Predicting Lethality of Car Crashes in New York City Boroughs

Team Members:
Sami Saleh and Aaryan Sumesh

10/21/2024

Table of Contents

Background and Project Goal

In New York City, a police report, also known as an MV104-AN, is required to be filled out for car collisions where someone is injured or killed, or where there is at least $1000 worth of damage. As cities like New York City grow and traffic conditions become more complex, it is essential to have a comprehensive view of the incidents that occur on roads and identify trends and factors that may lead to dangerous outcomes.

Classifying the lethality of car crashes based on various factors such as weather and time of day can allow the city of New York to use preventive measures to stop accidents.

Our project goal will be to classify car crashes in New York City. People can use our model and using the information that the model uses, they can predict whether or not the location is prone to car accidents. People can use this in urban planning, where they can see where to invest money in improving both pedestrian and driver safety.

Dataset Information

** Dataset link: https://catalog.data.gov/dataset/motor-vehicle-collisions-crashes **

Our initial dataset had 29 attributes (28 + class, which was NUMBER OF PERSONS KILLED), with 1,048,575 instances. Each instance represents a car crash that happened in New York City where repair costs were over $1,000. Below is a list of all the attributes present in the original data set with explanations for what they are:

1. CRASH DATE: Date of the car accident in format day/month/year
2. CRASH TIME: Time of car accident in format hour:min on 24 hour clock
3. BOROUGH: If the car accident occurred in a New York borough, that location is given from 5 choices: Brooklyn, Bronx, Manhattan, Queens, Staten Island
4. ZIP CODE: 5 digit area code where car accident occurred
5. LATITUDE: Latitude coordinate of car accident location
6. LONGITUDE: Longitude coordinate of car accident location
7. LOCATION: Location of incident in form (Latitude, Longitude)
8. ON STREET NAME: Name of street where car accident occurred
9. CROSS STREET NAME: Name of street that crosses the on street at the accident
10. OFF STREET NAME: Building address outside where accident occurred
11. NUMBER OF PERSONS INJURED: Total persons injured in car accident
12. NUMBER OF PERSONS KILLED: Total persons killed in car accident *Class Attribute
13. NUMBER OF PEDESTRIANS INJURED: Total pedestrians (people walking on street/sidewalk) injured in car accident
14. NUMBER OF PEDESTRIANS KILLED: Total pedestrians (people walking on street/sidewalk) killed in car accident
15. NUMBER OF CYCLIST INJURED: Total amount of people injured who were riding bikes at the time of the car accident
16. NUMBER OF CYCLIST KILLED: Total amount of people killed who were riding bikes at the time of the car accident
17. NUMBER OF MOTORIST INJURED: Total amount of people injured who were in or driving cars at the time of the car accident
18. NUMBER OF MOTORIST KILLED: Total amount of people killed who were in or driving cars at the time of the car accident
19. CONTRIBUTING FACTOR VEHICLE 1: A label of a possible reason that the first driver was involved in the car accident
20. CONTRIBUTING FACTOR VEHICLE 2: A label of a possible reason that the second driver was involved in the car accident (if there were 2 cars involved in the accident)
21. CONTRIBUTING FACTOR VEHICLE 3: A label of a possible reason that the third driver was involved in the car accident (if there were 3 cars involved in the accident)

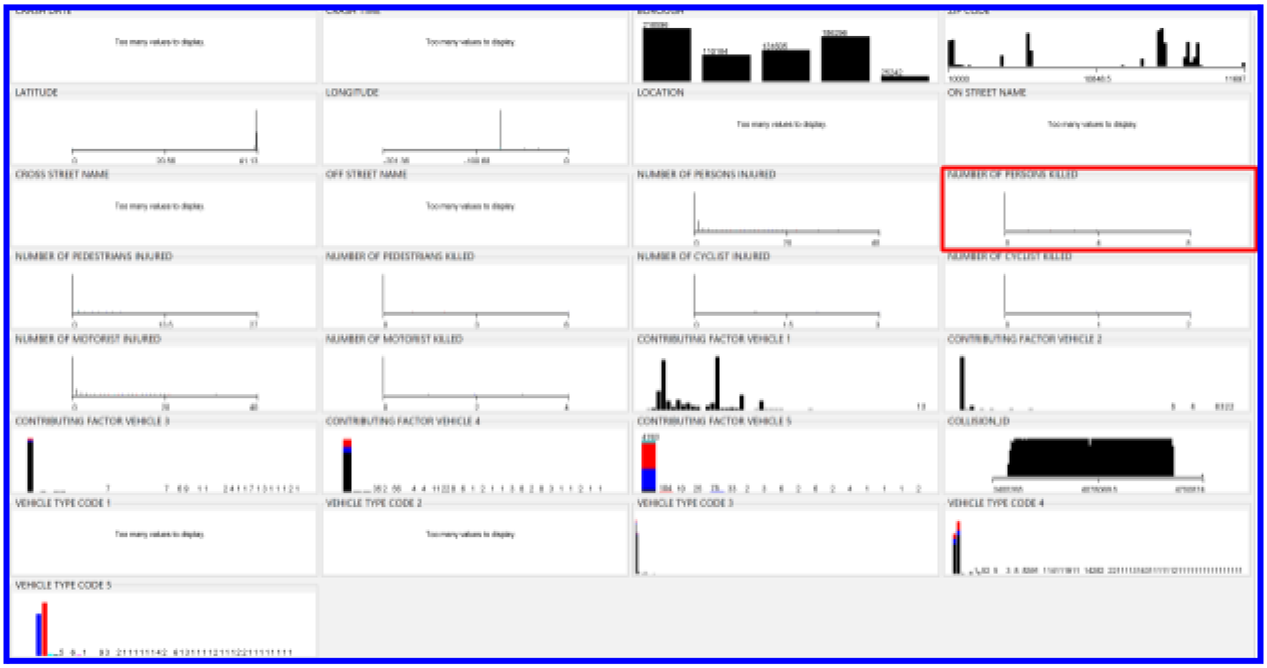
22. CONTRIBUTING FACTOR VEHICLE 4: A label of a possible reason that the fourth driver was involved in the car accident (if there were 4 cars involved in the car accident)
23. CONTRIBUTING FACTOR VEHICLE 5: A label of a possible reason that the fifth driver was involved in the car accident (if there were 5 cars involved in the car accident)
24. COLLISION_ID: Unique ID label for each instance of an accident
25. VEHICLE TYPE CODE 1: A label for the form of transportation of the first vehicle
26. VEHICLE TYPE CODE 2: A label for the form of transportation of the second vehicle
27. VEHICLE TYPE CODE 3: A label for the form of transportation of the third vehicle
28. VEHICLE TYPE CODE 4: A label for the form of transportation of the fourth vehicle
29. VEHICLE TYPE CODE 5: A label for the form of transportation of the fifth vehicle

Below is a list of each attribute with its corresponding number of missing values:

1. CRASH DATE - 0
2. CRASH TIME - 0
3. BOROUGH - 376652
4. ZIP CODE - 376836
5. LATITUDE - 75283
6. LONGITUDE - 75283
7. LOCATION - 75283
8. ON STREET NAME - 257041
9. CROSS STREET NAME - 543032
10. OFF STREET NAME - 792862
11. NUMBER OF PERSONS INJURED - 17
12. NUMBER OF PERSONS KILLED - 30
13. NUMBER OF PEDESTRIANS INJURED - 0
14. NUMBER OF PEDESTRIANS KILLED - 0
15. NUMBER OF CYCLIST INJURED - 0
16. NUMBER OF CYCLIST KILLED - 0
17. NUMBER OF MOTORIST INJURED - 0
18. NUMBER OF MOTORIST KILLED - 0
19. CONTRIBUTING FACTOR VEHICLE 1 - 3757
20. CONTRIBUTING FACTOR VEHICLE 2 - 178615
21. CONTRIBUTING FACTOR VEHICLE 3 - 970983
22. CONTRIBUTING FACTOR VEHICLE 4 - 1030528
23. CONTRIBUTING FACTOR VEHICLE 5 - 1043535
24. COLLISION_ID - 0
25. VEHICLE TYPE CODE 1 - 8659
26. VEHICLE TYPE CODE 2 - 249127
27. VEHICLE TYPE CODE 3 - 975403

28. VEHICLE TYPE CODE 4 - 1031430
29. VEHICLE TYPE CODE 5 - 1043737

Below are graphs showing the distribution of each attribute in our dataset. The main attribute we want to take a look at is the class attribute, or number of people killed.



