Thomas Jefferson High School

Q1 Project Report

Student :
Ishan Ajwani
Logan Bradley

Teacher :
Dr. Yilmaz

October 21, 2024

Contents

1 Project Goal

Maryland's Montgomery County continuously collects data on traffic violations, which contains 42 different attributes and a charge for the traffic violation. The dataset and a description of each individual attribute can be found at: `https://data.montgomerycountymd.gov/Public-Safety/Traffic-Violations/4mse-ku6q/about_data`.

The dataset contains various information about the driver, which we plan on using to classify each case as a certain violation type. These can be a warning, a citation, an ESERO (Electronic Safety Equipment Repair Order), or a SERO (Safety Equipment Repair Order). After determining which attributes are most important and using them to train a multi-classification model, we hope that our model will help officers make informed decisions about what action they should take based on precedent.

2 Dataset Description

The Montgomery County Traffic Violation database is highly extensive, containing over 1.9 million instances. Throughout these instances, there were 3,610,598 missing values. While this number is high, it is expected for such a large dataset with 42 attributes, and over 95% of all missing values stemmed from only 7 attributes. This is because they are only filled in when a search is conducted, and searches aren't regularly performed.

In our class label, Violation Type, the dataset is heavily skewed towards warnings and citations, while ESERO and SERO occur significantly less frequently.

3 Pre-Processing

3.1 Stratified Random Sampling

Because our dataset is so massive, pre-processing is an important step in order to achieve a functional model. First, our possible Violation Type class labels are Warning, Citation, ESERO, and SERO. Since ESERO and SERO are simply different types of repair orders, we combined them into one label, "Repair Order." This helps avoid the issue of SERO being extremely rare in the dataset. To preprocess the data, we used Python. First, we use Pandas to load the data in a dataframe, and use scikit-learn's StratifiedShuffleSplit to stratified random sample into a significantly smaller dataset of 0.5% of the original size, being 9758 instances.

```python
import pandas as pd
from sklearn.model_selection import StratifiedShuffleSplit

df = pd.read_csv('dataset.csv')
# combine SERO and ESERO into Repair Order
df["Violation Type"] = df["Violation Type"].replace({"SERO": "Repair Order", "ESERO": "Repair Order"})
# stratified split, keep 0.5% of data
split = StratifiedShuffleSplit(n_splits=1, test_size=0.005, random_state=42)
# split for each violation
for train_index, sample_index in split.split(df, df['Violation Type']):
    stratified_sample = df.iloc[sample_index]
# save to csv
stratified_sample.to_csv('stratified_sample.csv', index=False)
```

Figure 1: Code to sample the data

3.2 Data Cleaning

Next, we used Python's csv library to clean the dataset to make it compatible with Weka. The data is opened with the csv reader, and it is looped through line-by-line. On each line, quotation marks, apostrophes, newline characters, and carriage returns are removed from the value inside the csv at this point. The csv library is crucial here, as data needs to have outer quotation marks to avoid commas in the text being counted as the end of a value, but cannot have inner quotation marks or Weka will throw an error.

```
import csv

def clean_csv(input_file, output_file):
    with open(input_file, 'r', newline='', encoding='utf-8') as csv_in:
        with open(output_file, 'w', newline='', encoding='utf-8') as csv_out:
            reader = csv.reader(csv_in)
            writer = csv.writer(csv_out)

            for row in reader:
                # Remove quotes, newline characters, apostrophes
                new_row = [field.replace('"', '').replace('\n', '').replace('\r', '').replace("'", "") for field in row]
                writer.writerow(new_row)

clean_csv('stratified_sample.csv', 'cleaned.csv')
```

Figure 2: Code to clean the data

3.3 Dropping Attributes

Attributes that are unique or almost unique for each instance have no predictive power and were therefore dropped using Weka. This included the ID value of each instance, and the date and time. Location was kept as latitude and longitude, but dropped as an address. Also, Agency was dropped since there was only one possible value, making it meaningless. In addition, "HAZMAT" was dropped due to only having one value that wasn't a "No," making it essentially useless for predicting.

Next, all attributes related to searching were dropped besides whether a search was conducted or not. All of these either had a very large portion of data missing, being around 96% missing, or were repeats of other non-search attributes, such as "Search Reason For Stop" which contained essentially the same information as "Charge."

Finally, "Charge" was dropped because it is information that an officer would only have days to months later and could not be used on the scene to make decisions.

