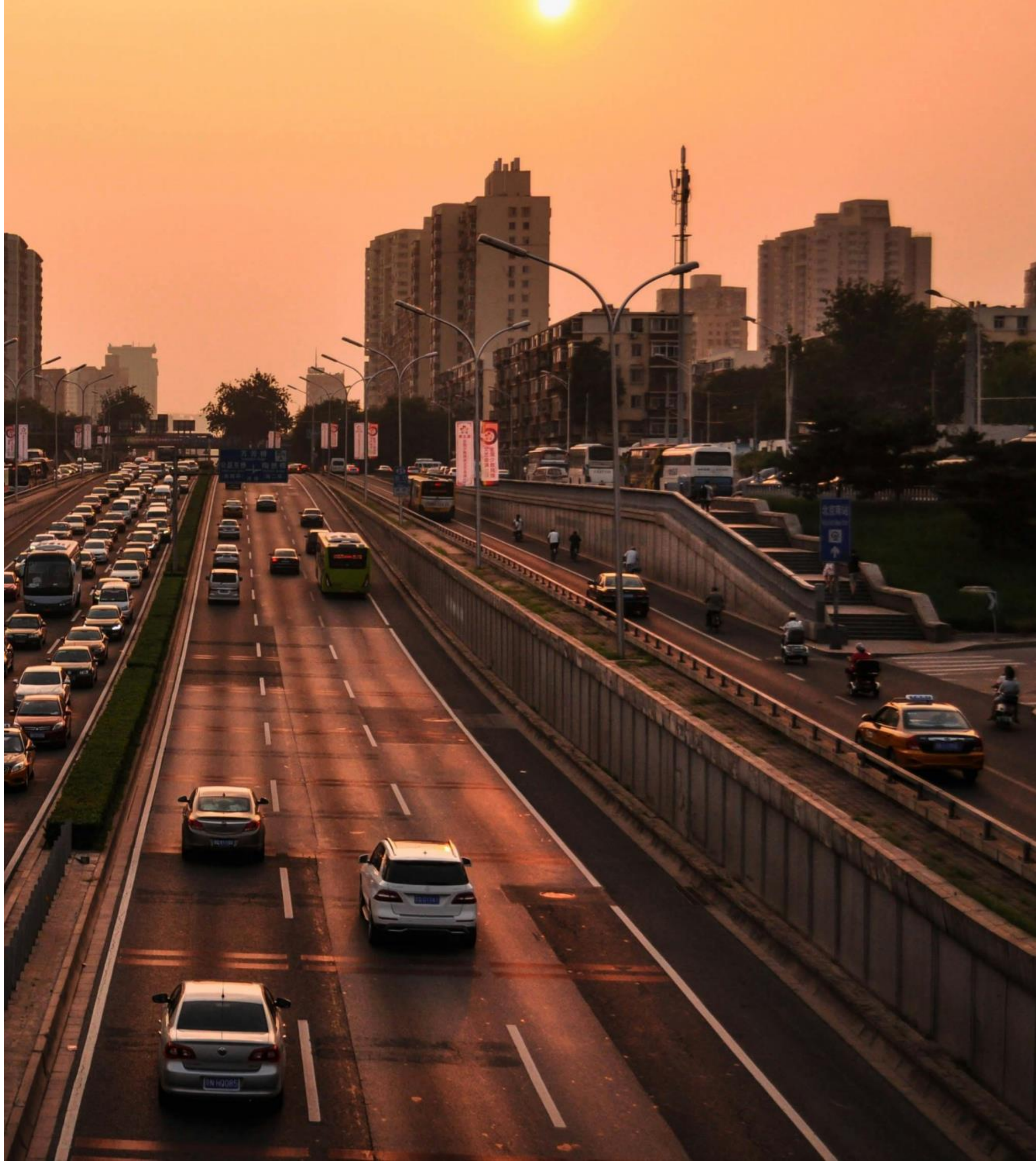
Ishan A & Logan B

TRAFFIC & TROUBLE

An Analysis of Predictive Models for Classifying Traffic Violations



I. PROJECT GOAL

A

Dataset

- Montgomery County Traffic Violations
- 42 attributes

B

Purpose

- Classify violations
- Help officers make informed decisions

II. Dataset

DESCRIPTION

A

Size

- 1.9 million instances
- 42 attributes

B

Missing Values

- 3.4 million missing
- 7 main attributes

C

Skew

- Few ESERO
- Fewer SERO

III. PRE

PROCESSING

A

Random Sampling

- Stratified
- 0.5%
- Repair Order (ESERO + SERO)