As mentioned, we are particularly interested in the “NUMBER OF PEOPLE KILLED” attribute from the dataset, which is boxed in red above. We can see that the data appears to only be a singular value of zero. However, this is only because the amount of non lethal, or zero death, car crashes was much, much greater than the number of fatal car crashes. In fact, we see in Weka that while the maximum value for number of people killed was 8, the mean was 0.001, with a standard deviation of 0.041 (see below). This would mean that only on very rare occasions are car crashes fatal in New York City. The most important takeaway from this is that the attributes for number of persons, cyclists, pedestrians, and motorists killed are all right skewed.

Statistic	Value
Minimum	0
Maximum	8
Mean	0.001
StdDev	0.041

Data Preprocessing

Opening the data in Weka

1. Open Weka explorer, and load in the dataset. The dataset is located here: original.csv.

Set the Number of Persons Killed as the Class Variable

2. Select the edit button, which is located near the top right corner. A new window should pop up displaying the data inside our dataset.
3. In this new window, scroll right until you find attribute number 12: NUMBER OF PERSONS KILLED.
4. Right click the cell which says "12: NUMBER OF PERSONS KILLED" and select "attribute as class." What we have done here is set the number of persons killed as our class, which is what we are trying to predict in our model. Press Ok.

Remove all instances where the class attribute value is missing

5. Now, what we will do is remove all instances where the class attribute value is missing. This is justified because if the class value is missing, there is no point in that data because we would not be able to compare the answer that we predicted to an actual value. In addition, our model would not be able to learn anything if the class value is missing, so deleting these instances is justified.
6. To do this, select Choose > Filters > Unsupervised > Instance > Remove With Values. Press on the bar on the right of the choose button, and a new menu should pop up. In this menu, change attributeIndex to 29 (our class attribute is number 29), change matchMissingValues to True, then press ok > apply. You should see now that there are no missing values in the class attribute.

Selected attribute

Name: NUMBER OF PERSONS KILLED Type: Numeric
Missing: 0 (0%) Distinct: 6 Unique: 1 (0%)

Statistic	Value
Minimum	0
Maximum	8
Mean	0.001
StdDev	0.041

Remove unnecessary attributes

7. There is one unnecessary attribute that we found. We found that Collision_id is an unnecessary attribute because we will not be using Collision_id to predict fatality in car crashes. Collision_id is completely irrelevant to our goal. Thus, we can remove that attribute.
8. To remove the attribute, select the checkbox for attribute number 23, which is COLLISION_ID. Once you see a blue check mark, press the remove button. You should now have 28 attributes.

Remove attributes with majority missing values, except for CONTRIBUTING FACTOR VEHICLE 3, 4, and 5

9. Now, we would like to remove attributes which have a majority of missing values. This is because using these attributes wouldn't make much sense because the data would be incomplete or unreliable, leading to biased or inaccurate results in the analysis. Removing such attributes helps improve the quality and accuracy of the model.
10. The attributes that we found that had a majority of missing values were Cross street name, Off street name, Contributing factor vehicle 3, 4, and 5, Vehicle Type code 3, 4, and 5. We can go ahead and delete Cross street name, Off street name, Vehicle Type code 3, 4, and 5. To do this, follow the same steps in step 8, except select the aforementioned attributes.
11. The reason why we didn't want to delete the Contributing factor vehicle 3, 4, and 5 was because these attributes give us information into the number of vehicles that were involved in the car crash. For instance, if all values for contributing factor vehicle 3, 4, and 5 are missing for a particular instance, that tells us that less than 3 cars were involved in the car accident. If one of these values was not missing, we know that there were more than 2 cars involved. We decided to take this into account and "merge" these three columns into a singular column called "more than two vehicles involved." We will get into how we did this later. (See step 18)

Remove redundant attributes: Number of pedestrians killed, number of motorists killed, number of cyclists killed, location

12. When examining the data, we see that there are attributes for number of pedestrians killed, number of motorists killed, and number of cyclists killed. These all are redundant to the class variable, and can be deleted without any loss of information. Thus, we will delete these attributes. To delete these attributes, refer to step 8, except select the aforementioned attributes.
13. Another redundant attribute is location. Notice that we have two other attributes for longitude and latitude. When examining the contents of the location attribute, we see that the data is in the form (latitude, longitude). Thus, we can remove location without the loss of any information. To delete this attribute, refer to step 8, except select location.

Alter Crash Date to Season

14. Another thing that we did was to change the Crash Date attribute into Season. While both are discrete variables, Season only can take 4 values while crash date can take many more. We chose to simplify crash date into season so our model could potentially take into account conditions of the road. For example, we know in winter it is more likely to have ice on the roads than in summer, but just using dates would make the model have a tougher time realizing this.
15. To do this, we had to use a python script. First, click save, then browse to your desired location. Then, change the file type to .csv and save the file using an appropriate name. I named this file checkpoint1.csv. Navigate to this folder in your computer. Here, create a python file. I am calling this file changes1.py. Here are the contents of the file. I will explain what each function does when I reach those steps. Run this file. You should see several new files show up.

```
# import
import pandas as pd
import csv

# removes troublesome commas and replaces them with semicolons
def fix_csv(input_file, output_file):
    with open(input_file, 'r', newline='', encoding='utf-8') as infile,
open(output_file, 'w', newline='', encoding='utf-8') as outfile:
        reader = csv.reader(infile)
        writer = csv.writer(outfile)

        for i, row in enumerate(reader):
            if len(row) != 19:
                continue
            new_row = [item.replace(',', ';') for item in row]
            writer.writerow(new_row)

fix_csv('checkpoint1.csv', 'checkpoint1_fixed.csv')

# read the fixed csv file
df = pd.read_csv('checkpoint1_fixed.csv')
df.columns = df.columns.str.replace("'", "").str.replace('"', '')

# convert CRASH DATE and CRASH TIME to date time format
df['CRASH DATE'] = pd.to_datetime(df['CRASH DATE'])
df['CRASH TIME'] = pd.to_datetime(df['CRASH TIME'], format='%H:%M')
```

```
# function to get the season given a month number
def get_season(month):
    if month in [12, 1, 2]:
        return 'Winter'
    elif month in [3, 4, 5]:
        return 'Spring'
    elif month in [6, 7, 8]:
        return 'Summer'
    else:
        return 'Fall'

# function to determine if the given hour is in a rush hour
def is_rush_hour(hour):
    if (6 <= hour <= 9) or (16 <= hour <= 19):
        return 'Yes'
    else:
        return 'No'

# function to determine if the time of day given an hour is Day or Night
def time_of_day(hour):
    if 6 <= hour < 18:
        return 'Day'
    else:
        return 'Night'

# function to determine if the car crash consisted of more than two
vehicles
def more_than_two_vehicles(row):
    factors = [
        row['CONTRIBUTING FACTOR VEHICLE 1'],
        row['CONTRIBUTING FACTOR VEHICLE 2'],
        row['CONTRIBUTING FACTOR VEHICLE 3'],
        row['CONTRIBUTING FACTOR VEHICLE 4'],
        row['CONTRIBUTING FACTOR VEHICLE 5']
    ]
    if sum(factor != '?' for factor in factors) > 2:
        return 'Yes'
    else:
        return 'No'
```

```
# apply changes to the dataframe
df['SEASON'] = df['CRASH DATE'].dt.month.apply(get_season)
df['RUSH HOUR'] = df['CRASH TIME'].dt.hour.apply(is_rush_hour)
df['TIME OF DAY'] = df['CRASH TIME'].dt.hour.apply(time_of_day)
df['MORE THAN 2 VEHICLES INVOLVED'] = df.apply(more_than_two_vehicles,
axis=1)

# delete the not needed columns
df = df.drop(columns=['CRASH DATE', 'CRASH TIME', 'CONTRIBUTING FACTOR
VEHICLE 3', 'CONTRIBUTING FACTOR VEHICLE 4', 'CONTRIBUTING FACTOR VEHICLE
5'])

# save to csv
df.to_csv('checkpoint1_modified.csv', index=False)

# done
print("done")
```

