings, denoting the values taken on by our categorical (discrete) features and our output will be a binary feature for each possible category with the value of 1 to the feature of each sample that belongs to the category, and a value of 0 for any other feature (Buitinck et al, 2013). An example is shown below for the variable *:LightingCond*:

Original Feature	**One-Hot Encoded Feature**
Darkness	[1, 0, 0, 0, 0, 0]
Darkness/Lighted	[0, 1, 0, 0, 0, 0]
Dawn	[0, 0, 1, 0, 0, 0]
Daylight	[0, 0, 0, 1, 0, 0]
Dusk	[0, 0, 0, 0, 1, 0]
Unknown	[0, 0, 0, 0, 0, 1]

Although, the One Hot Encoding technique will be useful to transform and preprocess our data so that our model can understand it better and learn from it more effectively, it comes with its own

advantages and disadvantages. We'll be analyzing the results as we test our model to report our findings.

Once the data has been transformed, we can use it to train our proposed models. The description, adequacy, results, and conclusions from every model tested is discussed in the following sections.

Models

a) Decision Tree Model

The first approach will be the development of a Decision Tree (DT) scheme. Decision Trees are non-parametric supervised learning methods, that can deal with large datasets without imposing complicated parametric structures, enabling them to predict the value of a target variable based on simple decision rules inferred from the data features. The objective is to find a set of decision rules that naturally partition the feature space to provide an informative and robust hierarchical classification model (Myles et al, 2004).

The DecisionTree.jl package available for Julia, provided us with a "DecisionTreeClassifier" model. This model requires the following hyperparameters:

- pruning_purity_threshold: (post-pruning) merge leaves having >=thresh combined purity (default: no pruning)
- max_depth: maximum depth of the decision tree (default: no maximum)
- min_samples_leaf: the minimum number of samples each leaf needs to have (default: 1)
- min_samples_split: the minimum number of samples in needed for a split (default: 2)
- min_purity_increase: minimum purity needed for a split (default: 0.0)
- n_subfeatures: number of features to select at random (default: keep all)
- rng: the random number generator to use. Can be an Int, which will be used to seed and create a new random number generator.

As the first approach, the values by default have been kept constant while the maximum depth of the decision tree is altered from scenario to scenario to evaluate the changes in the result. The first proposed model (Decision Tree) resulted in limited accuracy (78%) on the training data using the fully cleaned combined dataset. As a first estimate, this value could be seen as promising, however several issues could be immediately noticed. Despite testing several model parameters, the accuracy of the model did not improve considerably what indicates a problem in our dataset. Let's look at the general structure of our decision tree to understand this.

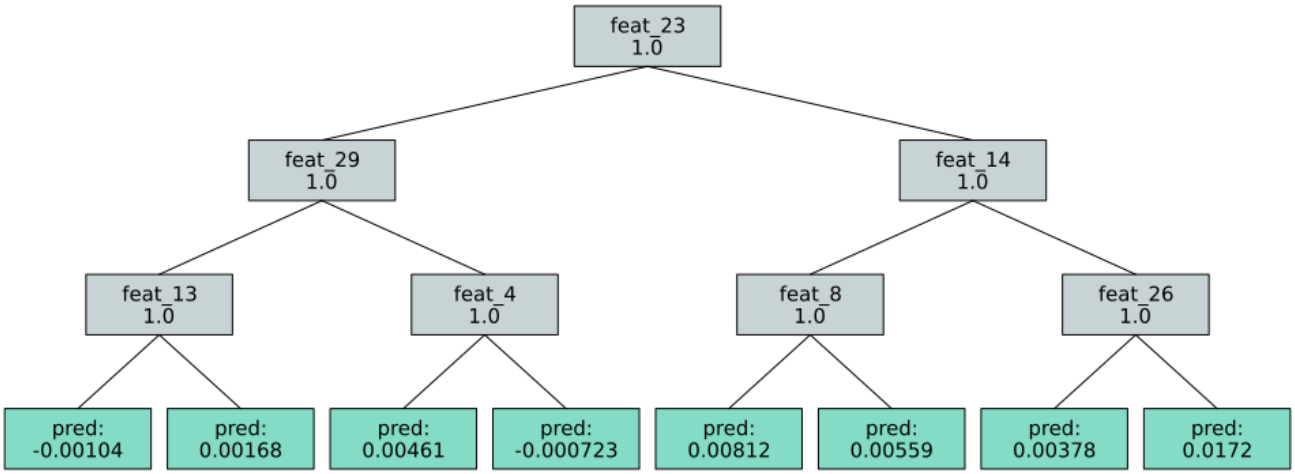

Figure 9: Decision Tree Structure. Using package: DecisionTreeClassifier.jl.