B

Cleaning

- Python
- CSV library
- Removed “, ‘, \n, \r

C

Attribute Removal

- ID
- Address
- Agency
- HAZMAT
- Search specifics

D

Extraction

- Description
- Alcohol revamp
- Speeding

E

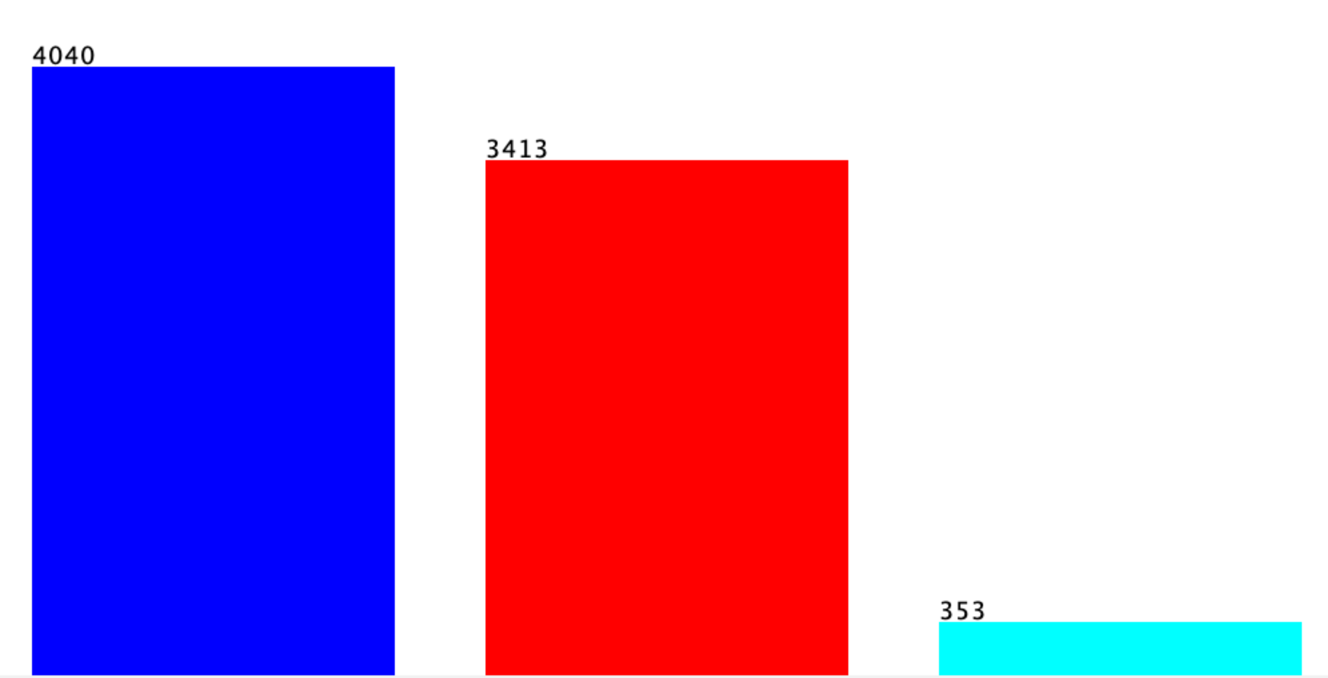
Splitting

- Stratified
- 80-20 split

Final Stats

Train: 7806 instances
Test: 1952 instances
29 attributes

51.76% warnings
43.72% citations
4.52% repair orders



IV. ATTRIBUTE SELECTION

A

Selection Algorithms

B

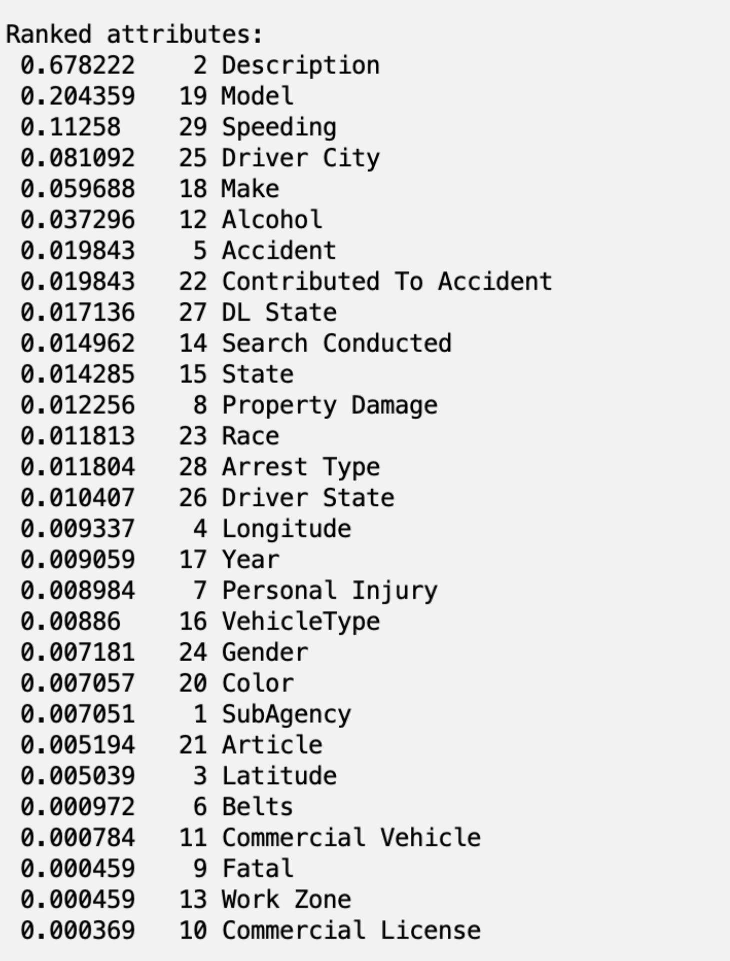
Classification

Selection Algorithms