16. Here, the relevant function is get_season. We just take in a number which is month and output a string, which is either Winter, Spring, Summer, or Fall.

```
def get_season(month):
    if month in [12, 1, 2]:
        return 'Winter'
    elif month in [3, 4, 5]:
        return 'Spring'
    elif month in [6, 7, 8]:
        return 'Summer'
    else:
        return 'Fall'
```

Alter Crash Time to Rush Hour and Time of Day

17. This is also part of the python script. The relevant functions here are is_rush_hour and time_of_day. These are two simple functions that return if the hour given is in a rush hour or if the time of day is night or day. These variables again make it easier for the model to use these seemingly important values.

```
def is_rush_hour(hour):
    if (6 <= hour <= 9) or (16 <= hour <= 19):
        return 'Yes'
    else:
        return 'No'
```

```python
def time_of_day(hour):
    if 6 <= hour < 18:
        return 'Day'
    else:
        return 'Night'
```

Alter contributing factors 3, 4, and 5 to More than two vehicles involved

18. This is also part of the python script. The relevant function here is more_than_two_vehicles, which takes a row and outputs a string which is either Yes or No depending on if more than two vehicles were involved in the car crash.

```python
def more_than_two_vehicles(row):
    factors = [
        row['CONTRIBUTING FACTOR VEHICLE 1'],
        row['CONTRIBUTING FACTOR VEHICLE 2'],
        row['CONTRIBUTING FACTOR VEHICLE 3'],
        row['CONTRIBUTING FACTOR VEHICLE 4'],
        row['CONTRIBUTING FACTOR VEHICLE 5']
    ]
    if sum(factor != '?' for factor in factors) > 2:
        return 'Yes'
    else:
        return 'No'
```

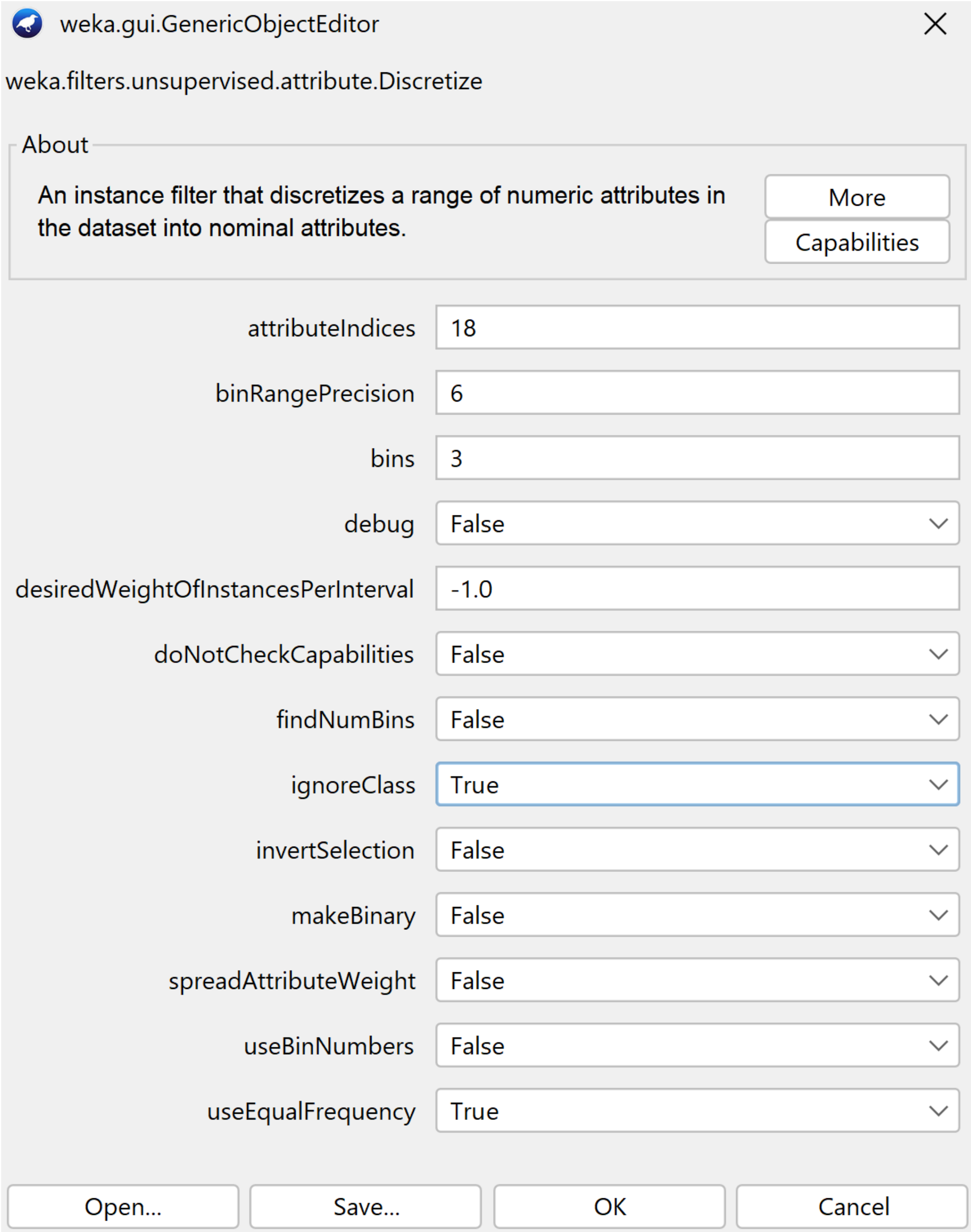
19. Open checkpoint1_modified.csv in Weka. You should now only have 18 attributes. You may have to make the number of persons killed the class again. Follow the aforementioned steps to do so.

Remove missing values

20. We can easily remove missing values using Weka. To do this, we go to the ReplaceMissingValues filter, and press apply.

Bin the class variable

21. To bin the class variable, we will use the Discretize filter in Weka. Choose the discretize filter in Weka, and use the following menu options. Then press ok > apply.



