Driver Liability Prediction Using Crash Data

Arjun Pagidi and Abhi Palikala

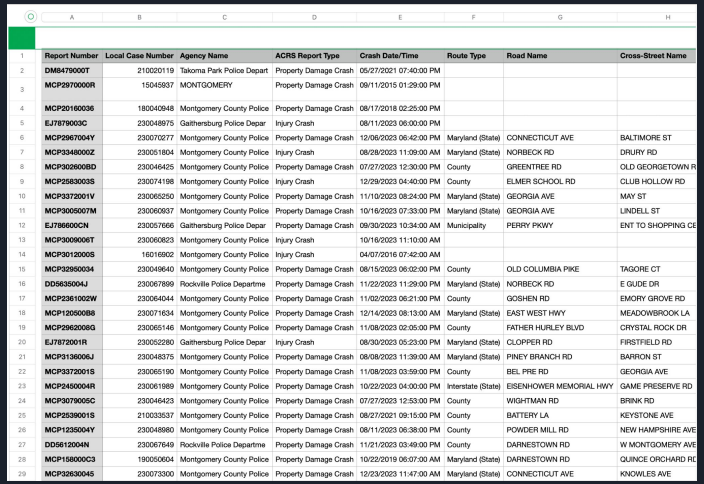
Statement and Project Goal

- Determining fault is vital for insurance payouts and other legal or monetary activities. However, determining fault is a difficult process when the parties don't agree.
- A good model could help improve road safety by highlighting key risk factors, guiding preventative measures, and informing traffic policies.
- Additionally, the insights from this project could be valuable for law enforcement agencies, insurance companies, and urban planners to better understand the dynamics of traffic incidents and ultimately reduce the number of preventable accidents on the road.

Therefore, the goal of this project is to develop a machine learning model that predicts whether a driver was at fault in a traffic incident using the Montgomery County Crash Reporting Incidents dataset.

Dataset

The data set consists of about 180,000 instances with 38 attributes (not including the class label). This dataset provides information on motor vehicle operators (drivers) involved in traffic collisions occurring on county and local roadways.