Figure 10 below shows the confusion plot for the decision tree model. It can be seen from the plot that model is predicting 691050 accurate values for "Property Damage" and 111 non-accurate values as "Injury" instead of "Property Damage". The prediction accuracy is much worse for the other two labels. For "Injury", the model predicted only 185 accurate values and predicted 189752 values for "Property Damage" instead of "Injury". Similarily, for "Fatal", the model predicted only 2 accurate values and inaccurately predicted 2692 "Property Damage" instead of "Fatal". This shows that the model is predicting "Property Damage" for most of the cases.

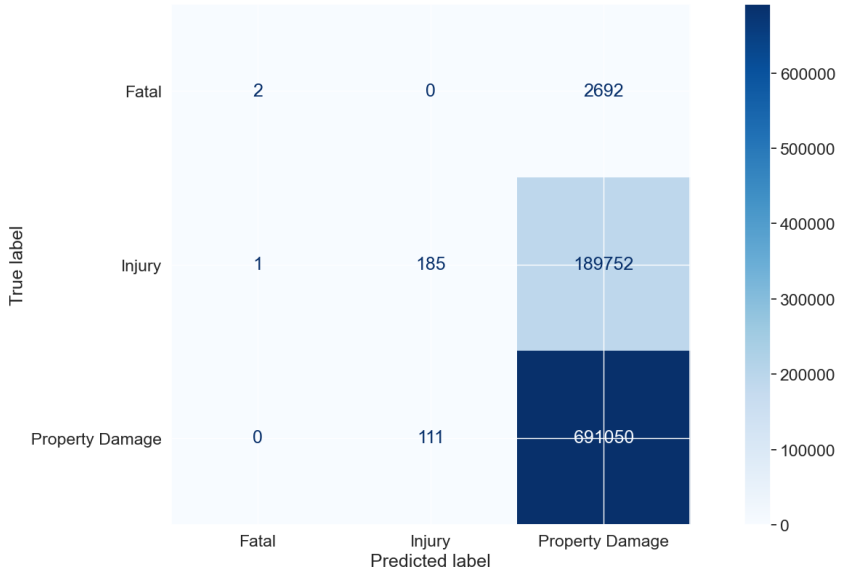

Figure 10: Confusion plot for the Decision Tree model

This issue can be summarized by stating that one of the possible outputs ("Property Damage"), the lowest severity crash type, dominates the dataset, accounting for almost 78% of the total number of cases. Consequently, a simple model that only predicts "Property Damage", independently of the entries, would have 78% accuracy. This imposes a huge bias in the criteria being used by the model to predict an output. As seen in the previous figure, most of the leaves of the decision tree get the right output just because the likelihood of predicting one of them is enormously higher than the likelihood of any of the other two.

Alternative strategies were brainstormed to overcome the imbalanced data issue. First, the DT models were re-trained using a database comprised of a sample of equal numbers of observations for the 3 different classes in the expected output vector. As this test didn't bring a considerable change either, a Random Forest (RF) model was also run for both datasets (full and reduced, equally sampled).

b) Random Forest

The Random Forest (RF) models are used to predict both categorical and continuous outputs. The background of the RF concept recalls for the presence of multiple classification trees, which partition the data using a sequence of binary splits on individual variables. The non-split nodes are called terminal nodes. Given the presence of multiple Decision Trees (DTs), the RF models use the bagging method to build decision trees as parallel estimators, which are finally averaged to give rise to the mean predictive model. It should be noted that improved RF estimations can be obtained by taking into account uncorrelated and difference between DTs, otherwise the final accuracy of the RF and DT

models would be similar. In Julia, the so-called “RandomForestClassifier” object can be used to build a RF model.

Given the limitations of the obtained DT results, a Random Forest (RF) model was implemented in order to evaluate if any increase in the model accuracy could be obtained. However, the accuracy obtained by the Random Forest model was also 78%. Figure 11 below shows the confusion plot for the random forest model. It can be seen in the figure that the model is working well only for “Property Damage” where it is predicting 691119 accurate values and only 42 non accurate ones. For “Injury”, the model is predicting 117 accurate values and is wrongly prediciting 189821 “Property Damage” values instead. In addition, for “Fatal”, the model is only predicting one accurate value and for the remaining ones, it is predicting “Property Damage” instead. Comapring the decison tree and the random forest models, the random forest is working even better for “Property Damage” and worse for the other 2, meaning, it is predicting more accurate values as “Property Damage” and less accurate values for “Injury” and “Fatal” keeping the overall accuracy the same at 78%.