22. You should see the following:

Selected attribute

Name: NUMBER OF PERSONS KILLED — Type: Nominal

Missing: 0 (0%) — Distinct: 3 — Unique: 0 (0%)

No.	Label	Count	Weight
1	'(-inf-0.5]'	1047040	1047040
2	'(0.5-1.5]'	1451	1451
3	'(1.5-inf)'	46	46

Replace bin names with better bin names (non_lethal, somewhat_lethal, very_lethal)

23. However, these bins have not so good names. We would like to change this to being non_lethal, somewhat_lethal, and very_lethal. To do this, we created a python script. First, save this file in the same directory as before, and name appropriately. I will call my file almost_done.csv.
24. Create a python script in the same directory to rename the bins. I called this file changes2.py. Here is the code. Then, run the file. Then, go back to Weka and

```
# import
import pandas as pd

# load df
df = pd.read_csv('almost_done.csv')

# replace bin names with more appropriate bin names
df["'NUMBER OF PERSONS KILLED'"] = df["'NUMBER OF PERSONS
KILLED'"].replace({
    "'\\'(-inf-0.5]\\''": 'non_lethal',
    "'\\'(0.5-1.5]\\''": 'somewhat_lethal',
    "'\\'(1.5-inf)\\''": 'very_lethal'
})

# save csv
df.to_csv('done.csv', index=False)
print(df.head())
print("done")
```

Go back into Weka, and open done.csv.

Train test split

25. Despite the name, we are not done yet. We still need to perform the train test split.
26. To do the train test split, we used RemovePercentage. In the RemovePercentage menu, we changed the percentage to 70%, meaning that 70% of the data would be removed. We then clicked apply, and saved this file as "testing.csv." We reopened the done.csv file and in the menu for RemovePercentage, we selected invertSelection. This would remove the other 30% of the data while keeping the 70% of the data that was deleted to form the testing data. We then saved this file as "testing.csv."
27. Using RemovePercentage is good for train test split because Weka maintains the ratio between the number of each class label, which is crucial for train test splitting.

Preprocessing done!

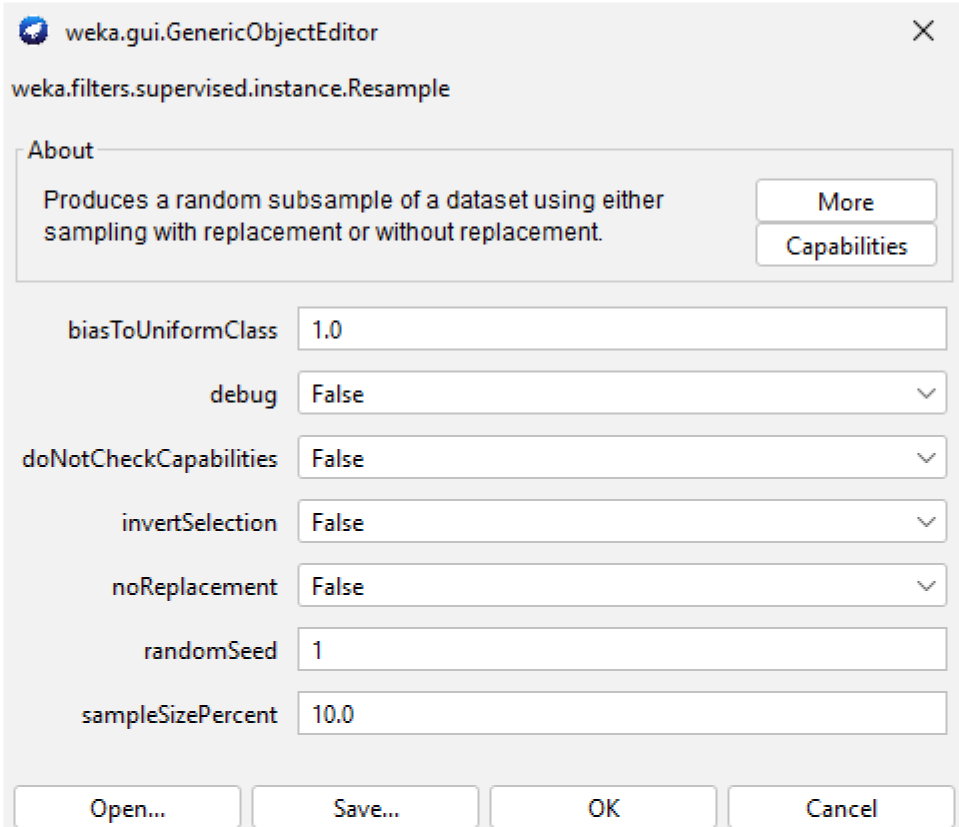
Attribute Selection Algorithms and Model Classifiers

The raw dataset after preprocessing has 17 attributes:

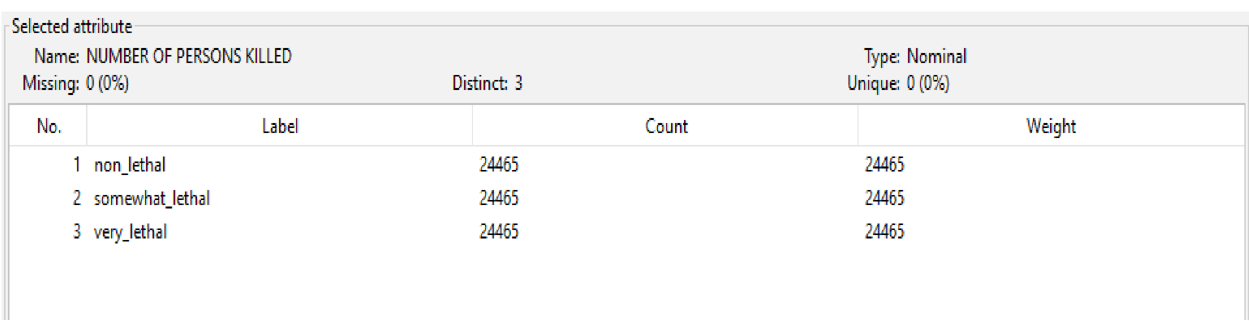
1. BOROUGH: The specific New York City borough where the crash took place.
2. ZIP CODE: The postal code identifying the location of the crash.
3. LATITUDE: The geographical north-south coordinate of the crash location.
4. LONGITUDE: The geographical east-west coordinate of the crash location.
5. ON STREET NAME: The name of the main street where the crash took place.
6. NUMBER OF PERSONS INJURED: The total count of individuals injured in the crash.
7. NUMBER OF PEDESTRIANS INJURED: The number of pedestrians injured in the crash.
8. NUMBER OF CYCLIST INJURED: The number of cyclists injured as a result of the crash.
9. NUMBER OF MOTORIST INJURED: The number of drivers or passengers in vehicles injured in the crash.
10. CONTRIBUTING FACTOR VEHICLE 1: The primary cause attributed to the first (most damaged) vehicle in the car crash.
11. CONTRIBUTING FACTOR VEHICLE 2: The primary cause attributed to the second (second most damaged) vehicle in the car crash.
12. VEHICLE TYPE CODE 1: The type or classification of the first vehicle involved in the crash (car, truck, etc.).
13. VEHICLE TYPE CODE 2: The type or classification of the second vehicle involved in the crash (car, truck, etc.).
14. SEASON: The season during which the crash occurred (summer, spring, winter, or fall).
15. RUSH HOUR: Indicates whether or not the crash took place during peak traffic hours.
16. TIME OF DAY: The general time when the crash occurred (Day or Night).
17. MORE THAN 2 VEHICLES INVOLVED: A binary indicator of whether or not more than two vehicles were involved in the crash.