	A	B	C	D	E	F	G	H
1	Report Number	Local Case Number	Agency Name	ACRS Report Type	Crash Date/Time	Route Type	Road Name	Cross-Street Name
2	DM8479000T	210020119	Takoma Park Police Depart	Property Damage Crash	05/27/2021 07:40:00 PM			
3	MCP2970000R	15045937	MONTGOMERY	Property Damage Crash	09/11/2015 01:29:00 PM			
4	MCP20160036	180040948	Montgomery County Police	Property Damage Crash	08/17/2018 02:25:00 PM			
5	EJ7879003C	230048975	Gaithersburg Police Depar	Injury Crash	08/11/2023 06:00:00 PM			
6	MCP2967004Y	230070277	Montgomery County Police	Property Damage Crash	12/06/2023 06:42:00 PM	Maryland (State)	CONNECTICUT AVE	BALTIMORE ST
7	MCP3348000Z	230051804	Montgomery County Police	Injury Crash	08/28/2023 11:09:00 AM	Maryland (State)	NORBECK RD	DRURY RD
8	MCP302600BD	230046425	Montgomery County Police	Property Damage Crash	07/27/2023 12:30:00 PM	County	GREENTREE RD	OLD GEORGETOWN R
9	MCP258300GS	230074198	Montgomery County Police	Injury Crash	12/29/2023 04:40:00 PM	County	ELMER SCHOOL RD	CLUB HOLLOW RD
10	MCP3372001V	230065250	Montgomery County Police	Property Damage Crash	11/10/2023 08:24:00 PM	Maryland (State)	GEORGIA AVE	MAY ST
11	MCP3005007M	230060937	Montgomery County Police	Property Damage Crash	10/16/2023 07:33:00 PM	Maryland (State)	GEORGIA AVE	LINDELL ST
12	EJ786600CN	230057666	Gaithersburg Police Depar	Property Damage Crash	09/30/2023 10:34:00 AM	Municipality	PERRY PKWY	ENT TO SHOPPING CE
13	MCP3009006T	230060823	Montgomery County Police	Injury Crash	10/16/2023 11:10:00 AM			
14	MCP301200GS	16016902	Montgomery County Police	Injury Crash	04/07/2016 07:42:00 AM			
15	MCP32950034	230049640	Montgomery County Police	Property Damage Crash	08/15/2023 06:02:00 PM	County	OLD COLUMBIA PIKE	TAGORE CT
16	DD5635004J	230067899	Rockville Police Departme	Property Damage Crash	11/22/2023 11:29:00 PM	Maryland (State)	NORBECK RD	E GUDE DR
17	MCP2361002W	230064044	Montgomery County Police	Property Damage Crash	11/02/2023 06:21:00 PM	County	GOSHEN RD	EMORY GROVE RD
18	MCP1205008B	230071634	Montgomery County Police	Property Damage Crash	12/14/2023 08:13:00 AM	Maryland (State)	EAST WEST HWY	MEADOWBROOK LA
19	MCP2962008G	230065146	Montgomery County Police	Property Damage Crash	11/08/2023 02:05:00 PM	County	FATHER HURLEY BLVD	CRYSTAL ROCK DR
20	EJ7872001R	230052280	Gaithersburg Police Depar	Injury Crash	08/30/2023 05:23:00 PM	Maryland (State)	CLOPPER RD	FIRSTFIELD RD
21	MCP3136006J	230048375	Montgomery County Police	Property Damage Crash	08/08/2023 11:39:00 AM	Maryland (State)	PINEY BRANCH RD	BARRON ST
22	MCP3372001S	230065190	Montgomery County Police	Property Damage Crash	11/08/2023 03:59:00 PM	County	BEL PRE RD	GEORGIA AVE
23	MCP2450004R	230061989	Montgomery County Police	Property Damage Crash	10/22/2023 04:00:00 PM	Interstate (State)	EISENHOWER MEMORIAL HWY	GAME PRESERVE RD
24	MCP3079005C	230046423	Montgomery County Police	Property Damage Crash	07/27/2023 12:53:00 PM	County	WIGHTMAN RD	BRINK RD
25	MCP2539001S	210033537	Montgomery County Police	Property Damage Crash	08/27/2021 09:15:00 PM	County	BATTERY LA	KEYSTONE AVE
26	MCP1235004Y	230048980	Montgomery County Police	Property Damage Crash	08/11/2023 06:38:00 PM	County	POWDER MILL RD	NEW HAMPSHIRE AVE
27	DD5612004N	230067649	Rockville Police Departme	Property Damage Crash	11/21/2023 03:49:00 PM	County	DARNESTOWN RD	W MONTGOMERY AVE
28	MCP158000C3	190050604	Montgomery County Police	Property Damage Crash	10/22/2019 06:07:00 AM	Maryland (State)	DARNESTOWN RD	QUINCE ORCHARD RD
29	MCP32630045	230073300	Montgomery County Police	Property Damage Crash	12/23/2023 11:47:00 AM	Maryland (State)	CONNECTICUT AVE	KNOWLES AVE

Dataset Continued...

Some of the attributes in the data set are…

Collision Type: Type of collision (e.g., rear-end, side-impact).

Weather: Weather conditions at the time of the incident.

Surface Condition: Road surface condition (e.g., wet, dry, icy).

Light: Lighting conditions (e.g., daylight, dark, dawn).

Traffic Control: Presence and type of traffic control devices (e.g., stop signs, signals).

Driver Substance Abuse: Information on driver impairment due to substances.

Non-Motorist Substance Abuse: Information on non-motorist impairment due to substances.

Person ID: Unique ID for the person involved in the incident.

There are many more

Preprocessing + Split

1. Formatting for Weka
    - Remove new lines
    - Remove quotation marks
    - Reformat Date to only keep the minutes
2. Remove redundant attributes
3. Remove instances with class null values
4. Replace null values in other attributes
5. Discretize/Binning

After preprocessing, we ended up with 24 attributes.

We did a stratified random train/test split with the use of Weka's "stratifiedRemoveFolds", yielding an 80-20 split stratified with respect to the class label "At Fault". The resulting train and test splits had the same percentage of each class (to the nearest whole instance).