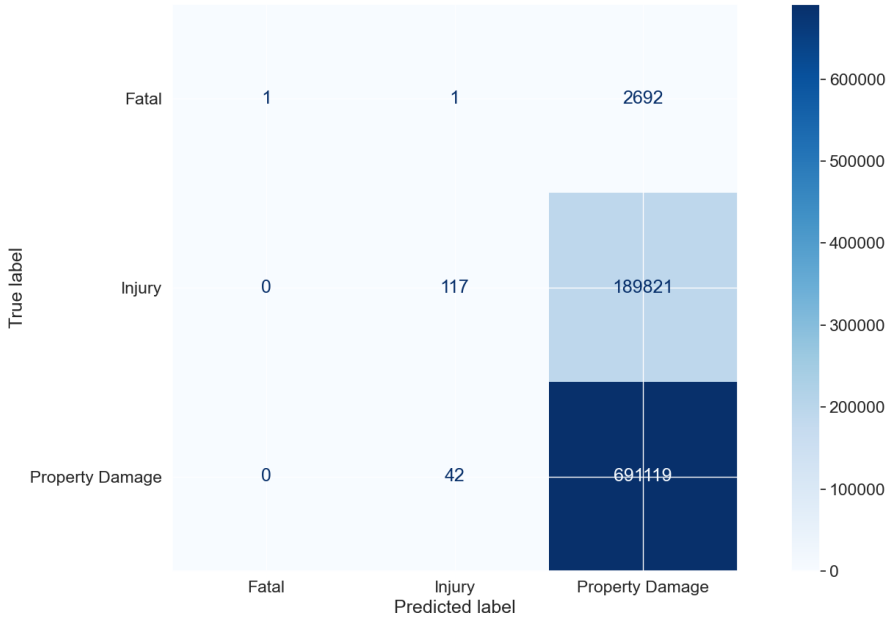

Figure 11: Confusion plot for the Random Forest model

c) Different Approaches for the Decision Tree and Random Forest Models

After achieving an accuracy of 78% for the decision tree and random forest models that were tested, different approaches were implemented to modify and adjust the data with the goal of achieving a higher accuracy in the predictions. These approaches are described below:

1- Removing rows with entries that were not very well understood

As mentioned earlier, the columns “RoadSurfaceCond”, “RoadDefects”, “LightingCond” and “WeatherCond” were the independent variables for this model. Each of these columns (attributes) has different possible values. Table 3 below shows the unique values for each attribute. For example, and as can be seen in the table, WeatherCond has 12 different possible values and for the one-hot encoding, this would create 12 columns just for the attribute “WeatherCond”. It was thought that reducing the number of the one-hot encoded columns would make the accuracy better as the model would not have as many columns to use for the prediction of the models. Thus, it was decided to

delete those rows including the label “Other” or “Unknown” in the attributes. These two values were chosen to be removed because they do not offer any significant or specific information about the corresponding attributes.

RoadSurfaceCond	RoadDefects	LightingCond	WeatherCond
Wet	Debris on Roadway	Darkness	Blowing Sand
Dry	No Defects	Darkness/Lighted Road	Blowing Snow
Ice	Other	Dawn	Clear
Snow or Slush	Rut Holes	Daylight	Cloudy/Overcast
Unknown	Shoulders	Dusk	Fog/Smoke/Haze
Other	Unknown		Freezing Rain
Sand/Mud/Dirt	Worn Surface		Other
			Rain
			Severe Cross Wind
			Sleet/Hail
			Snow
			Unknown

Table 3: Different values for each of the attributes chosen for the predictive model

After removing these rows, performing one-hot encoding again and running the decision tree/random forest models again, the same level of accuracy of 78% was achieved. This indicated that the higher number of one-hot encoded columns is not impacting the prediction. Figure 12 below shows the confusion matrix after implementing the first approach. It can be seen from the plot why the accuracy stayed at 78%. The model predicted accurate values in some cases but at the same predicted inaccurate ones in other cases.