The class attribute is ‘NUMBER OF PERSONS KILLED’ and there are three values for this class label: non_lethal, somewhat_lethal, and very_lethal

Due to the skewed distribution in the class attribute, with 732807 instances being classified as “non_lethal”, 1131 instances being classified as “somewhat_lethal”, and 38 instances being classified as “very_lethal”, we decided to take a stratified sample of the training data for attribute selection. To do this, we used the WEKA **Resample** filter with a sample size percent of 10% which works better with larger datasets. This is because large samples are typically expensive, and since we have such a large dataset, we can compensate with a smaller sample size while still capturing patterns accurately.



This leaves us with a stratified sample of the data, with each class label having 24465 instances:



Selected attribute

Name: NUMBER OF PERSONS KILLED   Type: Nominal

Missing: 0 (0%)   Distinct: 3   Unique: 0 (0%)

No.	Label	Count	Weight
1	non_lethal	24465	24465
2	somewhat_lethal	24465	24465
3	very_lethal	24465	24465

This dataset will be used for attribute selection, reducing the chance of bias affecting the selected attributes. Initially, the dataset was highly skewed towards the “non_lethal” category, but after this crucial step, we get a stratified sample that represents each class equally.

Attribute Selection Algorithm One: **GainRatioAttributeEval**

```
Attribute Evaluator (supervised, Class (nominal): 18 NUMBER OF PERSONS KILLED):
        Gain Ratio feature evaluator

Ranked attributes:
 0.14422   3 LATITUDE
 0.142     4 LONGITUDE
 0.13253   7 NUMBER OF PEDESTRIANS INJURED
 0.12548   8 NUMBER OF CYCLIST INJURED
 0.12393  10 CONTRIBUTING FACTOR VEHICLE 1
 0.11482   5 ON STREET NAME
 0.10769  11 CONTRIBUTING FACTOR VEHICLE 2
 0.09785   9 NUMBER OF MOTORIST INJURED
 0.09124  17 MORE THAN 2 VEHICLES INVOLVED
 0.08408   2 ZIP CODE
 0.08383   6 NUMBER OF PERSONS INJURED
 0.08249  16 TIME OF DAY
 0.07398  13 VEHICLE TYPE CODE 2
 0.06888  12 VEHICLE TYPE CODE 1
 0.06303  15 RUSH HOUR
 0.01096   1 BOROUGH
 0.00684  14 SEASON

Selected attributes: 3,4,7,8,10,5,11,9,17,2,6,16,13,12,15,1,14 : 17
```

Attributes were selected that had an information gain value greater than or equal to **0.1.** Therefore, we can remove the following attributes: SEASON, BOROUGH, RUSH HOUR, VEHICLE TYPE CODE 1, VEHICLE TYPE CODE 2, TIME OF DAY, NUMBER OF PERSONS INJURED, ZIP CODE, MORE THAN 2 VEHICLES INVOLVED, NUMBER OF MOTORIST INJURED.

Attribute Selection Algorithm Two: **InfoGainAttributeEval**

```
Attribute Evaluator (supervised, Class (nominal): 18 NUMBER OF PERSONS KILLED):
        Information Gain Ranking Filter

Ranked attributes:
 1.1871    4 LONGITUDE
 1.1859    3 LATITUDE
 0.9628    5 ON STREET NAME
 0.45     10 CONTRIBUTING FACTOR VEHICLE 1
 0.3458    2 ZIP CODE
 0.1734   12 VEHICLE TYPE CODE 1
 0.1454    9 NUMBER OF MOTORIST INJURED
 0.1378   13 VEHICLE TYPE CODE 2
 0.1298    6 NUMBER OF PERSONS INJURED
 0.086    11 CONTRIBUTING FACTOR VEHICLE 2
 0.0815   16 TIME OF DAY
 0.0625   17 MORE THAN 2 VEHICLES INVOLVED
 0.0547   15 RUSH HOUR
 0.0445    7 NUMBER OF PEDESTRIANS INJURED
 0.0173    1 BOROUGH
 0.0136   14 SEASON
 0.0127    8 NUMBER OF CYCLIST INJURED

Selected attributes: 4,3,5,10,2,12,9,13,6,11,16,17,15,7,1,14,8 : 17
```

Attributes were selected that had an information gain greater than or equal to **0.1.** Therefore, we can remove the following attributes: NUMBER OF CYCLIST INJURED, SEASON, BOROUGH, NUMBER OF PEDESTRIANS INJURED, RUSH HOUR, MORE THAN 2 VEHICLES INVOLVED, TIME OF DAY, CONTRIBUTING FACTOR VEHICLE 2.

Attribute Selection Algorithm Three: **CfsSubsetEval with GreedyStepwise Search Method**

```
Search Method:
        Greedy Stepwise (forwards).
        Start set: no attributes
        Merit of best subset found:    0.272

Attribute Subset Evaluator (supervised, Class (nominal): 18 NUMBER OF PERSONS KILLED):
        CFS Subset Evaluator
        Including locally predictive attributes

Selected attributes: 3,4,8,10 : 4
                     LATITUDE
                     LONGITUDE
                     NUMBER OF CYCLIST INJURED
                     CONTRIBUTING FACTOR VEHICLE 1
```

Using a GreedyStepwise search method, the attribute selection algorithm recommended to keep the following attributes: LATITUDE, LONGITUDE, NUMBER OF CYCLIST INJURED, CONTRIBUTING FACTOR VEHICLE 1.

The following attributes would be removed: NUMBER OF MOTORIST INJURED, TIME OF DAY, ON STREET NAME, CONTRIBUTING FACTOR VEHICLE 2, NUMBER OF PEDESTRIANS INJURED, VEHICLE TYPE CODE 2, MORE THAN 2 VEHICLES INVOLVED, VEHICLE TYPE CODE 1, RUSH HOUR, ZIP CODE, NUMBER OF PERSONS INJURED, SEASON, BOROUGH.

Attribute Selection Algorithm Four: **OneRAttributeEval**

```
Attribute Evaluator (supervised, Class (nominal): 18 NUMBER OF PERSONS KILLED):
        OneR feature evaluator.

        Using 10 fold cross validation for evaluating attributes.
        Minimum bucket size for OneR: 6

Ranked attributes:
90.115   3 LATITUDE
89.678   4 LONGITUDE
81.76    5 ON STREET NAME
62.499  10 CONTRIBUTING FACTOR VEHICLE 1
54.769   2 ZIP CODE
48.105  12 VEHICLE TYPE CODE 1
46.936   9 NUMBER OF MOTORIST INJURED
46.738  16 TIME OF DAY
44.134   6 NUMBER OF PERSONS INJURED
42.868  15 RUSH HOUR
42.47   13 VEHICLE TYPE CODE 2
42.206  17 MORE THAN 2 VEHICLES INVOLVED
39.552  11 CONTRIBUTING FACTOR VEHICLE 2
38.482   1 BOROUGH
38.365  14 SEASON
35.752   7 NUMBER OF PEDESTRIANS INJURED
34.417   8 NUMBER OF CYCLIST INJURED

Selected attributes: 3,4,5,10,2,12,9,16,6,15,13,17,11,1,14,7,8 : 17
```

Attributes were selected that had a OneR score greater than or equal to **40%**. Therefore, we can remove the following attributes: NUMBER OF CYCLIST INJURED, NUMBER OF PEDESTRIANS INJURED, SEASON, BOROUGH, CONTRIBUTING FACTOR VEHICLE 2.