Date/Time Attribute

DateFormatting.py

Users > abhinav > Desktop > Programming > Python > DateFormatting.py > convert_to_minutes

```python
import csv
from datetime import datetime

def convert_to_minutes(date_time_str):
    dt = datetime.strptime(date_time_str, "%m/%d/%Y %I:%M:%S %p")

    minutes_since_midnight = dt.hour * 60 + dt.minute
    return minutes_since_midnight

input_file = 'Crash_Reporting_-_Drivers_DataV4.csv'
output_file = 'CrashReportingDateTime.csv'

column_name = 'Crash Date/Time'

with open(input_file, mode='r', newline='') as infile, open(output_file, mode='w', newline='') as outfile:
    reader = csv.DictReader(infile)
    fieldnames = reader.fieldnames
    writer = csv.DictWriter(outfile, fieldnames=fieldnames)

    writer.writeheader()

    for row in reader:
        row[column_name] = convert_to_minutes(row[column_name])
        writer.writerow(row)

```

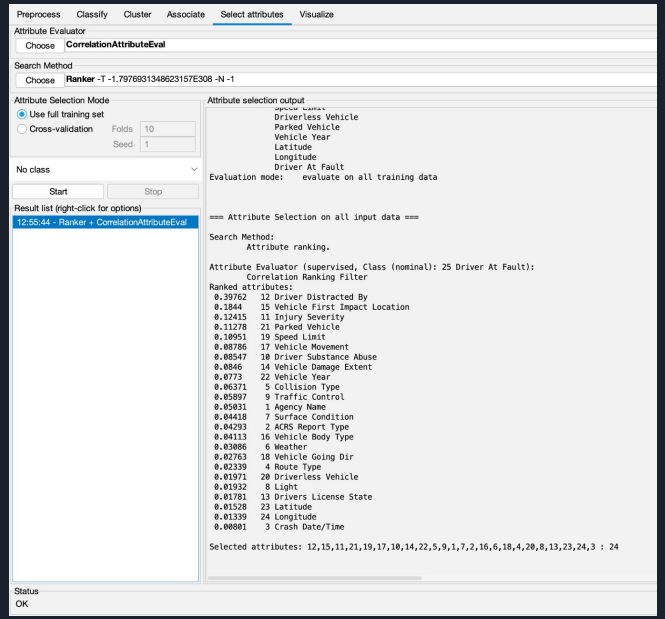
Crash Date/Time	Crash Date/Time
05/27/2021 07:40:00 PM	1180
09/11/2015 01:29:00 PM	809
	865
08/17/2018 02:25:00 PM	1080
08/11/2023 06:00:00 PM	1122
12/06/2023 06:42:00 PM	669
08/28/2023 11:09:00 AM	750
07/27/2023 12:30:00 PM	1000
12/29/2023 04:40:00 PM	1224
11/10/2023 08:24:00 PM	1173
10/16/2023 07:33:00 PM	634
09/30/2023 10:34:00 AM	

Attribute Selection

Correlation Based Feature Selection

Using the CorrelationAttributeEval attribute selection algorithm and the Ranker search method, we obtained the following analysis.

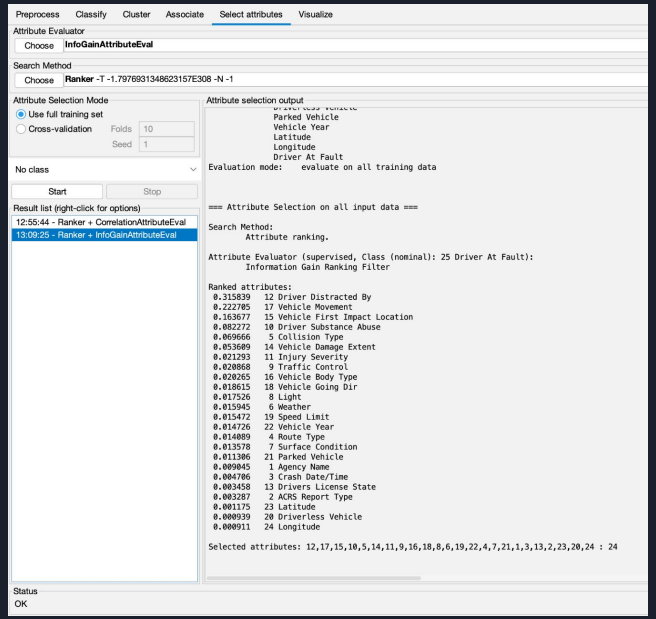
When selecting our attributes, we set a cutoff of 0.075. Thus, the following attributes were selected to be included based on this algorithm: ‘Driver Distracted By’, ‘Vehicle First Impact Location’, ‘Injury Severity’, ‘Parked Vehicle’, ‘Speed Limit’, ‘Vehicle Movement’, ‘Driver Substance Abuse’, ‘Vehicle Damage Extent’, and ‘Vehicle Year’.