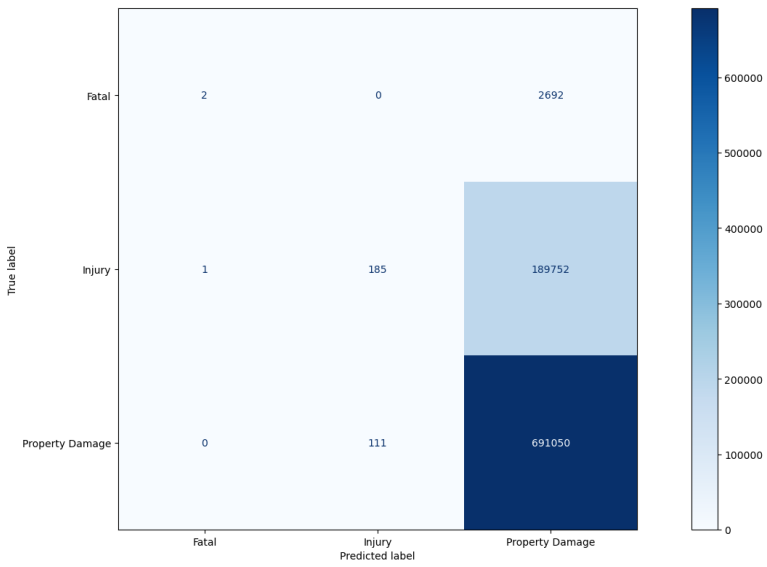

Figure 12: Confusion plot after implementing the first approach

2- Taking into consideration another independent variable

It was also decided to try considering one more column in the dataset and that could probably increase the accuracy. The reason behind this reasoning is if the model was offered more insight about the data, other than the surface road conditions, road defects, weather conditions and lighting conditions, this would probably help the model notice more relationships between the features and the dependent variable and thus increasing the accuracy. Therefore, and based on the previous analysis of correlations between the columns, it was found that the presence or absence of road intersections was relatively highly correlated with crash severity, it was re-considered as one of the features for modelling. This attribute has 2 values: Yes or No. One-hot encoding was performed again and the decision tree model was applied again. However, the accuracy was also 78%. This showed that increasing the number of attributes is not the solution to low accuracy, in this case. Figure 13 below shows the confusion plot for the model after implementing approach 2, which is adding one the "IntersectionRelated" attribute. It can be seen that did not help make better predictions.

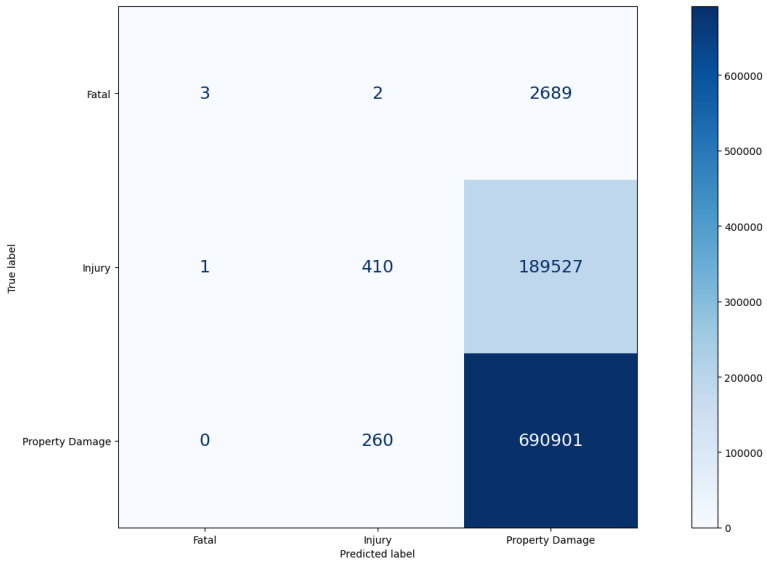

Figure 13: Confusion plot after implementing the second approach

3- Addressing the issue of imbalanced data

Because the problem was not the presence of several values for each column, dealing with the imbalanced dataset was another approach that was implemented by the team. This model is trying to predict the crash severity level which has three label options: "Property Damage", "Injury" and "Fatal". "Property Damage" constitutes roughly 80% of the data; 627706 rows out 809824 total rows (after cleaning the dataset), This imbalance is causing the model to predict "Property Damage" most of the time, decreasing the accuracy of the model. One way of addressing this is running the model with the same number of rows for each label. "Fatal" had the lowest number of rows in the data which was 2694. Therefore, the same number of rows was randomly selected from each of the other two labels: Property Damage and Injury. This approach also yielded approcimately the same accuracy on the training data which is 80%. This can be noticed by observing the confusion plot for this model. (Figure 14)