3.4 Extracting Attributes

Description is a very useful column in this dataset, and some information can be extracted into other columns to generalize it. First, the column "Alcohol" was modified to add any instance where the description mentioned alcohol, as the attribute was missing most of these and only had around 5% of the actual alcohol-related instances.

Next, a speeding column was created to find all mentions of speeding and mark them. A value of "Yes" was used if speeding was mentioned in the description, and "Specific" was used if the exact speed of the driver was reported. Some of these may have been missed, as the format differed in some values of Description, but in general this allowed us to know if the officer had enough information to issue a citation or if they didn't have any proof of the citizen speeding. Because of the missed Specific instances, there was likely a small negative impact on performance, but not substantial. If speeding was not mentioned, a value of "No" was used.

3.5 Data Splitting

We used scikit-learn's train-test split that was taught in class to divide the data into train and test data. We used an 80-20 split, ensuring that each set contained proportional class label counts as the original dataset. This was done by adding the "stratify" parameter to the calls to train-test split, specifying Violation Type as the class label.

```
import pandas as pd
from sklearn.model_selection import train_test_split
df = pd.read_csv('cleaned.csv')
train, test = train_test_split(df, test_size=0.2, random_state=42, stratify=df['Violation Type'])
# Save CSVs
train.to_csv("train_split.csv", index=False)
test.to_csv("test_split.csv", index=False)
```

Figure 3: Code to split the data

3.6 Finalized Dataset

Our finalized train set has 7806 instances, and our test set has 1952 instances, for a total dataset of 9758 instances. The dataset contains 51.76% warnings, 43.72% citations, and 4.52% repair orders.

4 Attribute Selection

Our data, after preprocessing, contained 28 attributes: SubAgency, Description, Latitude, Longitude, Accident, Belts, Personal Injury, Property Damage, Fatal, Commercial License, Commercial Vehicle, Alcohol, Work Zone, Search Conducted, State, VehicleType, Year, Make, Model, Color, Article, Contributed To Accident, Race, Gender, Driver City, Driver State, DL State, Arrest Type, Speeding, and Violation Type. The class is Violation Type.

4.1 InfoGain

This attribute selection method was also performed in Weka. It works by checking how splitting the data on a certain attribute reduces the entropy, or how predictable the class labels are, where lower entropy equals higher predictability. Higher values given by the formula are better. We used a cutoff of 0.05. This new dataset contains Description, Model, Speeding, Driver City, and Make as attributes.

```
Ranked attributes:
 0.678222    2 Description
 0.204359   19 Model
 0.11258    29 Speeding
 0.081092   25 Driver City
 0.059688   18 Make
 0.037296   12 Alcohol
 0.019843    5 Accident
 0.019843   22 Contributed To Accident
 0.017136   27 DL State
 0.014962   14 Search Conducted
 0.014285   15 State
 0.012256    8 Property Damage
 0.011813   23 Race
 0.011804   28 Arrest Type
 0.010407   26 Driver State
 0.009337    4 Longitude
 0.009059   17 Year
 0.008984    7 Personal Injury
 0.00886    16 VehicleType
 0.007181   24 Gender
 0.007057   20 Color
 0.007051    1 SubAgency
 0.005194   21 Article
 0.005039    3 Latitude
 0.000972    6 Belts
 0.000784   11 Commercial Vehicle
 0.000459    9 Fatal
 0.000459   13 Work Zone
 0.000369   10 Commercial License
```

Figure 4: Results of InfoGain evaluation

4.2 GainRatio

This attribute selection method was also performed in Weka. GainRatioAttributeEval calculates the information gain of an attribute, or how effectively it can predict the class, and then divides it by a split information value, which prevents attributes with many different unique values from being unfairly favored. We used a cutoff of 0.05. This new dataset contains Alcohol, Speeding, Accident, Contributed to Accident, Personal Injury, Work Zone, Fatal, Description, Property Damage, Article, and Search Conducted as attributes.

```
Ranked attributes:
 0.17544   12 Alcohol
 0.13808   29 Speeding
 0.11364    5 Accident
 0.11364   22 Contributed To Accident
 0.1068     7 Personal Injury
 0.09336   13 Work Zone
 0.09336    9 Fatal
 0.09297    2 Description
 0.08254    8 Property Damage
 0.06572   21 Article
 0.06019   14 Search Conducted
 0.027     19 Model
 0.02286   11 Commercial Vehicle
 0.01716   27 DL State
 0.01588   25 Driver City
 0.0148    15 State
 0.01436   26 Driver State
 0.01386    4 Longitude
 0.01116   18 Make
 0.01072   16 VehicleType
 0.01043   28 Arrest Type
 0.00958   17 Year
 0.00781   24 Gender
 0.00579   23 Race
 0.00509    3 Latitude
 0.0048     6 Belts
 0.00259    1 SubAgency
 0.00213   20 Color
 0.00194   10 Commercial License
```