- InfoGain
- GainRatio
- OneR
- WrapperSubset
- Self-Chosen

InfoGain

- Split on attribute to minimize entropy
- Cutoff of 0.05

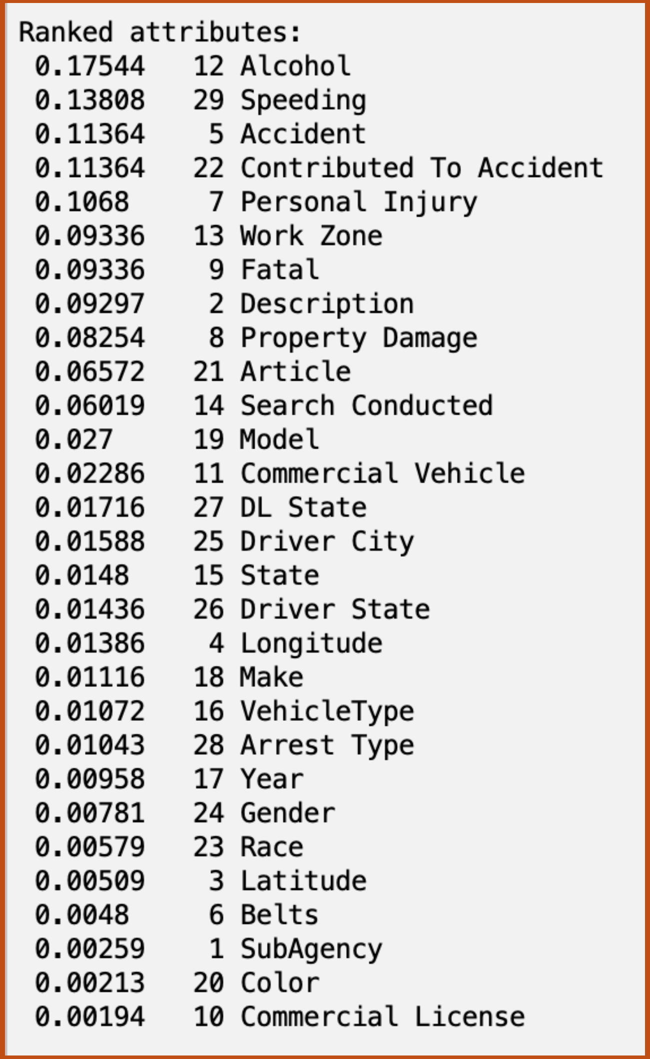


```
Ranked attributes:
 0.678222    2 Description
 0.204359   19 Model
 0.11258    29 Speeding
 0.081092   25 Driver City
 0.059688   18 Make
 0.037296   12 Alcohol
 0.019843    5 Accident
 0.019843   22 Contributed To Accident
 0.017136   27 DL State
 0.014962   14 Search Conducted
 0.014285   15 State
 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
```