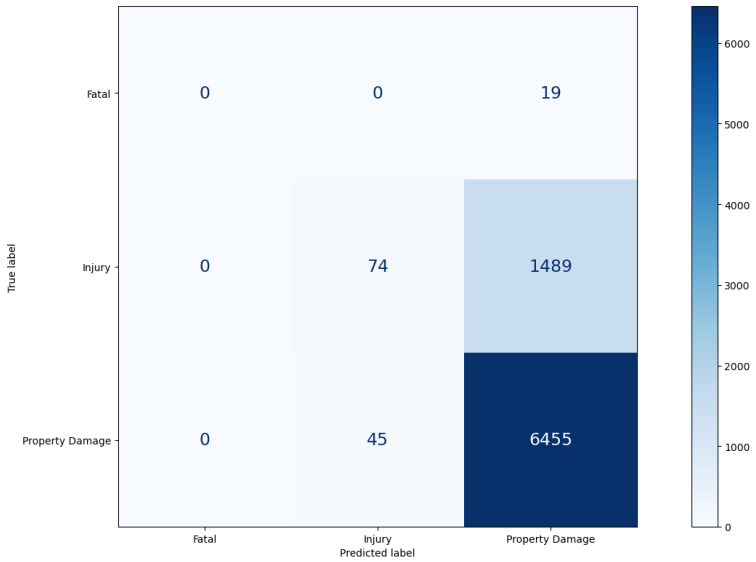

Figure 14: Confusion plot after implementing the third approach

4- Removing “perfect” conditions

Another way to look at the data imbalance of this dataset is by observing the features rather than the dependent variables. Part of the reason why the data is imbalanced is the fact that most of the time when the crashes happen, the coditions are “perfect”, meaning the weather condition is clear, the road has no defcts, the lighting condition is daylight and the road surface condition is dry. Therefore, one approach taken by the team was to remove those conditions and try to run the decision tree model for the remaining cases. This increased the model accuracy, at best, to 85%. This can be seen is Figure 15 below. The fraction of accurate predictions increased, compared to previous models and approaches, but this is still low for a training set.

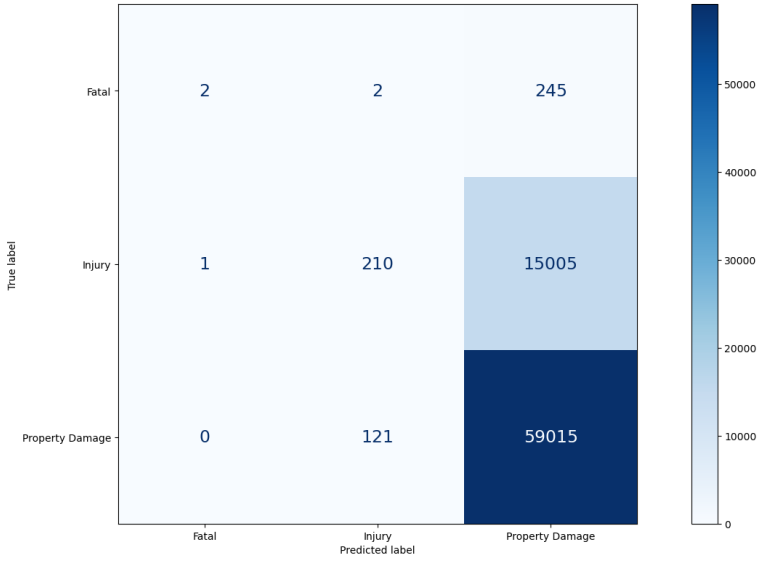

Figure 15: Confusion plot after implementing the fourth approach

c) Convolutional Neural Network (CNN)

According to Brunton and Kutz, practitioners would generally refer to Deep Convolutional Neural Networks when thinking of Neural Networks. Given its popularity, flexibility and effectiveness for classification problems, DCNN were used to model this problem.

The Deep Convolutional Neural Network was constructed using the Julia Machine Learning Library, Flux. This library allows for an easier construction of neural networks with predefined functions for different kinds of layers, activation functions, model training functions, and several other functions of interest when building up a neural network.

For this model, as a first attempt, the entire dataset was used. An important consideration to account for when using the Flux library is that the neural network expects a very specific structure of the data: a 3 dimensional matrix of the input data that contains the observations divided into vectors, and one-hot encoded variables.

During the training phase, the model included convolutional layers, dense layers (to reduce the dimensionality of the ouput), the "ReLU" activation function (one of the more popular activation functions), pooling layers (for a more efficient computation and reduce overfitting), a dropout layer (to specifically target overfitting), and a "Softmax" function (a generalized version of the functions used for classification problems). These functions where all put together using a "Chain" function. In the testing phase, the Dropout function is disabled.

The architecture of the Neural Network used is as follows:

```
model =Chain(       Conv((3,), 1=>2, pad=(1,)),       MaxPool((2,)),
Conv((3,), 2=>4, pad=(1,)),       MaxPool((2,)),       x -> reshape(x, :,
size(x, 3)),       Dropout(1),       Dense(28,3),       softmax   )
```

The error metric used for training this Neural Network was the logistical cross entropy loss function. This function matches the "Softmax" function output, and works as a generalized version of cross entropy loss functions used for "more-than-two" classes classification problems.

The hyper parameters used to train the neural network consisted of a batch size of approximately one third of the length of the original dataset (300000 observations), which is consistent with the amount of observations collected in one year (as described previously, the dataset consist of a compilation of data from 3 different years). The number of steps or "epochs" for training was chosen empirically, with 100 steps consistently showing a plateau in the accuracy improvement. Finally, the learning rate was defined as 0.01.