Figure 5: Results of GainRatio evaluation

4.3 OneR

This attribute selection method was also performed in Weka. OneRAttributeEval is a feature evaluation method that ranks attributes using the OneR algorithm. The OneR algorithm generates simple rules for each individual attribute and selects the attribute that performs best at classifying the data. We used a cutoff of 53.3 to determine which

attributes would be useful for classification. This new dataset contains the attributes Description, Speeding, Alcohol, Search Conducted, Race, Contributed To Accident, and Property Damage.

```
Ranked attributes:
77.1458    2 Description
58.0195   29 Speeding
55.0218   12 Alcohol
54.125    14 Search Conducted
53.9585   23 Race
53.9073   22 Contributed To Accident
53.9073    5 Accident
53.3051    8 Property Damage
53.2795    3 Latitude
52.7159    4 Longitude
52.6774    7 Personal Injury
52.6006   27 DL State
52.2931   28 Arrest Type
52.2803   17 Year
52.1138    6 Belts
51.9728   16 VehicleType
51.9216   18 Make
51.8063   26 Driver State
51.7935    9 Fatal
51.7935   13 Work Zone
51.7551   15 State
51.7551   10 Commercial License
51.7551   11 Commercial Vehicle
51.7551   24 Gender
51.7551   21 Article
51.5501   25 Driver City
51.486     1 SubAgency
51.4604   20 Color
51.1914   19 Model
```

Figure 6: Results of the OneR evaluation

4.4 WrapperSubset

This attribute selection method was performed in Weka as well. WrapperSubsetEval evaluates various subsets of features by selecting the most relevant features through testing many combinations of features with a certain model and using the model's performance for each as a comparison metric. We used a J48 decision tree classifier as our model. Upon completion, WrapperSubsetEval returns the subset of features which performed the best in its search. GreedyStepwise was used for searching. This new dataset contains the attributes Description, Accident, Belts, and VehicleType.

```
Evaluation mode:    evaluate on all training data



=== Attribute Selection on all input data ===

Search Method:
        Greedy Stepwise (forwards).
        Start set: no attributes
        Merit of best subset found:    0.773

Attribute Subset Evaluator (supervised, Class (nominal): 30 Violation Type):
        Wrapper Subset Evaluator
        Learning scheme: weka.classifiers.trees.J48
        Scheme options: -C 0.25 -M 2
        Subset evaluation: classification accuracy
        Number of folds for accuracy estimation: 5

Selected attributes: 2,5,6,16 : 4
                     Description
                     Accident
                     Belts
                     VehicleType
```

Figure 7: Results of the WrapperSubset evaluation

4.5 Self-Chosen

Using our knowledge of the data, we selected attributes ourselves that we believed contained the most predictive power. We kept the attributes Description, VehicleType, Search Conducted, Color, Race, Gender, Alcohol, Speeding, and Arrest Type.

5 Classifiers

5.1 DecisionTable

The DecisionTable algorithm creates a table of decisions based on a set of attributes, where each row is a combination of possible attribute values, where each column is a respective attribute value and the final column contains the predicted class label.

5.2 NaiveBayes

NaiveBayes assumes that, for an instance in the dataset with a known class label, knowing the value of one attribute gives you no information about the values of the other attributes.

It calculates the independent probabilities of each attribute occurring for each possible class label and uses these probabilities to make predictions for new instances.

5.3 J48

J48 recursively splits the data by an attribute, choosing the one that has the highest gain ratio for the current split. It continues until splitting is fully completed or no information is gained. This created decision tree is used to classify new instances.

5.4 RandomForest

RandomForest works by creating multiple decision trees using different subsets of instances and randomly selected attributes at each split. At each split in the tree, a random subset of attributes are chosen and the one with the highest information gain is selected. Once all trees are generated, their predictions are combined using a majority vote to classify new instances.