GainRatio

- Information gain / split info value

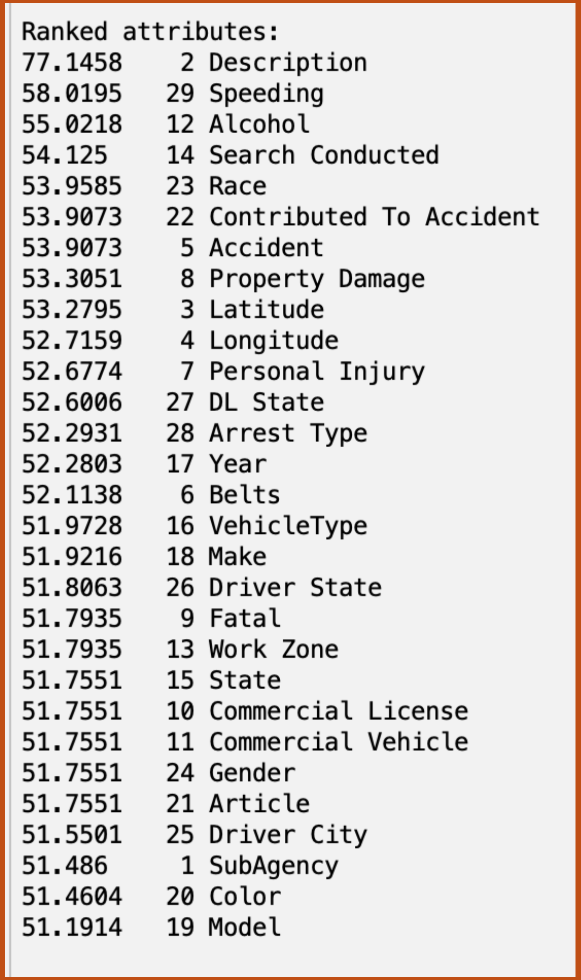
- Cutoff of 0.05



```
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
```