Attribute Selection Algorithm Five: **Non-WEKA Approach**

Looking at the previous four attribute selection algorithms, the following attributes were recommended to be removed by all four algorithms:

SEASON
BOROUGH

It is reasonable to assume that both attributes will not have patterns with lethality of car crashes, as there are a relatively equal number of instances where car crashes occur for each label in either attribute. Therefore, for our 5th attribute selection, we will remove just these two attributes.

Classifier Models

1. **J48 -** A decision tree algorithm that recursively partitions the dataset by selecting the attribute that provides the highest information gain at each node, making it efficient for handling both categorical and numerical data.

2. **NaiveBayes -** A probabilistic model based on Bayes' Theorem, which assumes independence between features, and calculates the likelihood of each class by multiplying the probabilities of individual attributes, making it efficient for large datasets such as ours.

3. **OneR -** A rule based algorithm that creates a rule for each attribute by dividing the data into categories and selecting the rule that results in the lowest error.

4. **DecisionTable -** A rule based algorithm that creates a decision table by evaluating a subset of attributes and generating a set of rules based on combinations of these attributes.

We also stratified the testing data, to ensure that the minority class labels (somewhat_lethal and non_lethal) are not ignored and are represented in the testing data to mitigate bias.

Results

GainRatioAttributeEval

GainRatioAttributeEval with J48

```
Correctly Classified Instances       29414               93.5114 %
Incorrectly Classified Instances      2041                6.4886 %
Kappa statistic                          0.9027
Mean absolute error                      0.0455
Root mean squared error                  0.1615
Relative absolute error                 10.2483 %
Root relative squared error             34.2615 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.840    0.017    0.961      0.840    0.896      0.853    0.985     0.977     non_lethal
                 0.966    0.075    0.865      0.966    0.913      0.868    0.990     0.973     somewhat_lethal
                 1.000    0.005    0.990      1.000    0.995      0.993    1.000     1.000     very_lethal
Weighted Avg.    0.935    0.032    0.939      0.935    0.935      0.905    0.992     0.983

=== Confusion Matrix ===

     a     b     c   <-- classified as
  8803  1579   103 |     a = non_lethal
   359 10126     0 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

GainRatioAttributeEval with NaiveBayes

```
Correctly Classified Instances       21332               67.8175 %
Incorrectly Classified Instances     10123               32.1825 %
Kappa statistic                          0.5173
Mean absolute error                      0.2145
Root mean squared error                  0.3876
Relative absolute error                 48.26   %
Root relative squared error             82.2132 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.712    0.148    0.707      0.712    0.709      0.563    0.889     0.823     non_lethal
                 0.322    0.009    0.948      0.322    0.481      0.466    0.923     0.861     somewhat_lethal
                 1.000    0.326    0.605      1.000    0.754      0.639    0.992     0.964     very_lethal
Weighted Avg.    0.678    0.161    0.753      0.678    0.648      0.556    0.935     0.883

=== Confusion Matrix ===

     a     b     c   <-- classified as
  7467   185  2833 |     a = non_lethal
  3100  3380  4005 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

GainRatioAttributeEval with OneR

```
Correctly Classified Instances       30174               95.9275 %
Incorrectly Classified Instances      1281                4.0725 %
Kappa statistic                          0.9389
Mean absolute error                      0.0271
Root mean squared error                  0.1648
Relative absolute error                  6.1087 %
Root relative squared error             34.9535 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.878    0.000    1.000      0.878    0.935      0.910    0.939     0.919     non_lethal
                 1.000    0.060    0.893      1.000    0.943      0.916    0.970     0.893     somewhat_lethal
                 1.000    0.001    0.998      1.000    0.999      0.998    0.999     0.998     very_lethal
Weighted Avg.    0.959    0.020    0.964      0.959    0.959      0.941    0.969     0.936

=== Confusion Matrix ===

     a     b     c   <-- classified as
  9204  1260    21 |     a = non_lethal
     0 10485     0 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

GainRatioAttributeEval with DecisionTable

```
Correctly Classified Instances       29994               95.3553 %
Incorrectly Classified Instances      1461                4.6447 %
Kappa statistic                          0.9303
Mean absolute error                      0.0614
Root mean squared error                  0.1557
Relative absolute error                 13.8054 %
Root relative squared error             33.0319 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.861    0.000    1.000      0.861    0.925      0.897    0.984     0.976     non_lethal
                 1.000    0.069    0.879      1.000    0.935      0.904    0.984     0.953     somewhat_lethal
                 1.000    0.001    0.999      1.000    0.999      0.999    1.000     0.999     very_lethal
Weighted Avg.    0.954    0.023    0.959      0.954    0.953      0.933    0.989     0.976

=== Confusion Matrix ===

     a     b     c   <-- classified as
  9024  1447    14 |     a = non_lethal
     0 10485     0 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

InfoGainAttributeEval

InfoGainAttributeEval with J48

```
Correctly Classified Instances       29917               95.1105 %
Incorrectly Classified Instances      1538                4.8895 %
Kappa statistic                          0.9267
Mean absolute error                      0.0312
Root mean squared error                  0.138
Relative absolute error                  7.0307 %
Root relative squared error             29.2666 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.856    0.002    0.996      0.856    0.921      0.891    0.989     0.987     non_lethal
                 0.997    0.067    0.882      0.997    0.936      0.905    0.994     0.979     somewhat_lethal
                 1.000    0.005    0.990      1.000    0.995      0.993    1.000     1.000     very_lethal
Weighted Avg.    0.951    0.024    0.956      0.951    0.951      0.930    0.994     0.988

=== Confusion Matrix ===

     a     b     c   <-- classified as
  8979  1404   102 |     a = non_lethal
    32 10453     0 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

InfoGainAttributeEval with NaiveBayes

```
Correctly Classified Instances       25143               79.9332 %
Incorrectly Classified Instances      6312               20.0668 %
Kappa statistic                          0.699
Mean absolute error                      0.1481
Root mean squared error                  0.3136
Relative absolute error                 33.327  %
Root relative squared error             66.5352 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.908    0.204    0.690      0.908    0.784      0.669    0.935     0.891     non_lethal
                 0.490    0.011    0.956      0.490    0.648      0.600    0.946     0.904     somewhat_lethal
                 1.000    0.086    0.854      1.000    0.921      0.883    0.993     0.979     very_lethal
Weighted Avg.    0.799    0.100    0.833      0.799    0.784      0.717    0.958     0.925

=== Confusion Matrix ===

     a     b     c   <-- classified as
  9519   235   731 |     a = non_lethal
  4279  5139  1067 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

InfoGainAttributeEval with OneR

```
Correctly Classified Instances       30174               95.9275 %
Incorrectly Classified Instances      1281                4.0725 %
Kappa statistic                          0.9389
Mean absolute error                      0.0271
Root mean squared error                  0.1648
Relative absolute error                  6.1087 %
Root relative squared error             34.9535 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.878    0.000    1.000      0.878    0.935      0.910    0.939     0.919     non_lethal
                 1.000    0.060    0.893      1.000    0.943      0.916    0.970     0.893     somewhat_lethal
                 1.000    0.001    0.998      1.000    0.999      0.998    0.999     0.998     very_lethal
Weighted Avg.    0.959    0.020    0.964      0.959    0.959      0.941    0.969     0.936

=== Confusion Matrix ===

     a     b     c   <-- classified as
  9204  1260    21 |     a = non_lethal
     0 10485     0 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

InfoGainAttributeEval with DecisionTable