6 Results & Analysis

6.1 Results

```
=== Summary ===

Correctly Classified Instances        1471               75.3586 %
Incorrectly Classified Instances       481               24.6414 %
Kappa statistic                          0.5315
Mean absolute error                      0.2288
Root mean squared error                  0.3304
Relative absolute error                 63.6807 %
Root relative squared error             77.9645 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.905    0.409    0.704      0.905    0.792      0.525    0.848     0.844     Warning
                 0.553    0.087    0.831      0.553    0.664      0.509    0.839     0.813     Citation
                 0.955    0.000    1.000      0.955    0.977      0.976    0.995     0.973     Repair Order
Weighted Avg.    0.754    0.250    0.773      0.754    0.744      0.538    0.850     0.836

=== Confusion Matrix ===

   a   b   c   <-- classified as
 915  96   0 |   a = Warning
 381 472   0 |   b = Citation
   4   0  84 |   c = Repair Order
```

Figure 8: InfoGainAttributeEval with rules.DecisionTable

```
=== Summary ===

Correctly Classified Instances        1450               74.2828 %
Incorrectly Classified Instances       502               25.7172 %
Kappa statistic                          0.5115
Mean absolute error                      0.2142
Root mean squared error                  0.3381
Relative absolute error                 59.6293 %
Root relative squared error             79.7926 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.840    0.348    0.722      0.840    0.776      0.503    0.835     0.838     Warning
                 0.634    0.158    0.757      0.634    0.690      0.490    0.826     0.821     Citation
                 0.682    0.001    0.984      0.682    0.805      0.812    0.998     0.970     Repair Order
Weighted Avg.    0.743    0.249    0.749      0.743    0.740      0.511    0.838     0.837

=== Confusion Matrix ===

   a   b   c   <-- classified as
 849 162   0 |   a = Warning
 311 541   1 |   b = Citation
  16  12  60 |   c = Repair Order
```

Figure 9: InfoGainAttributeEval with bayes.NaiveBayes

```
=== Summary ===

Correctly Classified Instances        1491               76.3832 %
Incorrectly Classified Instances       461               23.6168 %
Kappa statistic                          0.5529
Mean absolute error                      0.1957
Root mean squared error                  0.3227
Relative absolute error                 54.4792 %
Root relative squared error             76.1417 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.893    0.375    0.719      0.893    0.797      0.540    0.852     0.842     Warning
                 0.589    0.098    0.823      0.589    0.686      0.525    0.848     0.818     Citation
                 0.977    0.000    1.000      0.977    0.989      0.988    0.999     0.988     Repair Order
Weighted Avg.    0.764    0.237    0.777      0.764    0.757      0.554    0.857     0.838

=== Confusion Matrix ===

   a   b   c   <-- classified as
 903 108   0 |   a = Warning
 351 502   0 |   b = Citation
   2   0  86 |   c = Repair Order
```

Figure 10: InfoGainAttributeEval with trees.J48

```
=== Summary ===

Correctly Classified Instances        1472               75.4098 %
Incorrectly Classified Instances       480               24.5902 %
Kappa statistic                          0.537
Mean absolute error                      0.2244
Root mean squared error                  0.3315
Relative absolute error                 62.4712 %
Root relative squared error             78.217  %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.850    0.349    0.724      0.850    0.782      0.513    0.837     0.826     Warning
                 0.618    0.138    0.776      0.618    0.688      0.499    0.832     0.810     Citation
                 0.977    0.000    1.000      0.977    0.989      0.988    0.999     0.992     Repair Order
Weighted Avg.    0.754    0.241    0.759      0.754    0.750      0.528    0.842     0.827

=== Confusion Matrix ===

   a   b   c   <-- classified as
 859 152   0 |   a = Warning
 326 527   0 |   b = Citation
   2   0  86 |   c = Repair Order
```

Figure 11: InfoGainAttributeEval with trees.RandomForest

```
=== Summary ===

Correctly Classified Instances        1475               75.5635 %
Incorrectly Classified Instances       477               24.4365 %
Kappa statistic                          0.5355
Mean absolute error                      0.2288
Root mean squared error                  0.3294
Relative absolute error                 63.6917 %
Root relative squared error             77.7255 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.905    0.405    0.706      0.905    0.793      0.529    0.851     0.846     Warning
               0.558    0.087    0.832      0.558    0.668      0.513    0.840     0.814     Citation
               0.955    0.000    1.000      0.955    0.977      0.976    0.995     0.972     Repair Order
Weighted Avg.  0.756    0.248    0.774      0.756    0.747      0.542    0.853     0.838

=== Confusion Matrix ===

   a   b   c   <-- classified as
 915  96   0 |   a = Warning
 377 476   0 |   b = Citation
   4   0  84 |   c = Repair Order
```