OneR

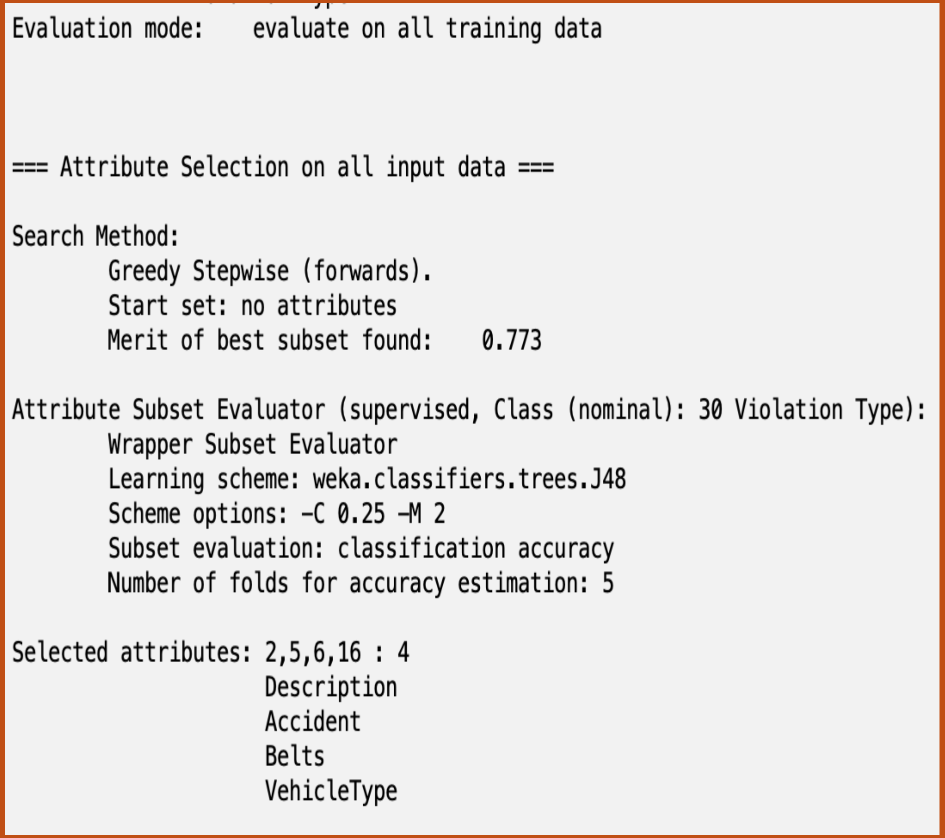
- One set of rules – 1 attribute
- Cutoff of 53.3



```
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
```

WrapperSubset

- Combinations of features
- Tested with J48 tree



```
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
```

Self-Chosen

- Using our knowledge of the data

- Description
- VehicleType
- Search Conducted
- Color
- Race
- Gender
- Alcohol
- Speeding
- Arrest Type

A

Selection Algorithms

B

Classification

Classification

rules.DecisionTable

bayes.NaiveBayes

trees.J48

trees.RandomForest

rules.DecisionTable

- Creates a table of decisions from attributes
- Each row: combination of attribute values
- Final column: predicted class label

bayes.NaiveBayes

- Creates a table of decisions from attributes
- Each row: combination of attribute values
- Final column: predicted class label

bayes.NaiveBayes

Assumes independence between attributes
Calculates probabilities for each class
Uses these to predict new instances

trees.J48

Assumes independence between attributes
Calculates probabilities for each class
Uses these to predict new instances

trees.J48

Recursively splits data by attribute
Chooses split based on highest gain ratio
Builds a decision tree for classification

trees.RandomFores

trees.J48

Recursively splits data by attribute
Chooses split based on highest gain ratio
Builds a decision tree for classification

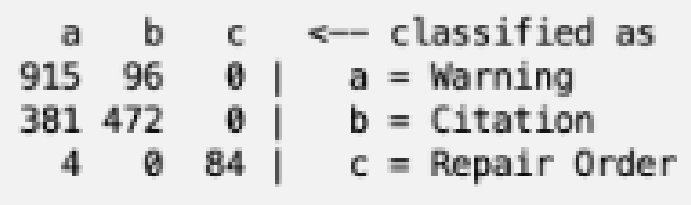
trees.RandomForest

Builds multiple trees from random subsets
At each split, selects random attributes
Uses majority vote from trees for final prediction

V. FINAL

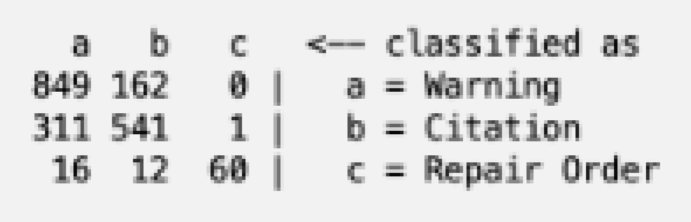
RESULTS

InfoGain



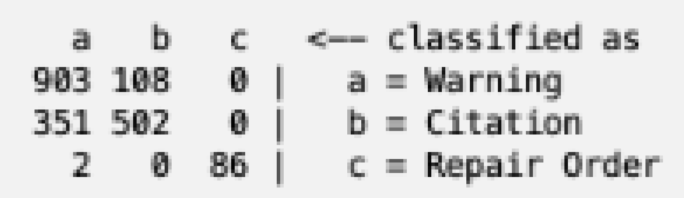
```
  a   b   c   <-- classified as
915  96   0 |   a = Warning
381 472   0 |   b = Citation
  4   0  84 |   c = Repair Order
```