Similarly to the other predictive models used, the Convolutional Neural Network achieved an accuracy of 78.3%. The training process is illustrated in figure 10:

Convolutional Neural Network Training

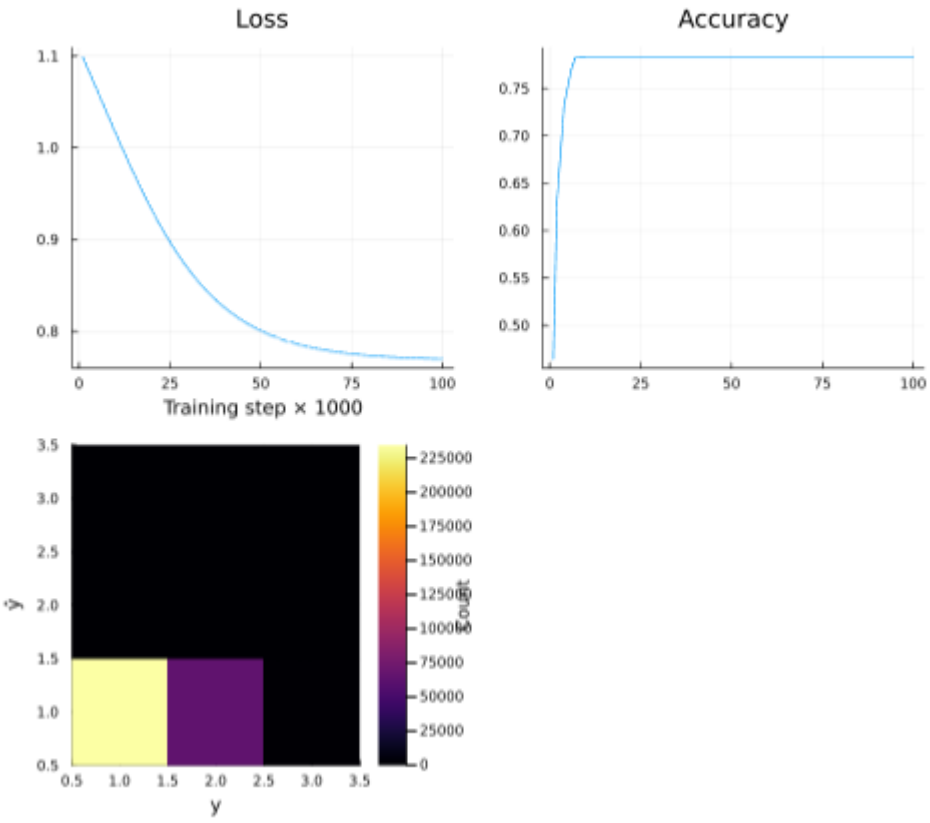

Figure 16: Using the complete dataset

Given that the results did not show a significant improvement compared to the Decision Tree predictive models, a “worst-case scenario” of the data was generated and used to train this model. In an attempt to reduce the outstanding difference of “Crash Severity” conditions, this new dataset excluded the “Clear” weather condition, “No defects” road defect condition, and “Dry” road surface condition. Since the dataset reduce its size to approximately 20000 observations, the batch size was reduced respectively to train the model.

With this “worst case scenario” dataset, the accuracy achieved reached 85.1%. The progress of training the Neural Network with this data are shown in figure 11:

Convolutional Neural Network Training

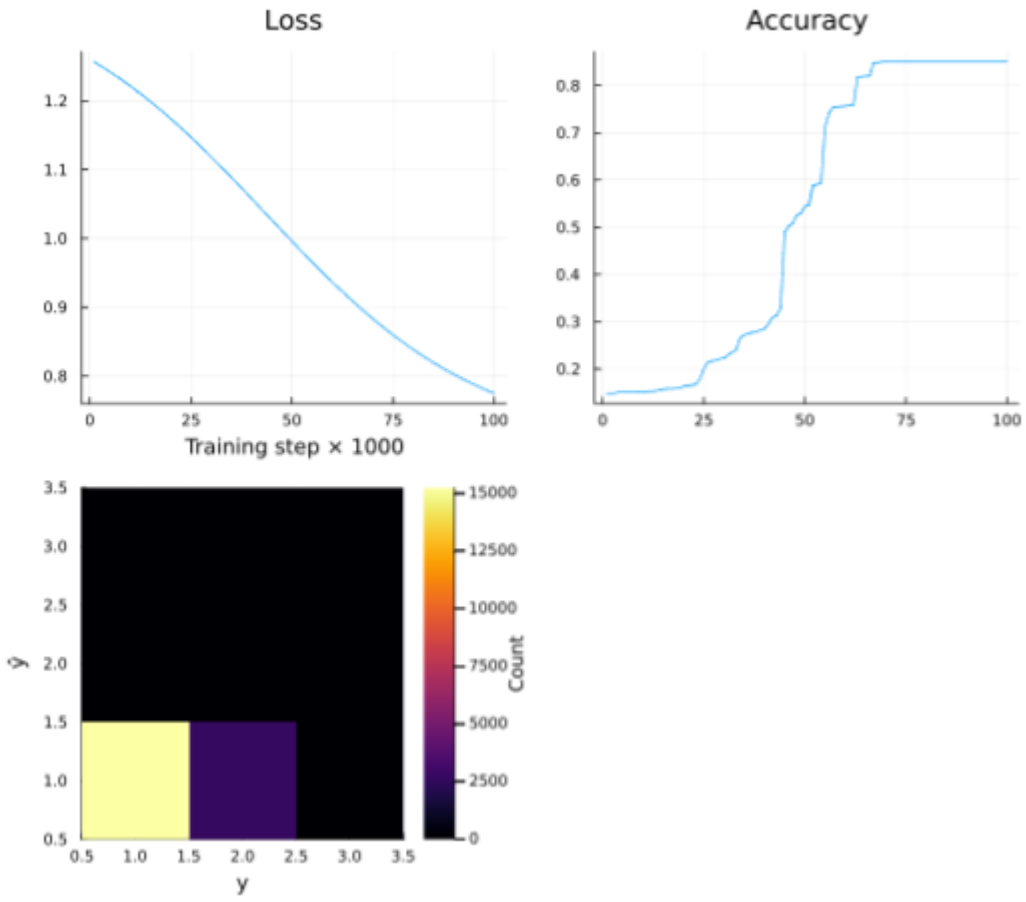

Figure 17: Using a reduced dataset

The results obtained using this predictive model may not be able to improve any further given the nature of the Convolutional Neural Networks and the dataset used to train the predictive models. It is relevant to note that:

1. Convolutional Neural Networks are designed to pick up spatial patterns in “gridded” data, which is not one of the characteristics of the data analyzed in this project. Hence, this predictive model’s effectiveness may be limited by the lack of spatial patterns in the data.

2. Since the amount of “Property Damage” as one of the possible “Crash Severity” is overwhelmingly greater than the other 2 possible classes, the models tend to learn that the majority of combinations of the selected parameters would lead to a “Property Damage” prediction.

Preliminary Conclusions

The main objective of this section was to go over the strategies for predicting the Crash Severity based on the Crash Datasets from the state of Illinois based on critical environmental/roadway-related characteristics. After the dataset preparation, which included the cleaning of the entries, as well as a thorough exploratory analysis of the data, the first steps were taken, starting from basic decision tree and random forest models, traditionally used for classification problems. For running any of these models, however, encoding the inputs was necessary, since these structures take numbers as inputs (not strings). The DT and RF models resulted in an accuracy that initially was interpreted as promising, but the limitations of the procedures taken were immediately realized, being the resulting models biased by the dominance of the “property damage” class on the dataset.

When the dataset was sampled to make the number of entries corresponding to every output even, the accuracy decreased, indicating that the dataset was not large enough to train the model with enough information. On the other hand, when the entries corresponding to the most common output were removed, the accuracy for predicting the two least likely outputs increased. Finally, when testing a Convolutional Neural Network, the accuracy of the predictions was once again around 78% considering a batch size of around a third part form the total number of entries, and using the dataset removing “perfect” conditions, accuracy reached 85%. The preliminary results evidence that further techniques may be applied to the original dataset to sample it a more convenient way to optimize the overall accuracy. The results of every model were tested for different parameters to verify the convergence and consistency of the predictions, which could be considered acceptable for a classification problem. However, for the future deliveries, further refinement may help to keep increasing the achieved accuracy, and a final consolidated model will therefore be assessed in terms of validation using datasets from years not considered for the training data.

Discussion

Suggestions for Improvements in Crash Record Methods

When addressing road safety and the technology needs for this area of the transportation engineering, it should be thought about how it is possible to induce drivers to proactively adopt measures for avoiding crashes (reducing speeds, turning on the headlights, opting to travel on a road with less defects, for example). In this work, models were developed to be potentially incorporated into navigation systems (such as Google Maps), in a way that these would give alerts to the driver to proceed with more caution when specific combinations of road surface conditions, road defects presence, lighting conditions, and weather conditions are present. However, because of factors such as the imbalance of data and the low correlation between the features and the labels (among others), the models developed were not able to achieve this in a satisfactory way, ending up predicting "low-severity" crashes in a frequency much higher than what happens in the real data. The limitations observed in the models were also caused by the limitations of the databases itself.