Figure 12: GainRatioAttributeEval with rules.DecisionTable

```
=== Summary ===

Correctly Classified Instances        1539               78.8422 %
Incorrectly Classified Instances       413               21.1578 %
Kappa statistic                          0.6006
Mean absolute error                      0.1924
Root mean squared error                  0.3144
Relative absolute error                 53.5545 %
Root relative squared error             74.1907 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.894    0.325    0.747      0.894    0.814      0.586    0.867     0.859     Warning
               0.646    0.097    0.837      0.646    0.729      0.576    0.864     0.851     Citation
               0.955    0.000    1.000      0.955    0.977      0.976    0.998     0.986     Repair Order
Weighted Avg.  0.788    0.211    0.798      0.788    0.784      0.599    0.872     0.861

=== Confusion Matrix ===

   a   b   c   <-- classified as
 904 107   0 |   a = Warning
 302 551   0 |   b = Citation
   4   0  84 |   c = Repair Order
```

Figure 13: GainRatioAttributeEval with bayes.NaiveBayes

```
=== Summary ===

Correctly Classified Instances        1520               77.8689 %
Incorrectly Classified Instances       432               22.1311 %
Kappa statistic                          0.5816
Mean absolute error                      0.1934
Root mean squared error                  0.3201
Relative absolute error                 53.8348 %
Root relative squared error             75.5387 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.899    0.351    0.734      0.899    0.808      0.569    0.851     0.819     Warning
               0.615    0.093    0.837      0.615    0.709      0.555    0.848     0.827     Citation
               0.977    0.000    1.000      0.977    0.989      0.988    1.000     0.991     Repair Order
Weighted Avg.  0.779    0.222    0.791      0.779    0.773      0.582    0.856     0.830

=== Confusion Matrix ===

   a   b   c   <-- classified as
 909 102   0 |   a = Warning
 328 525   0 |   b = Citation
   2   0  86 |   c = Repair Order
```

Figure 14: GainRatioAttributeEval with trees.J48

```
=== Summary ===

Correctly Classified Instances        1535               78.6373 %
Incorrectly Classified Instances       417               21.3627 %
Kappa statistic                          0.5972
Mean absolute error                      0.1916
Root mean squared error                  0.3165
Relative absolute error                 53.3432 %
Root relative squared error             74.6776 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.889    0.324    0.747      0.889    0.812      0.581    0.860     0.851     Warning
               0.645    0.102    0.831      0.645    0.726      0.569    0.858     0.837     Citation
               0.977    0.000    1.000      0.977    0.989      0.988    1.000     0.993     Repair Order
Weighted Avg.  0.786    0.212    0.795      0.786    0.782      0.594    0.865     0.851

=== Confusion Matrix ===

   a   b   c   <-- classified as
 899 112   0 |   a = Warning
 303 550   0 |   b = Citation
   2   0  86 |   c = Repair Order
```

Figure 15: GainRatioAttributeEval with trees.RandomForest

```
=== Summary ===

Correctly Classified Instances        1475               75.5635 %
Incorrectly Classified Instances       477               24.4365 %
Kappa statistic                          0.5355
Mean absolute error                      0.2288
Root mean squared error                  0.3294
Relative absolute error                 63.6917 %
Root relative squared error             77.7255 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.905    0.405    0.706      0.905    0.793      0.529    0.851     0.846     Warning
               0.558    0.087    0.832      0.558    0.668      0.513    0.840     0.814     Citation
               0.955    0.000    1.000      0.955    0.977      0.976    0.995     0.972     Repair Order
Weighted Avg.  0.756    0.248    0.774      0.756    0.747      0.542    0.853     0.838

=== Confusion Matrix ===

   a   b   c   <-- classified as
 915  96   0 |   a = Warning
 377 476   0 |   b = Citation
   4   0  84 |   c = Repair Order
```

Figure 16: OneRAttributeEval with rules.DecisionTable

```
=== Summary ===

Correctly Classified Instances        1535               78.6373 %
Incorrectly Classified Instances       417               21.3627 %
Kappa statistic                          0.5963
Mean absolute error                      0.1939
Root mean squared error                  0.3161
Relative absolute error                 53.9852 %
Root relative squared error             74.5842 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.901    0.337    0.742      0.901    0.814      0.584    0.863     0.859     Warning
               0.632    0.091    0.844      0.632    0.723      0.572    0.861     0.853     Citation
               0.966    0.000    1.000      0.966    0.983      0.982    0.999     0.988     Repair Order
Weighted Avg.  0.786    0.214    0.798      0.786    0.782      0.596    0.868     0.862

=== Confusion Matrix ===

   a   b   c   <-- classified as
 911 100   0 |   a = Warning
 314 539   0 |   b = Citation
   3   0  85 |   c = Repair Order
```