Using DecisionTable



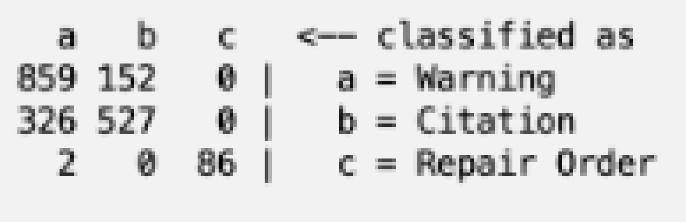
```
  a   b   c   <-- classified as
849 162   0 |   a = Warning
311 541   1 |   b = Citation
 16  12  60 |   c = Repair Order
```

Using NaiveBayes



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

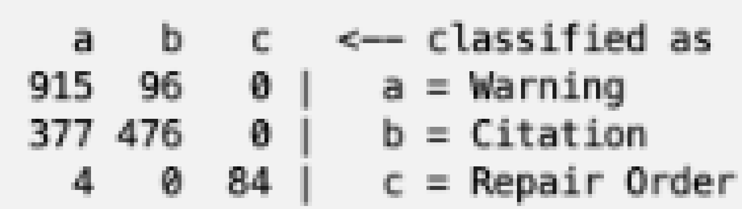
Using J48



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

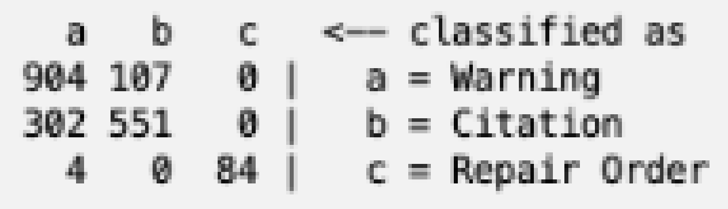
Using RandomForest

GainRatio



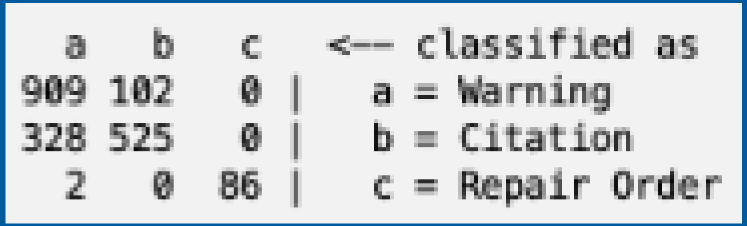
```
  a   b   c   <-- classified as
915  96   0 |   a = Warning
377 476   0 |   b = Citation
  4   0  84 |   c = Repair Order
```

Using DecisionTable



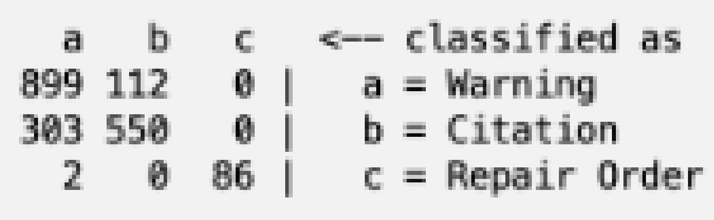
```
  a   b   c   <-- classified as
904 107   0 |   a = Warning
302 551   0 |   b = Citation
  4   0  84 |   c = Repair Order
```