Aiming to improve the development of models like these in the future, more information (i.e., columns) could be added to the features' data frame. Besides capturing information that can potentially be better correlated with the expected outputs, acquiring quantitative data could help on avoid relying on the one-hot encoding for the modeling. As a suggestion, the IDOT dataset could be improved by adding information such as:

1. Maximum speed of the section: quantitative data that surrogates the patterns of speed.
2. Traffic(Average Annual Daily Traffic, AADT): quantitative data that surrogates the freedom of movement within the section.
3. Class of vehicles involved in the crash: when heavy vehicles are involved, the impacts of the crash can potentially be more relevant than the environment conditions and better correlate with the severity of the crash.
4. Passing zone: this true/false feature can potentially capture the occurrence of front-to-front crashes, which tend to be more severe given the summation of the speed's impact.
5. Crossing zone: this true/false feature can potentially capture the occurrence of perpendicular crashes.

In a near future, it is expected that technologies such as the V2V (vehicle-to-vehicle) communication and the autonomous/connected vehicles will become more accessible, and therefore more widespread. In this case, databases such as the one studied in this project (made by IDOT) might also include vehicles and drivers' information. This will be of utmost importance, given that the behavioral patterns are a big source of uncertainties when it comes to the analysis of crash data. For example, drivers that have a more aggressive behavior tend to drive at higher speeds and maintain a smaller gap to the leading vehicle. As for now, the categories that are recorded in the datasets do not capture this. Transportation engineers can eventually find surrogates of this, but still, this is not an easy task.

References

Abdulhafedh, A. (2017) Road Crash Prediction Models: Different Statistical Modeling Approaches. Journal of Transportation Technologies, 7, 190-205. doi: 10.4236/jtts.2017.72014.

Akbar Danesh, Mehrdad Ehsani, Fereidoon Moghadas Nejad & Hamzeh Zakeri (2022) Prediction model of crash severity in imbalanced dataset using data leveling methods and metaheuristic optimization algorithms, International Journal of Crashworthiness, DOI: 10.1080/13588265.2022.2028471

Brunton, S., Kutz, N., (2017) Data Driven Science & Engineering: Machine Learning, Dynamical Systems, and Control.

Chin, H. C.; Quek, S. T. Measurement of Traffic Conflicts. Safety Science, Vol. 26(3), p. 169-185, 1997. DOI: https://doi.org/10.1016/S0925-7535(97)00041-6.

Farmer, C. M. Reliability of Police-Reported Information for Determining Crash and Injury Severity. Traffic Injury Prevention, 2003, n.4, p.38-44, 2003. DOI: https://doi.org/10.1080/15389580309855.

Hauer, E.; Hakkert, A. S. The Extent and Implications of Incomplete Accident Reporting. Transportation Research Record: Journal of the Transportation Research Board, n.1185, p.1-10, 1989.

Illinois Department of Transportation | IDOTAdmin https://gis-idot.opendata.arcgis.com/search?groupIds=6d2862031a6d47c7a8c211e38e423e05

Illinois Department of Transportation | Traffic Crash Report SR 1050 Instruction Manual https://idot.illinois.gov/home/resources/Manuals/Manuals-and-Guides

Lee, C., Hellinga, B., & Saccomanno, F. (2003). Real-Time Crash Prediction Model for Application to Crash Prevention in Freeway Traffic. Transportation Research Record, 1840(1), 67–77. https://doi.org/10.3141/1840-08

Mohammad Hesam Rashidi, Soheil Keshavarz, Parham Pazari, Navid Safahieh, Amir Samimi, Modeling the accuracy of traffic crash prediction models, IATSS Research, Volume 46, Issue 3, 2022, Pages 345-352, ISSN 0386-1112, https://doi.org/10.1016/j.iatssr.2022.03.004.

Myles, A.J., Feudale, R.N., Liu, Y., Woody, N.A. and Brown, S.D. (2004), An introduction to decision tree modeling. J. Chemometrics, 18: 275-285. https://doi.org/10.1002/cem.873

Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine Learning in {P}ython, Journal of Machine Learning Research, Volume 12, pages 2825-2830.

Safarpour, H., Khorasani-Zavareh, D., & Mohammadi, R. (2020). The common road safety approaches: A scoping review and thematic analysis. Chinese journal of traumatology, 23(02), 113-121. https://doi.org/10.1016/j.cjtee.2020.02.005

World Health Organization (WHO). (2022). Road traffic injuries World Health Organization Regional Office for the Eastern Mediterranean. Geneva, Switzerland. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

Yasin, Y. J., Grivna, M., & Abu-Zidan, F. M. (2021). Global impact of COVID-19 pandemic on road traffic collisions. World journal of emergency surgery, 16(1), 1-14.

Yu, R., Han, L., & Zhang, H. (2021). Trajectory data based freeway high-risk events prediction and its influencing factors analyses. Accident Analysis & Prevention, 154, 106085. https://doi.org/10.1016/j.aap.2021.106085.