```
Correctly Classified Instances       29994               95.3553 %
Incorrectly Classified Instances      1461                4.6447 %
Kappa statistic                          0.9303
Mean absolute error                      0.0614
Root mean squared error                  0.1557
Relative absolute error                 13.8054 %
Root relative squared error             33.0319 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.861    0.000    1.000      0.861    0.925      0.897    0.984     0.976     non_lethal
                 1.000    0.069    0.879      1.000    0.935      0.904    0.984     0.953     somewhat_lethal
                 1.000    0.001    0.999      1.000    0.999      0.999    1.000     0.999     very_lethal
Weighted Avg.    0.954    0.023    0.959      0.954    0.953      0.933    0.989     0.976

=== Confusion Matrix ===

     a     b     c   <-- classified as
  9024  1447    14 |     a = non_lethal
     0 10485     0 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

CfsSubsetEval with GreedyStepwise Search Method

CfsSubsetEval with J48

```
Correctly Classified Instances       30596               97.2691 %
Incorrectly Classified Instances       859                2.7309 %
Kappa statistic                          0.959
Mean absolute error                      0.025
Root mean squared error                  0.1246
Relative absolute error                  5.614  %
Root relative squared error             26.433  %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.920    0.001    0.998      0.920    0.957      0.939    0.982     0.981     non_lethal
                 0.998    0.039    0.927      0.998    0.961      0.943    0.989     0.962     somewhat_lethal
                 1.000    0.001    0.998      1.000    0.999      0.999    1.000     0.999     very_lethal
Weighted Avg.    0.973    0.014    0.974      0.973    0.973      0.960    0.990     0.980

=== Confusion Matrix ===

     a     b     c   <-- classified as
  9646   819    20 |     a = non_lethal
    20 10465     0 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

CfsSubsetEval with NaiveBayes

```
Correctly Classified Instances       15669               49.814  %
Incorrectly Classified Instances     15786               50.186  %
Kappa statistic                          0.2472
Mean absolute error                      0.3447
Root mean squared error                  0.5415
Relative absolute error                 77.5494 %
Root relative squared error            114.8729 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.432    0.110    0.662      0.432    0.523      0.368    0.760     0.640     non_lethal
                 0.062    0.003    0.921      0.062    0.117      0.189    0.690     0.582     somewhat_lethal
                 1.000    0.640    0.439      1.000    0.610      0.397    0.867     0.757     very_lethal
Weighted Avg.    0.498    0.251    0.674      0.498    0.416      0.318    0.772     0.660

=== Confusion Matrix ===

     a     b     c   <-- classified as
  4531    56  5898 |     a = non_lethal
  2310   653  7522 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

CfsSubsetEval with OneR

```
Correctly Classified Instances       30174               95.9275 %
Incorrectly Classified Instances      1281                4.0725 %
Kappa statistic                          0.9389
Mean absolute error                      0.0271
Root mean squared error                  0.1648
Relative absolute error                  6.1087 %
Root relative squared error             34.9535 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.878    0.000    1.000      0.878    0.935      0.910    0.939     0.919     non_lethal
                 1.000    0.060    0.893      1.000    0.943      0.916    0.970     0.893     somewhat_lethal
                 1.000    0.001    0.998      1.000    0.999      0.998    0.999     0.998     very_lethal
Weighted Avg.    0.959    0.020    0.964      0.959    0.959      0.941    0.969     0.936

=== Confusion Matrix ===

     a     b     c   <-- classified as
  9204  1260    21 |     a = non_lethal
     0 10485     0 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

CfsSubsetEval with DecisionTable

```
Correctly Classified Instances       29994               95.3553 %
Incorrectly Classified Instances      1461                4.6447 %
Kappa statistic                          0.9303
Mean absolute error                      0.0614
Root mean squared error                  0.1557
Relative absolute error                 13.8054 %
Root relative squared error             33.0319 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.861    0.000    1.000      0.861    0.925      0.897    0.984     0.976     non_lethal
                 1.000    0.069    0.879      1.000    0.935      0.904    0.984     0.953     somewhat_lethal
                 1.000    0.001    0.999      1.000    0.999      0.999    1.000     0.999     very_lethal
Weighted Avg.    0.954    0.023    0.959      0.954    0.953      0.933    0.989     0.976

=== Confusion Matrix ===

     a     b     c   <-- classified as
  9024  1447    14 |     a = non_lethal
     0 10485     0 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

OneRAttributeEval

OneRAttributeEval with J48

```
Correctly Classified Instances       30360               96.5188 %
Incorrectly Classified Instances      1095                3.4812 %
Kappa statistic                          0.9478
Mean absolute error                      0.0288
Root mean squared error                  0.1322
Relative absolute error                  6.4713 %
Root relative squared error             28.0387 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.899    0.002    0.996      0.899    0.945      0.922    0.988     0.987     non_lethal
                 0.997    0.049    0.911      0.997    0.952      0.928    0.994     0.978     somewhat_lethal
                 1.000    0.002    0.996      1.000    0.998      0.997    1.000     1.000     very_lethal
Weighted Avg.    0.965    0.017    0.968      0.965    0.965      0.949    0.994     0.988

=== Confusion Matrix ===

     a     b     c   <-- classified as
  9426  1020    39 |     a = non_lethal
    36 10449     0 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

OneRAttributeEval with NaiveBayes

```
Correctly Classified Instances       25677               81.6309 %
Incorrectly Classified Instances      5778               18.3691 %
Kappa statistic                          0.7245
Mean absolute error                      0.1398
Root mean squared error                  0.3064
Relative absolute error                 31.4481 %
Root relative squared error             64.9907 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.921    0.186    0.712      0.921    0.803      0.699    0.943     0.905     non_lethal
                 0.528    0.013    0.954      0.528    0.680      0.626    0.945     0.904     somewhat_lethal
                 1.000    0.076    0.867      1.000    0.929      0.895    0.993     0.980     very_lethal
Weighted Avg.    0.816    0.092    0.844      0.816    0.804      0.740    0.960     0.930

=== Confusion Matrix ===

     a     b     c   <-- classified as
  9653   268   564 |     a = non_lethal
  3906  5539  1040 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

OneRAttributeEval with OneR

```
Correctly Classified Instances       30174               95.9275 %
Incorrectly Classified Instances      1281                4.0725 %
Kappa statistic                          0.9389
Mean absolute error                      0.0271
Root mean squared error                  0.1648
Relative absolute error                  6.1087 %
Root relative squared error             34.9535 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.878    0.000    1.000      0.878    0.935      0.910    0.939     0.919     non_lethal
                 1.000    0.060    0.893      1.000    0.943      0.916    0.970     0.893     somewhat_lethal
                 1.000    0.001    0.998      1.000    0.999      0.998    0.999     0.998     very_lethal
Weighted Avg.    0.959    0.020    0.964      0.959    0.959      0.941    0.969     0.936

=== Confusion Matrix ===

     a     b     c   <-- classified as
  9204  1260    21 |     a = non_lethal
     0 10485     0 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

OneRAttributeEval with DecisionTable

```
Correctly Classified Instances       30180               95.9466 %
Incorrectly Classified Instances      1275                4.0534 %
Kappa statistic                          0.9392
Mean absolute error                      0.0652
Root mean squared error                  0.1493
Relative absolute error                 14.6614 %
Root relative squared error             31.6714 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.884    0.003    0.994      0.884    0.936      0.910    0.989     0.984     non_lethal
                 0.995    0.058    0.896      0.995    0.943      0.915    0.990     0.972     somewhat_lethal
                 1.000    0.000    0.999      1.000    1.000      0.999    1.000     0.999     very_lethal
Weighted Avg.    0.959    0.020    0.963      0.959    0.959      0.941    0.993     0.985