Info Gain Based Attribute Selection

Our second dataset was created using the InfoGainAttributeEval algorithm provided by WEKA, which also uses the Ranker search method.

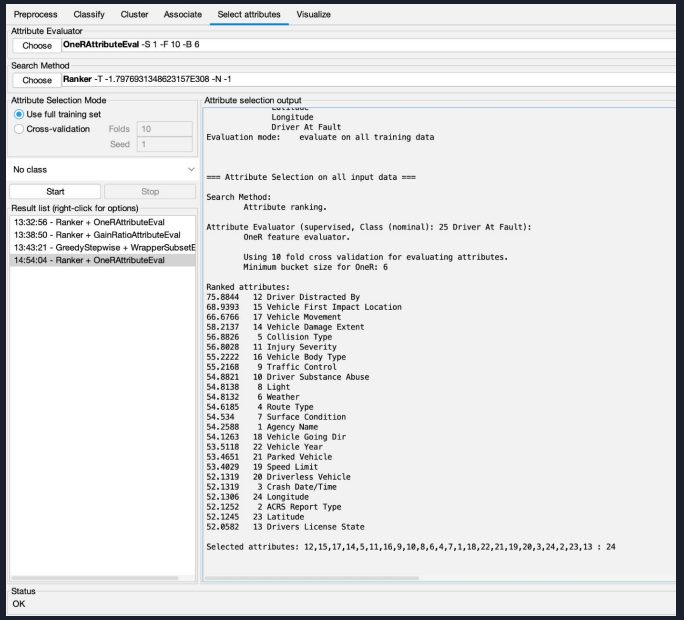
We chose a cutoff of 0.02. As such, we kept the attributes ‘Driver Distracted By’, ‘Vehicle Movement’, ‘Vehicle First Impact Location’, ‘Driver Substance Abuse’, ‘Collision Type’, ‘Vehicle Damage Extent’, ‘Injury Severity’, ‘Traffic Control’, and ‘Vehicle Body Type’.



OneR Based Selection

This dataset was created using the OneRAttributeEval attribute evaluator, which uses the Ranker search method. The following results were obtained.

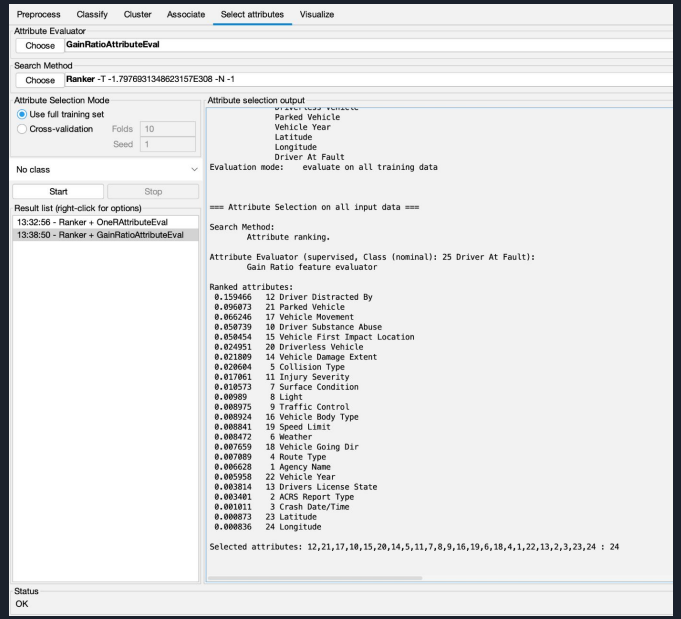
Based on the cutoff value of 54.5, the attributes we chose to keep were: ‘Driver Distracted By’, ‘Vehicle First Impact Location’, ‘Vehicle Movement’, ‘Vehicle Damage Extent’, ‘Collision Type’, ‘Injury Severity’, ‘Vehicle Body Type’, ‘Traffic Control’, ‘Driver Substance Abuse’, ‘Light’, ‘Weather’, ‘Route Type’, and ‘Surface Condition’.