Figure 17: OneRAttributeEval with bayes.NaiveBayes

```
=== Summary ===

Correctly Classified Instances        1518               77.7664 %
Incorrectly Classified Instances       434               22.2336 %
Kappa statistic                          0.5802
Mean absolute error                      0.1932
Root mean squared error                  0.3208
Relative absolute error                 53.7738 %
Root relative squared error             75.7099 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.889    0.342    0.736      0.889    0.806      0.565    0.849     0.816     Warning
                 0.625    0.102    0.826      0.625    0.712      0.551    0.846     0.826     Citation
                 0.977    0.000    1.000      0.977    0.989      0.988    1.000     0.991     Repair Order
Weighted Avg.    0.778    0.222    0.788      0.778    0.773      0.578    0.855     0.828

=== Confusion Matrix ===

   a   b   c   <-- classified as
 899 112   0 |   a = Warning
 320 533   0 |   b = Citation
   2   0  86 |   c = Repair Order
```

Figure 18: OneRAttributeEval with trees.J48

```
=== Summary ===

Correctly Classified Instances        1521               77.9201 %
Incorrectly Classified Instances       431               22.0799 %
Kappa statistic                          0.5848
Mean absolute error                      0.1921
Root mean squared error                  0.3243
Relative absolute error                 53.4606 %
Root relative squared error             76.5152 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.864    0.312    0.748      0.864    0.802      0.563    0.850     0.841     Warning
                 0.658    0.125    0.804      0.658    0.723      0.552    0.847     0.822     Citation
                 0.977    0.000    1.000      0.977    0.989      0.988    1.000     0.997     Repair Order
Weighted Avg.    0.779    0.216    0.784      0.779    0.776      0.577    0.855     0.840

=== Confusion Matrix ===

   a   b   c   <-- classified as
 874 137   0 |   a = Warning
 292 561   0 |   b = Citation
   2   0  86 |   c = Repair Order
```

Figure 19: OneRAttributeEval with trees.RandomForest

```
=== Summary ===

Correctly Classified Instances        1475               75.5635 %
Incorrectly Classified Instances       477               24.4365 %
Kappa statistic                          0.5355
Mean absolute error                      0.2288
Root mean squared error                  0.3294
Relative absolute error                 63.6917 %
Root relative squared error             77.7255 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.905    0.405    0.706      0.905    0.793      0.529    0.851     0.846     Warning
                 0.558    0.087    0.832      0.558    0.668      0.513    0.840     0.814     Citation
                 0.955    0.000    1.000      0.955    0.977      0.976    0.995     0.972     Repair Order
Weighted Avg.    0.756    0.248    0.774      0.756    0.747      0.542    0.853     0.838

=== Confusion Matrix ===

   a   b   c   <-- classified as
 915  96   0 |   a = Warning
 377 476   0 |   b = Citation
   4   0  84 |   c = Repair Order
```

Figure 20: WrapperSubsetEval with rules.DecisionTable

```
=== Summary ===

Correctly Classified Instances        1498               76.7418 %
Incorrectly Classified Instances       454               23.2582 %
Kappa statistic                          0.5605
Mean absolute error                      0.2144
Root mean squared error                  0.3238
Relative absolute error                 59.6851 %
Root relative squared error             76.4125 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.886    0.357    0.727      0.886    0.799      0.548    0.854     0.851     Warning
                 0.608    0.105    0.819      0.608    0.698      0.534    0.849     0.829     Citation
                 0.943    0.002    0.965      0.943    0.954      0.952    0.997     0.976     Repair Order
Weighted Avg.    0.767    0.231    0.778      0.767    0.762      0.560    0.858     0.847

=== Confusion Matrix ===

   a   b   c   <-- classified as
 896 114   1 |   a = Warning
 332 519   2 |   b = Citation
   4   1  83 |   c = Repair Order
```

Figure 21: WrapperSubsetEval with bayes.NaiveBayes

```
=== Summary ===

Correctly Classified Instances        1504               77.0492 %
Incorrectly Classified Instances       448               22.9508 %
Kappa statistic                          0.566
Mean absolute error                      0.1938
Root mean squared error                  0.322
Relative absolute error                 53.9325 %
Root relative squared error             75.9743 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.892    0.360    0.727      0.892    0.801      0.552    0.853     0.843     Warning
                 0.605    0.099    0.826      0.605    0.698      0.538    0.849     0.816     Citation
                 0.977    0.000    1.000      0.977    0.989      0.988    0.999     0.988     Repair Order
Weighted Avg.    0.770    0.230    0.782      0.770    0.765      0.566    0.858     0.838

=== Confusion Matrix ===

   a   b   c   <-- classified as
 902 109   0 |   a = Warning
 337 516   0 |   b = Citation
   2   0  86 |   c = Repair Order
```