Using NaiveBayes



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

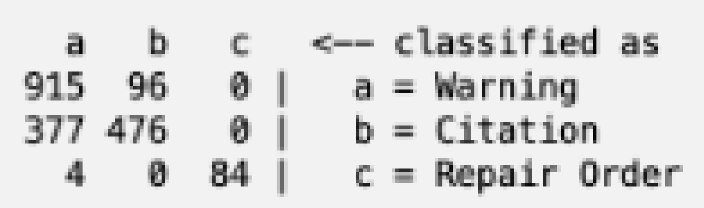
Using J48



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

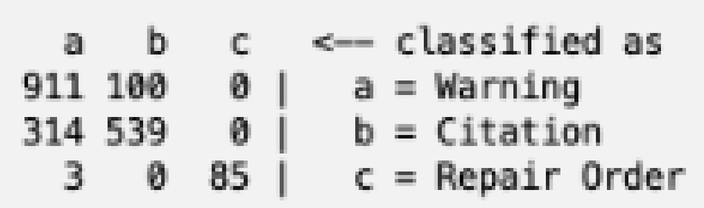
Using RandomForest

OneR



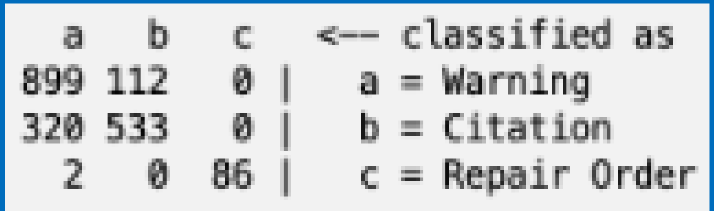
```
  a   b   c   <-- classified as
915  96   0 |   a = Warning
377 476   0 |   b = Citation
  4   0  84 |   c = Repair Order
```

Using DecisionTable



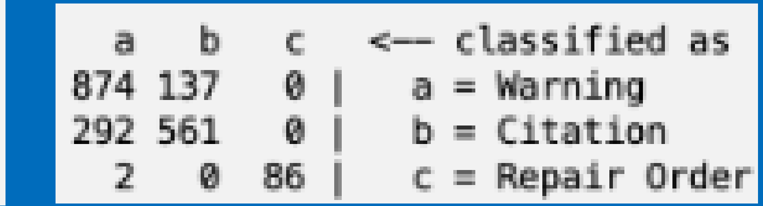
```
  a   b   c   <-- classified as
911 100   0 |   a = Warning
314 539   0 |   b = Citation
  3   0  85 |   c = Repair Order
```

Using NaiveBayes



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

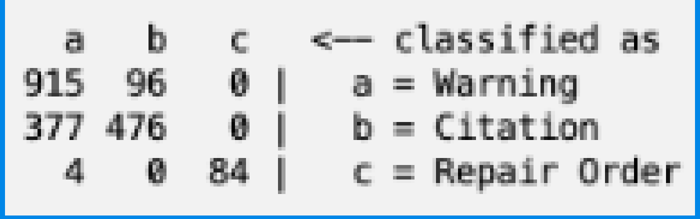
Using J48



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

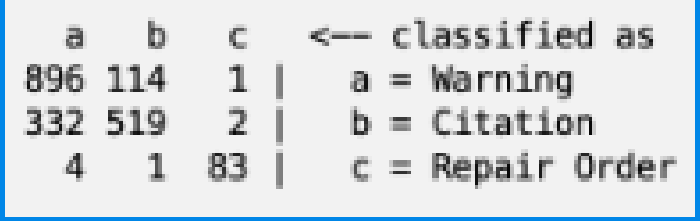
Using RandomForest

Wrapper Subset



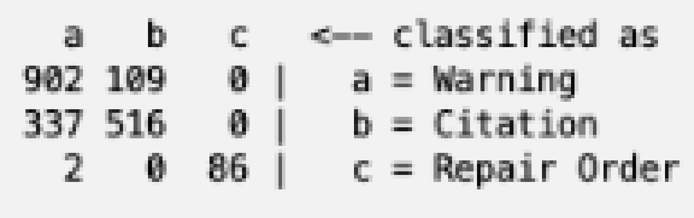
```
  a   b   c   <-- classified as
915  96   0 |   a = Warning
377 476   0 |   b = Citation
  4   0  84 |   c = Repair Order
```

Using DecisionTable



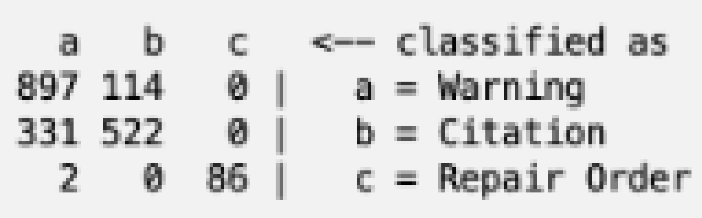
```
  a   b   c   <-- classified as
896 114   1 |   a = Warning
332 519   2 |   b = Citation
  4   1  83 |   c = Repair Order
```