Gain Ratio Evaluation

This dataset was created using the GainRatioAttributeEval, which uses the Ranker search method. The results are shown below.

After setting a cutoff of 0.01, we decided to keep the following attributes: ‘Driver Distracted By’, ‘Parked Vehicle’, ‘Vehicle Movement’, ‘Driver Substance Abuse’, ‘Vehicle First Impact Location’, ‘Driverless Vehicle’, ‘Vehicle Damage Extent’, ‘Collision Type’, ‘Injury Severity’, and ‘Surface Condition’.



Self Selected Attributes

For the self-selected attributes, we chose to include any attribute that was suggested to be kept by any one of the attribute selection algorithms above. In other words, the only attributes removed were the ones that were considered ‘useless’ by every single attribute selection algorithm.

Model Selection

Algorithms and Models Used

1. J48
2. Naive Bayes: This model forms probabilistic predictions based on the training set using Bayes' theorem and the naive assumption that attributes are independent from one another. The following formulas are used:

$$P(C_i|X) = \frac{P(X|C_i)\,P(C_i)}{P(X)} \text{ when } P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) \text{ where } X = (x_1, x_2, \ldots, x_n)$$

   is the instance. The prediction is the class (C_i) *with the highest* $P(C_i|X)$ *value.*

3. 1R: Level 1 decision tree. It forms a rule set with the following pseudocode:

   For each attribute
   For each value of the attribute
   count frequency of each class
   find the most frequent class
   make rule: assign that class to this attribute-value
   Compute the error rate of the rules (of this attribute)
   Choose the rules with the smallest error rate

4. Random Tree

Classification Results: Naive Bayes

Attribute Selection Type	Accuracy
Correlation	80.52%
GainRatio	79.58
OneR	79.89
InfoGain	78.40
Self-Picked	81.16

Classification Results: J48

Attribute Selection Type	Accuracy
Correlation	83.35
GainRatio	86.46
OneR	86.92
InfoGain	87.85
Self-Picked	86.87

Classification Results: OneR

Attribute Selection Type	Accuracy
Correlation	76.13
GainRatio	76.13
OneR	76.13
InfoGain	76.13
Self-Picked	76.13

Classification Results: Random Tree

Attribute Selection Type	Accuracy
Correlation	78.63
GainRatio	85.01
OneR	84.44
InfoGain	81.84
Self-Picked	81.325

Results and Conclusions

- We used accuracy as a measure of how good the model was because our data had very little skew (56% yes, and 44% no)
- Our model determined that J48 with the InfoGainEval dataset had the best accuracy of 87%.
- Highest TP Rate: 90.3%
- Lowest Mean Squared Error: 0.264
- This accuracy is decently high and provides decent prediction of who is at fault

How to Reproduce

1. Under the Preprocess tab, remove the attributes which prevent arff to csv conversion
2. Open Weka and load the “rawdata.csv” dataset.
3. Remove redundant attributes
4. Parse dates into only the minutes using the python code (code included on report)
5. Select the attributes according to the corresponding attribute selection algorithm.
6. Split the data into train and test splits.
7. Train the desired model (J48) using the test set as a “supplied test set”.
8. Click Start.

Sources

Google Drive:
https://drive.google.com/drive/folders/110G13sFcdTgHHQABhNVw3vTYZuXPTyYH

Dataset:

https://catalog.data.gov/dataset/crash-reporting-drivers-data