Figure 22: WrapperSubsetEval with trees.J48

```
=== Summary ===

Correctly Classified Instances        1505               77.1004 %
Incorrectly Classified Instances       447               22.8996 %
Kappa statistic                          0.5673
Mean absolute error                      0.1975
Root mean squared error                  0.3255
Relative absolute error                 54.985  %
Root relative squared error             76.8069 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.887    0.354    0.729      0.887    0.801      0.552    0.849     0.842     Warning
                 0.612    0.104    0.821      0.612    0.701      0.538    0.845     0.813     Citation
                 0.977    0.000    1.000      0.977    0.989      0.988    0.999     0.989     Repair Order
Weighted Avg.    0.771    0.229    0.781      0.771    0.766      0.566    0.854     0.836

=== Confusion Matrix ===

   a   b   c   <-- classified as
 897 114   0 |   a = Warning
 331 522   0 |   b = Citation
   2   0  86 |   c = Repair Order
```

Figure 23: WrapperSubsetEval with trees.RandomForest

```
=== Summary ===

Correctly Classified Instances        1471               75.3586 %
Incorrectly Classified Instances       481               24.6414 %
Kappa statistic                          0.5315
Mean absolute error                      0.2288
Root mean squared error                  0.3304
Relative absolute error                 63.6807 %
Root relative squared error             77.9645 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.905    0.409    0.704      0.905    0.792      0.525    0.848     0.844     Warning
                 0.553    0.087    0.831      0.553    0.664      0.509    0.839     0.813     Citation
                 0.955    0.000    1.000      0.955    0.977      0.976    0.995     0.973     Repair Order
Weighted Avg.    0.754    0.250    0.773      0.754    0.744      0.538    0.850     0.836

=== Confusion Matrix ===

   a   b   c   <-- classified as
 915  96   0 |   a = Warning
 381 472   0 |   b = Citation
   4   0  84 |   c = Repair Order
```

Figure 24: Self-Chosen Attributes with rules.DecisionTable

```
=== Summary ===

Correctly Classified Instances        1506               77.1516 %
Incorrectly Classified Instances       446               22.8484 %
Kappa statistic                          0.5696
Mean absolute error                      0.2004
Root mean squared error                  0.3202
Relative absolute error                 55.7852 %
Root relative squared error             75.5483 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.875    0.337    0.736      0.875    0.800      0.553    0.859     0.857     Warning
                 0.630    0.114    0.811      0.630    0.709      0.540    0.856     0.848     Citation
                 0.955    0.002    0.955      0.955    0.955      0.952    0.999     0.979     Repair Order
Weighted Avg.    0.772    0.224    0.779      0.772    0.767      0.566    0.864     0.858

=== Confusion Matrix ===

   a   b   c   <-- classified as
 885 124   2 |   a = Warning
 314 537   2 |   b = Citation
   3   1  84 |   c = Repair Order
```

Figure 25: Self-Chosen Attributes with bayes.NaiveBayes

```
=== Summary ===

Correctly Classified Instances        1512               77.459  %
Incorrectly Classified Instances       440               22.541  %
Kappa statistic                          0.5742
Mean absolute error                      0.1898
Root mean squared error                  0.3217
Relative absolute error                 52.8325 %
Root relative squared error             75.9224 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.889    0.349    0.733      0.889    0.803      0.559    0.851     0.838     Warning
                 0.618    0.102    0.825      0.618    0.706      0.545    0.849     0.818     Citation
                 0.977    0.000    1.000      0.977    0.989      0.988    1.000     0.991     Repair Order
Weighted Avg.    0.775    0.225    0.785      0.775    0.769      0.572    0.857     0.836

=== Confusion Matrix ===

   a   b   c   <-- classified as
 899 112   0 |   a = Warning
 326 527   0 |   b = Citation
   2   0  86 |   c = Repair Order
```