Using NaiveBayes



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

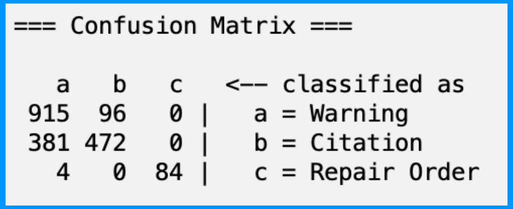
Using J48



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

Using RandomForest

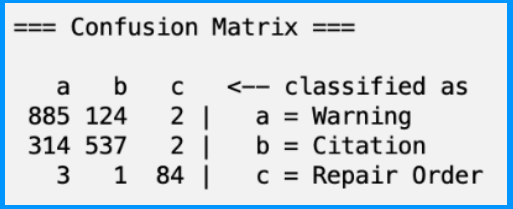
Self Chosen



```
=== Confusion Matrix ===

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

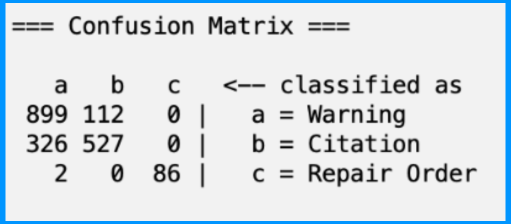
Using DecisionTable



```
=== Confusion Matrix ===

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

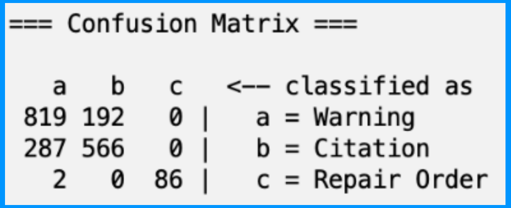
Using NaiveBayes



```
=== Confusion Matrix ===

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

Using J48



```
=== Confusion Matrix ===

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

Using RandomForest

Summary

	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	77.1	75.36

Gain Ratio
With NaïveBayes

0.325
(FP rate, warnings)

78.84%

Reproduction

1. Open Weka and load train_split.csv (located in the “Cleaned Data” folder of our Google Drive folder).
2. Ensure “Violation Type” is already 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 number in threshold to 0.05. Click OK.
5. Set the class by clicking on “No class” and changing it to “(Nom) Violation Type.”
6. Click Start.
7. 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.
8. For future use, save this train dataset as an arff file.
9. Open the Classify tab, click Choose, open the bayes folder, and select NaiveBayes.
10. 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.
11. Click Start.
12. The model will be created and its output will appear in the output window.

Sources

DecisionTable. (n.d.). In WEKA Documentation. https://weka.sourceforge.io/doc.dev/weka/classifiers/rules/DecisionTable.html

GainRatioAttributeEval. (n.d.). In WEKA Documentation. https://weka.sourceforge.io/doc.dev/weka/attributeSelection/GainRatioAttributeEval.html

InfoGainAttributeEval. (n.d.). In WEKA Documentation. https://weka.sourceforge.io/doc.dev/weka/attributeSelection/InfoGainAttributeEval.html

J48. (n.d.). In WEKA Documentation. https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/J48.html

NaiveBayes. (n.d.). In WEKA Documentation. https://weka.sourceforge.io/doc.dev/weka/classifiers/bayes/NaiveBayes.html

OneRAttributeEval. (n.d.). In WEKA Documentation. https://weka.sourceforge.io/doc.dev/weka/attributeSelection/OneRAttributeEval.html

RandomForest. (n.d.). In WEKA Documentation. https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/RandomForest.html

WrapperSubsetEval. (n.d.). In WEKA Documentation. https://weka.sourceforge.io/doc.dev/weka/attributeSelection/WrapperSubsetEval.html

Thank you for your undivided and desegregated attention and concentration!

We hope you enjoyed this intellectual and spiritual journey through the realm of machine Learning models