=== Confusion Matrix ===

     a     b     c   <-- classified as
  9265  1213     7 |     a = non_lethal
    55 10430     0 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

Non-WEKA Approach

Non-WEKA Approach with J48

```
Correctly Classified Instances       30389               96.611  %
Incorrectly Classified Instances      1066                3.389  %
Kappa statistic                          0.9492
Mean absolute error                      0.0278
Root mean squared error                  0.1303
Relative absolute error                  6.2521 %
Root relative squared error             27.6335 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.899    0.000    0.999      0.899    0.946      0.925    0.988     0.988     non_lethal
                 0.999    0.049    0.911      0.999    0.953      0.931    0.994     0.979     somewhat_lethal
                 1.000    0.002    0.996      1.000    0.998      0.997    1.000     1.000     very_lethal
Weighted Avg.    0.966    0.017    0.969      0.966    0.966      0.951    0.994     0.989

=== Confusion Matrix ===

     a     b     c   <-- classified as
  9428  1018    39 |     a = non_lethal
     9 10476     0 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

Non-WEKA Approach with NaiveBayes

```
Correctly Classified Instances       25851               82.1841 %
Incorrectly Classified Instances      5604               17.8159 %
Kappa statistic                          0.7328
Mean absolute error                      0.1323
Root mean squared error                  0.2991
Relative absolute error                 29.771  %
Root relative squared error             63.4448 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.940    0.196    0.706      0.940    0.806      0.706    0.953     0.923     non_lethal
                 0.525    0.012    0.955      0.525    0.678      0.625    0.949     0.909     somewhat_lethal
                 1.000    0.059    0.894      1.000    0.944      0.917    0.993     0.975     very_lethal
Weighted Avg.    0.822    0.089    0.852      0.822    0.809      0.749    0.965     0.936

=== Confusion Matrix ===

     a     b     c   <-- classified as
  9857   261   367 |     a = non_lethal
  4106  5509   870 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

Non-WEKA Approach with OneR

```
Correctly Classified Instances       30174               95.9275 %
Incorrectly Classified Instances      1281                4.0725 %
Kappa statistic                          0.9389
Mean absolute error                      0.0271
Root mean squared error                  0.1648
Relative absolute error                  6.1087 %
Root relative squared error             34.9535 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.878    0.000    1.000      0.878    0.935      0.910    0.939     0.919     non_lethal
                 1.000    0.060    0.893      1.000    0.943      0.916    0.970     0.893     somewhat_lethal
                 1.000    0.001    0.998      1.000    0.999      0.998    0.999     0.998     very_lethal
Weighted Avg.    0.959    0.020    0.964      0.959    0.959      0.941    0.969     0.936

=== Confusion Matrix ===

     a     b     c   <-- classified as
  9204  1260    21 |     a = non_lethal
     0 10485     0 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

Non-WEKA Approach with DecisionTable

```
Correctly Classified Instances       30180               95.9466 %
Incorrectly Classified Instances      1275                4.0534 %
Kappa statistic                          0.9392
Mean absolute error                      0.0652
Root mean squared error                  0.1493
Relative absolute error                 14.6614 %
Root relative squared error             31.6714 %
Total Number of Instances            31455

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.884    0.003    0.994      0.884    0.936      0.910    0.989     0.984     non_lethal
                 0.995    0.058    0.896      0.995    0.943      0.915    0.990     0.972     somewhat_lethal
                 1.000    0.000    0.999      1.000    1.000      0.999    1.000     0.999     very_lethal
Weighted Avg.    0.959    0.020    0.963      0.959    0.959      0.941    0.993     0.985

=== Confusion Matrix ===

     a     b     c   <-- classified as
  9265  1213     7 |     a = non_lethal
    55 10430     0 |     b = somewhat_lethal
     0     0 10485 |     c = very_lethal
```

Analysis

Highest accuracy attribute selection algorithm and classifier model:
CfsSubsetEval with J48 - **0.972691**

Lowest root mean squared error:
CfsSubsetEval with J48 - **0.1246**

Highest TP Rate:
CfsSubsetEval with J48 - **0.973**

Lowest FP Rate:
CfsSubsetEval with J48 - **0.014**

Out of all the classification models, J48 was consistently producing the highest accuracy, with OneR and DecisionTable being close seconds, and NaiveBayes being the least accurate. Based on these values, we can see that the **CfsSubsetEval** attribute selector combined with the **J48** model classifier is the best model to be used on future datasets. We know that the model's ability to balance both high accuracy and low error rates shows that it can generalize well across different datasets, reducing the risk of overfitting or underfitting. The high true positive (TP) rate ensures that the model correctly predicts most instances of the target class, while the low false positive (FP) rate minimizes incorrect classifications, which is important for applications where false positives can lead to significant consequences.

Although the model was highly accurate on our stratified sample of the dataset, we don't know if it will be as accurate on the full dataset due to the highly skewed class attribute. This makes sense as a majority of the reported car crashes are due to total damages being over $1000 and people getting injured as opposed to deaths. In future projects the dataset could be used to focus on specific attributes, such as which streets or boroughs seem to be the most deadly or most prone to car crashes with injuries. Additionally, applying area-specific analyses could provide deeper insights. For example, identifying patterns in high-lethality zones across different boroughs or understanding factors contributing to higher crash rates on particular streets could lead to more targeted interventions, such as better traffic management, road design improvements, or stricter enforcement in high-risk areas. Of course, it is important to note that some things cannot be controlled such as distracted driving or vehicle malfunctions, but this approach could allow for more meaningful applications of the model by reducing the impact of future car crashes.

Conclusion

From our analysis above, we concluded that the CfsSubsetEval attribute selection algorithm combined with the J48 classifier model was most accurate in predicting the lethality of car crashes in New York City Boroughs. We realized the importance of using data-mining techniques such as stratified sampling or SMOTE (Synthetic Minority Over-sampling Technique) to handle imbalanced data better. We also learned about the importance of pre-processing attributes that at first glance may seem unimportant, but could be beneficial to the model. For example, our initial dataset contained the times of the car crash, which we binned into three categories for time of day, which was used throughout classification testing.

Recreating our Model

1. Open WEKA, Explorer, and open the cfssubset_final_test_data.csv file
2. Go to the "Select Attributes" tab and select 'NUMBER OF PERSONS KILLED' as the class attribute
3. Select the J48 model under the trees folder in the 'Classify' tab
4. Click start

Members and Contributions
Members: Aaryan Sumesh and Sami Saleh

Finding the Data & Building Proposal: **Aaryan**
Preprocessing Initial Attempt: **Aaryan**
Preprocessing & Project Update: **Aaryan**
Non-Weka Attribute Selection Algorithm: **Sami**
Attribute Selection Algorithms and Classifiers: **Sami**
Results Output: **Sami**
Results Analysis: **Sami**
Building Final Report: **Aaryan and Sami**

Sources

"City of New York - Motor Vehicle Collisions - Crashes." *Catalog*, Publisher data.cityofnewyork.us, 19 Oct. 2024, catalog.data.gov/dataset/motor-vehicle-collisions-crashes.

Khanna, Nilima. "J48 Classification (C4.5 Algorithm) in a Nutshell." *Medium*, Medium, 18 Aug. 2021,
medium.com/@nilimakhanna1/j48-classification-c4-5-algorithm-in-a-nutshell-24c50d20658e.

Zhang, Zixuan. "Naive Bayes Explained." *Medium*, Towards Data Science, 14 Aug. 2019, towardsdatascience.com/naive-bayes-explained-9d2b96f4a9c0.