Figure 26: Self-Chosen Attributes with trees.J48

```
=== Summary ===

Correctly Classified Instances        1471               75.3586 %
Incorrectly Classified Instances       481               24.6414 %
Kappa statistic                          0.5388
Mean absolute error                      0.1981
Root mean squared error                  0.3426
Relative absolute error                 55.1407 %
Root relative squared error             80.8435 %
Total Number of Instances             1952

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.810    0.307    0.739      0.810    0.773      0.507    0.827     0.817     Warning
               0.664    0.175    0.747      0.664    0.703      0.498    0.823     0.799     Citation
               0.977    0.000    1.000      0.977    0.989      0.988    1.000     0.996     Repair Order
Weighted Avg.  0.754    0.235    0.754      0.754    0.752      0.525    0.833     0.817

=== Confusion Matrix ===

   a   b   c   <-- classified as
 819 192   0 |   a = Warning
 287 566   0 |   b = Citation
   2   0  86 |   c = Repair Order
```

Figure 27: Self-Chosen Attributes with trees.RandomForest

6.2 Analysis

Accuracy Table (% correct on test data)					
	InfoGain	GainRatio	OneR	WrapperSubset	Self-Chosen
DecisionTable	75.36	75.56	75.56	75.56	75.36
NaiveBayes	74.28	78.84	78.64	76.74	77.15
J48	76.38	77.87	77.77	77.05	77.46
RandomForest	75.41	78.64	77.92	77.10	75.36

Figure 28: Summary of Model Accuracies

7 Conclusion

As previously stated, GainRatio with NaiveBayes performed the best out of all 20 models tested. With a FP Rate of 0.325 for warnings, it is too lenient on many traffic violations, and so should probably not be blindly used by officers. However, since it has an overall accuracy of 78.84%, the model does perform fairly well on our dataset, and is still an overall successful model for testing purposes.

8 Reproducibility

1. Open Weka and load `train_split.csv` (located in the "Cleaned Data" folder of our Google Drive folder).

2. Ensure "Violation Type" is set as the class variable. If not, open the Editor by clicking the "Edit..." button, right-click on "Violation Type," select "Attribute as class," and click "OK."

3. Go to the "Select attributes" tab, click the top "Choose" button in the "Attribute Evaluator" box, and select `GainRatioAttributeEval`. Click "Yes" on the alert that pops up to switch to the Ranker search method.

4. Click on the Search Method box where it says `Ranker`, and in the resulting popup, change the threshold value to 0.05. Click OK.
5. Set the class by clicking on "No class" and changing it to "(Nom) Violation Type."
6. Click Start. The window will show the attributes to be kept. Keep these and the class label Violation Type, remove all of the other attributes in the Preprocess tab.
7. For future use, save this train dataset as an `.arff` file.
8. Open the Classify tab, click Choose, open the `bayes` folder, and select `NaiveBayes`.
9. Under Test Options, choose "Supplied test set," then "Open file..." and select `train_split.csv` (located in the "Cleaned Data" folder of our Google Drive folder). Ensure the Class dropdown box has "(Nom) Violation Type" selected; if not, select it. Click Close.
10. Click Start. The model will be created and its output will appear in the output window.

9 Tasks Performed

- Finding the dataset: Logan & Ishan
- Building proposal: Mostly Ishan
- Preprocessing Initial Attempt: Logan
- Preprocessing & Project Update: Mostly Logan
- Non-Weka Attribute Selection Algorithm: Logan & Ishan
- Attribute Selection Algorithms and Classifiers: Mostly Logan
- Results Output: Logan
- Results Analysis: Logan & Ishan
- Building Final Report: Logan & Ishan

10 Appendix & Sources

Link to dataset: `https://data.montgomerycountymd.gov/Public-Safety/Traffic-Violations/4mse-ku6q/about_data`

Files Attached (in Google Drive Folder):

- `Traffic_Violations_20240920.csv`: Our starting dataset
- `.arff` files: The five resulting datasets, one for each attribute selection method
- `clean_data.py`: Removes quotation marks, newlines, carriage returns, and apostrophes to make the dataset load in Weka

- `description_split.py`: Uses the Description attribute to make the Speeding attribute and recalculate the Alcohol attribute as outlined in section 3.4
- `stratified_sample.py`: Replaces ESERO and SERO with Repair Order in Violation Type, uses stratified random sampling to keep only 0.5% of data
- `train_test_split.py`: Uses an 80-20 stratified split to create a train set and test set
- `cleaned_dropped_missingvals_replaced.csv`: The dataset, after cleaning, removing preprocessing columns, and replacing missing values
- `description_split.csv`: The dataset, after everything in `cleaned_dropped_missingvals_repl` and running `description_split.py` to create the Speeding attribute and recalculate the Alcohol attribute
- `train_split.csv`: The train set
- `test_split.csv`: The test set